Chapter 5 – Matrix Approach to Simple Linear Regression

- Definition: A matrix is a rectangular array of numbers or symbolic elements
- In many applications, the rows of a matrix will represent individuals cases (people, items, plants, animals,...) and columns will represent attributes or characteristics
- The dimension of a matrix is it number of rows and columns, often denoted as *r* x *c* (*r* rows by *c* columns)
- Can be represented in full form or abbreviated form:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1c} \\ a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2c} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{ic} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{r1} & a_{r2} & \cdots & a_{rj} & \cdots & a_{rc} \end{bmatrix} = \left[a_{ij} \right] \quad i = 1,...,r;\ j = 1,...,c$$

Special Types of Matrices

Square Matrix: Number of rows = # of Columns $(r = c)$

$$\mathbf{A} = \begin{bmatrix} 20 & 32 & 50 \\ 12 & 28 & 42 \\ 28 & 46 & 60 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

Vector: Matrix with one column (column vector) or one row (row vector)

$$\mathbf{C} = \begin{bmatrix} 57 \\ 24 \\ 18 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} \qquad \mathbf{E}' = \begin{bmatrix} 17 & 31 \end{bmatrix} \qquad \mathbf{F}' = \begin{bmatrix} f_1 & f_2 & f_3 \end{bmatrix}$$

Transpose: Matrix formed by interchanging rows and columns of a matrix (use "prime" to denote transpose)

$$\underset{2\times3}{\mathbf{G}} = \begin{bmatrix} 6 & 15 & 22 \\ 8 & 13 & 25 \end{bmatrix} \qquad \underset{3\times2}{\mathbf{G}'} = \begin{bmatrix} 6 & 8 \\ 15 & 13 \\ 22 & 25 \end{bmatrix}$$

$$\underset{r\times c}{\mathbf{H}} = \begin{bmatrix} h_{11} & \cdots & h_{1c} \\ \vdots & & \vdots \\ h_{r1} & \cdots & h_{rc} \end{bmatrix} = \left[h_{ij} \right] \quad i = 1,...,r;\ j = 1,...,c \quad \Rightarrow \quad \underset{c\times r}{\mathbf{H}'} = \begin{bmatrix} h_{11} & \cdots & h_{r1} \\ \vdots & & \vdots \\ h_{1c} & \cdots & h_{rc} \end{bmatrix} = \left[h_{ji} \right] \quad j = 1,...,c;\ i = 1,...,r$$

Matrix Equality: Matrices of the same dimension, and corresponding elements in same cells are all equal:

$$\mathbf{A} = \begin{bmatrix} 4 & 6 \\ 12 & 10 \end{bmatrix} = \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \quad \Rightarrow \quad b_{11} = 4, b_{12} = 6, b_{21} = 12, b_{22} = 10$$

Example: Bollywood Box Office Data

Data in original (non-transformed) units.

$$\text{Response Vector: } \underset{n\times 1}{\mathbf{Y}} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}$$

$$\underset{1\times n}{\mathbf{Y}'} = \begin{bmatrix} Y_1 & Y_2 & \cdots & Y_n \end{bmatrix}$$

$$\text{Design Matrix: } \underset{n\times 2}{\mathbf{X}} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix}$$

$$\underset{2\times n}{\mathbf{X}'} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ X_1 & X_2 & \cdots & X_n \end{bmatrix}$$

As we will see below, the **X** matrix includes a column of 1^s, this will allow for the intercept term.

X			Y
1	36.00		95.64
1	77.00		55.65
1	90.00		110.01
1	16.00		11.16
1	9.50		5.19
1	4.50		2.23
1	26.00		49.07
1	36.00		101.61
1	39.00		53.04
1	18.00		46.59
1	24.00		61.47
1	52.00		72.26
1	120.00		107.71
1	125.00		23.07
1	16.00		11.69
1	11.00		10.42
1	5.50		3.36
1	8.00		2.17
1	4.50		0.97
1	5.00		1.02
1	11.00		1.28
1	5.50		0.63
1	26.00		9.44
1	29.00		3.93
1	16.00		2.10
1	34.00		34.03
1	27.00		6.76
1	9.00		5.79
1	16.00		3.72
1	16.00		1.55
1	22.00		12.06
1	27.00		13.32
1	18.00		5.91
1	43.00		37.95
1	32.00		27.71
1	15.00		5.70
1	24.00		36.52
1	11.00		1.16
1	12.00		2.41
1	19.00		31.04
1	31.00		25.87
1	20.00		11.25
1	10.00		3.83
1	150.00		262.58
1	130.00		208.44
1	115.00		181.11
1	50.00		185.83
1	65.00		66.10
1	83.00		112.96
1	72.00		52.38
1	78.00		55.79
1	80.00		60.93
1	52.00		109.18
1	65.00		96.34
1	10.50		78.42

Each row represents an individual film.
There are n=55 rows in both **X** and **Y**.

Matrix Addition and Subtraction

Addition and Subtraction of 2 Matrices of Common Dimension:

$$\underset{2\times2}{\mathbf{C}}=\begin{bmatrix}4 & 7\\ 10 & 12\end{bmatrix} \qquad \underset{2\times2}{\mathbf{D}}=\begin{bmatrix}2 & 0\\ 14 & 6\end{bmatrix} \qquad \underset{2\times2}{\mathbf{C}+\mathbf{D}}=\begin{bmatrix}4+2 & 7+0\\ 10+14 & 12+6\end{bmatrix}=\begin{bmatrix}6 & 7\\ 24 & 18\end{bmatrix} \qquad \underset{2\times2}{\mathbf{C}-\mathbf{D}}=\begin{bmatrix}4-2 & 7-0\\ 10-14 & 12-6\end{bmatrix}=\begin{bmatrix}2 & 7\\ -4 & 6\end{bmatrix}$$

$$\underset{r\times c}{\mathbf{A}}=\begin{bmatrix}a_{11} & \cdots & a_{1c}\\ \vdots & & \vdots\\ a_{r1} & \cdots & a_{rc}\end{bmatrix}=\left[a_{ij}\right] \quad i=1,...,r;\ j=1,...,c \qquad \underset{r\times c}{\mathbf{B}}=\begin{bmatrix}b_{11} & \cdots & b_{1c}\\ \vdots & & \vdots\\ b_{r1} & \cdots & b_{rc}\end{bmatrix}=\left[b_{ij}\right] \quad i=1,...,r;\ j=1,...,c$$

$$\underset{r\times c}{\mathbf{A}+\mathbf{B}}=\begin{bmatrix}a_{11}+b_{11} & \cdots & a_{1c}+b_{1c}\\ \vdots & & \vdots\\ a_{r1}+b_{r1} & \cdots & a_{rc}+b_{rc}\end{bmatrix}=\left[a_{ij}+b_{ij}\right] \quad i=1,...,r;\ j=1,...,c$$

$$\underset{r\times c}{\mathbf{A}-\mathbf{B}}=\begin{bmatrix}a_{11}-b_{11} & \cdots & a_{1c}-b_{1c}\\ \vdots & & \vdots\\ a_{r1}-b_{r1} & \cdots & a_{rc}-b_{rc}\end{bmatrix}=\left[a_{ij}-b_{ij}\right] \quad i=1,...,r;\ j=1,...,c$$

Regression Example:

$$\underset{n\times1}{\mathbf{Y}}=\begin{bmatrix}Y_1\\ Y_2\\ \vdots\\ Y_n\end{bmatrix} \qquad \underset{n\times1}{\mathbf{E}\{\mathbf{Y}\}}=\begin{bmatrix}E\{Y_1\}\\ E\{Y_2\}\\ \vdots\\ E\{Y_n\}\end{bmatrix} \qquad \underset{n\times1}{\boldsymbol{\varepsilon}}=\begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \vdots\\ \varepsilon_n\end{bmatrix} \qquad \underset{n\times1}{\mathbf{Y}}=\underset{n\times1}{\mathbf{E}\{\mathbf{Y}\}}+\underset{n\times1}{\boldsymbol{\varepsilon}} \quad \text{since} \quad \begin{bmatrix}Y_1\\ Y_2\\ \vdots\\ Y_n\end{bmatrix}=\begin{bmatrix}E\{Y_1\}\\ E\{Y_2\}\\ \vdots\\ E\{Y_n\}\end{bmatrix}+\begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \vdots\\ \varepsilon_n\end{bmatrix}=\begin{bmatrix}E\{Y_1\}+\varepsilon_1\\ E\{Y_2\}+\varepsilon_2\\ \vdots\\ E\{Y_n\}+\varepsilon_n\end{bmatrix}$$

Matrix Multiplication

Multiplication of a Matrix by a Scalar (single number):

$$k=3 \quad \mathbf{A}=\begin{bmatrix}2 & 1\\ -2 & 7\end{bmatrix} \quad \Rightarrow \quad k\mathbf{A}=\begin{bmatrix}3(2) & 3(1)\\ 3(-2) & 3(7)\end{bmatrix}=\begin{bmatrix}6 & 3\\ -6 & 21\end{bmatrix}$$

Multiplication of a Matrix by a Matrix (#cols(**A**) = #rows(**B**)):

If $c_A=r_B$: $\underset{r_A\times c_A}{\mathbf{A}}\ \underset{r_B\times c_B}{\mathbf{B}}=\underset{r_A\times c_B}{\mathbf{AB}}=\left[ab_{ij}\right]\ i=1,...,r_A;\ j=1,...,c_B$

$ab_{ij}\equiv$ sum of the products of the $c_A=r_B$ elements of i$^{\text{th}}$ row of **A** and j$^{\text{th}}$ column of **B**:

$$\underset{3\times2}{\mathbf{A}}=\begin{bmatrix}2 & 5\\ 3 & -1\\ 0 & 7\end{bmatrix} \qquad \underset{2\times2}{\mathbf{B}}=\begin{bmatrix}3 & -1\\ 2 & 4\end{bmatrix}$$

$$\underset{3\times2}{\mathbf{A}}\ \underset{2\times2}{\mathbf{B}}=\underset{3\times2}{\mathbf{AB}}=\begin{bmatrix}2(3)+5(2) & 2(-1)+5(4)\\ 3(3)+(-1)(2) & 3(-1)+(-1)(4)\\ 0(3)+7(2) & 0(-1)+7(4)\end{bmatrix}=\begin{bmatrix}16 & 18\\ 7 & -7\\ 14 & 28\end{bmatrix}$$

If $c_A=r_B=c$: $\underset{r_A\times c_A}{\mathbf{A}}\ \underset{r_B\times c_B}{\mathbf{B}}=\underset{r_A\times c_B}{\mathbf{AB}}=\left[ab_{ij}\right]=\left[\sum_{k=1}^{c}a_{ik}b_{kj}\right]\ i=1,...,r_A;\ j=1,...,c_B$

Note that the $(i,j)^{th}$ element of **AB** is the **sumproduct** of the i^{th} row of **A** and the j^{th} column of **B**. Also, it is important to remember that unlike matrix addition and subtraction, matrix multiplication is not element by element.

Examples of Matrix Multiplication

Simultaneous Equations: $a_{11}x_1 + a_{12}x_2 = y_1 \quad a_{21}x_1 + a_{22}x_2 = y_2$

(2 equations: x_1, x_2 unknown): $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$

$\Rightarrow \quad \begin{bmatrix} a_{11}x_1 + a_{12}x_2 \\ a_{21}x_1 + a_{22}x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \quad \Rightarrow \quad \mathbf{AX = Y}$

Sum of Squares: $4^2 + (-2)^2 + 3^2 = \begin{bmatrix} 4 & -2 & 3 \end{bmatrix}\begin{bmatrix} 4 \\ -2 \\ 3 \end{bmatrix} = \begin{bmatrix} 29 \end{bmatrix}$

Regression Equation (Expected Values): $\begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix}\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 X_1 \\ \beta_0 + \beta_1 X_2 \\ \vdots \\ \beta_0 + \beta_1 X_n \end{bmatrix}$

Matrices used in simple linear regression (that generalize to multiple regression):

$$\mathbf{Y'Y} = \begin{bmatrix} Y_1 & Y_2 & \cdots & Y_n \end{bmatrix}\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \sum_{i=1}^{n} Y_i^2$$

$$\mathbf{X'X} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ X_1 & X_2 & \cdots & X_n \end{bmatrix}\begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix} = \begin{bmatrix} n & \sum_{i=1}^{n} X_i \\ \sum_{i=1}^{n} X_i & \sum_{i=1}^{n} X_i^2 \end{bmatrix}$$

$$\mathbf{X'Y} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ X_1 & X_2 & \cdots & X_n \end{bmatrix}\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} Y_i \\ \sum_{i=1}^{n} X_i Y_i \end{bmatrix} \qquad \mathbf{X\beta} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix}\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 X_1 \\ \beta_0 + \beta_1 X_2 \\ \vdots \\ \beta_0 + \beta_1 X_n \end{bmatrix}$$

Example: Bollywood Box Office Data

Y'Y		X'X			X'Y
304471.8		55	2147		2578.35
		2147	155976.5		190927.5

Special Types of Matrices

Symmetric Matrix: Square matrix with a transpose equal to itself: **A = A'**:

$$\mathbf{A} = \begin{bmatrix} 6 & 19 & -8 \\ 19 & 14 & 3 \\ -8 & 3 & 1 \end{bmatrix} \qquad \mathbf{A'} = \begin{bmatrix} 6 & 19 & -8 \\ 19 & 14 & 3 \\ -8 & 3 & 1 \end{bmatrix} = \mathbf{A}$$

Diagonal Matrix: Square matrix with all off-diagonal elements equal to 0:

$$\mathbf{A} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{bmatrix}$$

Note:Diagonal matrices are symmetric (not vice versa)

Identity Matrix: Diagonal matrix with all diagonal elements equal to 1 (acts like multiplying a scalar by 1):

$$\underset{3\times3}{\mathbf{I}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \underset{3\times3}{\mathbf{A}} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \Rightarrow \mathbf{IA} = \mathbf{AI} = \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Scalar Matrix: Diagonal matrix with all diagonal elements equal to a single number"

$$\begin{bmatrix} k & 0 & 0 & 0 \\ 0 & k & 0 & 0 \\ 0 & 0 & k & 0 \\ 0 & 0 & 0 & k \end{bmatrix} = k \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = k \underset{4\times4}{\mathbf{I}}$$

1-Vector and matrix and zero-vector:

$$\underset{r\times1}{\mathbf{1}} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \qquad \underset{r\times r}{\mathbf{J}} = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} \qquad \underset{r\times1}{\mathbf{0}} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Note: $\underset{1\times r}{\mathbf{1}'}\underset{r\times1}{\mathbf{1}} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = r \qquad \underset{r\times1}{\mathbf{1}}\underset{1\times r}{\mathbf{1}'} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} = \underset{r\times r}{\mathbf{J}}$

All of these matrices have important roles in regression analysis, particularly when we have multiple predictor variables. Software packages (including R and EXCEL (to a lesser extent)) can be used to make all relevant matrix computations.

Linear Dependence and Matrix Rank

- Linear Dependence: When a linear combination of the columns (rows) of a matrix produces a zero vector (one or more columns (rows) can be written as linear function of the other columns (rows))
- Rank of a matrix: Number of linearly independent columns (rows) of the matrix. Rank cannot exceed the minimum of the number of rows or columns of the matrix. rank(**A**) $\leq \min(r_A, c_a)$
- A matrix is full rank if rank(**A**) = $\min(r_A, c_a)$

$$\underset{2\times 2}{\mathbf{A}} = \begin{bmatrix} 1 & -3 \\ -4 & 12 \end{bmatrix} = \begin{bmatrix} \underset{2\times 1}{\mathbf{A}_1} & \underset{2\times 1}{\mathbf{A}_2} \end{bmatrix} \qquad 3\mathbf{A}_1 + \mathbf{A}_2 = \mathbf{0}$$ Columns of **A** are linearly dependent rank(**A**) = 1

$$\underset{2\times 2}{\mathbf{B}} = \begin{bmatrix} 4 & -3 \\ 4 & 12 \end{bmatrix} = \begin{bmatrix} \underset{2\times 1}{\mathbf{B}_1} & \underset{2\times 1}{\mathbf{B}_2} \end{bmatrix} \qquad 0\mathbf{B}_1 + 0\mathbf{B}_2 = \mathbf{0}$$ Columns of **B** are linearly independent rank(**B**) = 2

For all well posed regression problems, **X** and **X'X** will be full rank.

Matrix Inverse

- Note: For scalars (except 0), when we multiply a number, by its reciprocal, we get 1: $2(1/2)=1 \qquad x(1/x)=x(x^{-1})=1$
- In matrix form if **A** is a square matrix and full rank (all rows and columns are linearly independent), then **A** has an inverse: $\mathbf{A}^{-1}$ such that: $\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$

$$\mathbf{A} = \begin{bmatrix} 2 & 8 \\ 4 & -2 \end{bmatrix} \qquad \mathbf{A}^{-1} = \begin{bmatrix} \frac{2}{36} & \frac{8}{36} \\ \frac{4}{36} & \frac{-2}{36} \end{bmatrix}$$

$$\mathbf{A}^{-1}\mathbf{A} = \begin{bmatrix} \frac{2}{36} & \frac{8}{36} \\ \frac{4}{36} & \frac{-2}{36} \end{bmatrix} \begin{bmatrix} 2 & 8 \\ 4 & -2 \end{bmatrix} = \begin{bmatrix} \frac{4}{36}+\frac{32}{36} & \frac{16}{36}-\frac{16}{36} \\ \frac{8}{36}-\frac{8}{36} & \frac{32}{36}+\frac{4}{36} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{I}$$

$$\mathbf{B} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 6 \end{bmatrix} \qquad \mathbf{B}^{-1} = \begin{bmatrix} 1/4 & 0 & 0 \\ 0 & -1/2 & 0 \\ 0 & 0 & 1/6 \end{bmatrix}$$

$$\mathbf{B}\mathbf{B}^{-1} = \begin{bmatrix} 4(1/4)+0+0 & 0+0+0 & 0+0+0 \\ 0+0+0 & 0+(-2)(-1/2)+0 & 0+0+0 \\ 0+0+0 & 0+0+0 & 0+0+6(1/6) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{I}$$

Obtaining inverses of 2x2 matrices is very simple. For matrices that are larger than 2x2, we will use R and/or EXCEL to obtain inverses when necessary. Note that all software packages are internally doing the computations in linear regression programs.

Computing the Inverse of a 2x2 Matrix

$\underset{2\times2}{\mathbf{A}} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \equiv$ full rank (columns/rows are linearly independent)

Determinant of $\mathbf{A} \equiv |\mathbf{A}| = a_{11}a_{22} - a_{12}a_{21}$

Note: If A is not full rank (for some value k): $a_{11} = ka_{12} \quad a_{21} = ka_{22}$

$\Rightarrow |\mathbf{A}| = a_{11}a_{22} - a_{12}a_{21} = ka_{12}a_{22} - a_{12}ka_{22} = 0$

$\underset{2\times2}{\mathbf{A}^{-1}} = \frac{1}{|\mathbf{A}|}\begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$ Thus $\mathbf{A}^{-1}$ does not exist if $\mathbf{A}$ is not full rank

While there are rules for general $r \times r$ matrices, we will use computers to solve them

Regression Example:

$$\mathbf{X} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix} \quad \Rightarrow \quad \mathbf{X'X} = \begin{bmatrix} n & \sum_{i=1}^{n} X_i \\ \sum_{i=1}^{n} X_i & \sum_{i=1}^{n} X_i^2 \end{bmatrix}$$

$$\Rightarrow \quad |\mathbf{X'X}| = n\sum_{i=1}^{n} X_i^2 - \left(\sum_{i=1}^{n} X_i\right)^2 = n\left(\sum_{i=1}^{n} X_i^2 - \frac{\left(\sum_{i=1}^{n} X_i\right)^2}{n}\right) = n\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2$$

$$\Rightarrow \quad \left(\mathbf{X'X}\right)^{-1} = \frac{1}{n\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2}\begin{bmatrix} \sum_{i=1}^{n} X_i^2 & -\sum_{i=1}^{n} X_i \\ -\sum_{i=1}^{n} X_i & n \end{bmatrix}$$

Note: $\sum_{i=1}^{n} X_i = n\overline{X} \qquad \sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2 = \sum_{i=1}^{n} X_i^2 - n\overline{X}^2 \Rightarrow \sum_{i=1}^{n} X_i^2 = \sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2 + n\overline{X}^2$

$$\Rightarrow \quad \left(\mathbf{X'X}\right)^{-1} = \begin{bmatrix} \frac{1}{n} + \frac{\overline{X}^2}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} & -\frac{\overline{X}}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} \\ -\frac{\overline{X}}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} & \frac{1}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} \end{bmatrix}$$

Example: Bollywood Box Office Data

$$n = 55 \quad \overline{X} = 39.04 \quad \sum X_i = 2147 \quad \sum X_i^2 = 155976.5 \quad SS_{XX} = \sum \left(X_i - \overline{X} \right)^2 = 72165.43$$

$$\Rightarrow \quad \left(\mathbf{X'X} \right)^{-1} = \frac{1}{55(72165.43)} \begin{bmatrix} 155976.5 & -2147 \\ -2147 & 55 \end{bmatrix} = \begin{bmatrix} \frac{1}{55} + \frac{39.04^2}{72165.43} & -\frac{39.04}{72165.43} \\ -\frac{39.04}{72165.43} & \frac{1}{72165.43} \end{bmatrix}$$

Solving Simultaneous Equations with a Matrix Inverse

AY = C where **A** and **C** are matrices of of constants, **Y** is matrix of unknowns

$\Rightarrow \mathbf{A^{-1}AY} = \mathbf{A^{-1}C} \quad \Rightarrow \quad \mathbf{Y = A^{-1}C}$ (assuming **A** is square and full rank)

Equation 1: $12y_1 + 6y_2 = 48$ Equation 2: $10y_1 - 2y_2 = 12$

$$\mathbf{A} = \begin{bmatrix} 12 & 6 \\ 10 & -2 \end{bmatrix} \qquad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 48 \\ 12 \end{bmatrix} \qquad \mathbf{Y = A^{-1}C}$$

$$\Rightarrow \quad \mathbf{A^{-1}} = \frac{1}{12(-2) - 6(10)} \begin{bmatrix} -2 & -6 \\ -10 & 12 \end{bmatrix} = \frac{1}{84} \begin{bmatrix} 2 & 6 \\ 10 & -12 \end{bmatrix}$$

$$\mathbf{Y = A^{-1}C} = \frac{1}{84} \begin{bmatrix} 2 & 6 \\ 10 & -12 \end{bmatrix} \begin{bmatrix} 48 \\ 12 \end{bmatrix} = \frac{1}{84} \begin{bmatrix} 96 + 72 \\ 480 - 144 \end{bmatrix} = \frac{1}{84} \begin{bmatrix} 168 \\ 336 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

Note the wisdom of waiting to divide by $|\mathbf{A}|$ at end of calculation!

Serious rounding errors can occur when dividing by the determinant "too early" in manual computations.

Some Useful Matrix Results (Assuming Conformability of Matrices to Operations

All rules assume that the matrices are conformable to operations:

Addition Rules: $\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A} \qquad (\mathbf{A} + \mathbf{B}) + \mathbf{C} = \mathbf{A} + (\mathbf{B} + \mathbf{C})$

Multiplication Rules: $\mathbf{(AB)C} = \mathbf{A(BC)} \qquad \mathbf{C}(\mathbf{A} + \mathbf{B}) = \mathbf{CA} + \mathbf{CB} \qquad k(\mathbf{A} + \mathbf{B}) = k\mathbf{A} + k\mathbf{B} \quad k \equiv$ scalar

Transpose Rules: $(\mathbf{A}')' = \mathbf{A} \qquad (\mathbf{A} + \mathbf{B})' = \mathbf{A}' + \mathbf{B}' \qquad (\mathbf{AB})' = \mathbf{B'A'} \qquad \mathbf{(ABC)' = C'B'A'}$

Inverse Rules (Full Rank, Square Matrices): $\mathbf{(AB)^{-1} = B^{-1}A^{-1}} \qquad \mathbf{(ABC)^{-1} = C^{-1}B^{-1}A^{-1}} \qquad \mathbf{(A^{-1})^{-1} = A} \qquad \mathbf{(A')^{-1} = (A^{-1})'}$

Random Vectors and Matrices

Shown for case of n=3, generalizes to any n:

Random variables: $Y_1, Y_2, Y_3 \quad \Rightarrow \quad \mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix}$

Expectation: $\mathbf{E}\{\mathbf{Y}\} = \begin{bmatrix} E\{Y_1\} \\ E\{Y_2\} \\ E\{Y_3\} \end{bmatrix}$ In general: $\underset{n\times p}{\mathbf{E}\{\mathbf{Y}\}} = \left[E\{Y_{ij}\} \right] \quad i = 1,...,n;\ j = 1,...,p$

Variance-Covariance Matrix for a Random Vector:

$$\sigma^2\{\mathbf{Y}\} = E\left\{\left[\mathbf{Y} - \mathbf{E}\{\mathbf{Y}\}\right]\left[\mathbf{Y} - \mathbf{E}\{\mathbf{Y}\}\right]'\right\} = \mathbf{E}\left\{\begin{bmatrix} Y_1 - E\{Y_1\} \\ Y_2 - E\{Y_2\} \\ Y_3 - E\{Y_3\} \end{bmatrix}\begin{bmatrix} Y_1 - E\{Y_1\} & Y_2 - E\{Y_2\} & Y_3 - E\{Y_3\} \end{bmatrix}\right\} =$$

$$= \mathbf{E}\left\{\begin{bmatrix} \left(Y_1 - E\{Y_1\}\right)^2 & \left(Y_1 - E\{Y_1\}\right)\left(Y_2 - E\{Y_2\}\right) & \left(Y_1 - E\{Y_1\}\right)\left(Y_3 - E\{Y_3\}\right) \\ \left(Y_2 - E\{Y_2\}\right)\left(Y_1 - E\{Y_1\}\right) & \left(Y_2 - E\{Y_2\}\right)^2 & \left(Y_2 - E\{Y_2\}\right)\left(Y_3 - E\{Y_3\}\right) \\ \left(Y_3 - E\{Y_3\}\right)\left(Y_1 - E\{Y_1\}\right) & \left(Y_3 - E\{Y_3\}\right)\left(Y_2 - E\{Y_2\}\right) & \left(Y_3 - E\{Y_3\}\right)^2 \end{bmatrix}\right\} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 \end{bmatrix} = \Sigma$$

Linear Regression Example (n = 3)

Error terms are assumed to be independent, with mean 0, constant variance σ^2:

$$\Rightarrow E\{\varepsilon_i\} = 0 \qquad \sigma^2\{\varepsilon_i\} = \sigma^2 \qquad \sigma\{\varepsilon_i, \varepsilon_j\} = 0 \quad \forall\, i \neq j$$

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix} \qquad \mathbf{E}\{\boldsymbol{\varepsilon}\} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \boldsymbol{\sigma}^2\{\boldsymbol{\varepsilon}\} = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix} = \sigma^2\mathbf{I}$$

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \qquad \mathbf{E}\{\mathbf{Y}\} = \mathbf{E}\{\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}\} = \mathbf{X}\boldsymbol{\beta} + \mathbf{E}\{\boldsymbol{\varepsilon}\} = \mathbf{X}\boldsymbol{\beta} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ 1 & X_3 \end{bmatrix}\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 X_1 \\ \beta_0 + \beta_1 X_2 \\ \beta_0 + \beta_1 X_3 \end{bmatrix}$$

$$\boldsymbol{\sigma}^2\{\mathbf{Y}\} = \boldsymbol{\sigma}^2\{\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}\} = \boldsymbol{\sigma}^2\{\boldsymbol{\varepsilon}\} = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix} = \sigma^2\mathbf{I}$$

Mean and Variance of Linear Functions of Y

$\underset{k\times n}{\mathbf{A}} \equiv$ matrix of fixed constants $\quad \underset{n\times 1}{\mathbf{Y}} \equiv$ random vector

$$\underset{k\times 1}{\mathbf{W}} = \mathbf{AY} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{k1} & \cdots & a_{kn} \end{bmatrix} \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} \equiv \text{ random vector: } \mathbf{W} = \begin{bmatrix} W_1 \\ \vdots \\ W_k \end{bmatrix} = \begin{bmatrix} a_{11}Y_1 + \ldots + a_{1n}Y_n \\ \vdots \\ a_{k1}Y_1 + \ldots + a_{kn}Y_n \end{bmatrix}$$

$$\mathbf{E}\{\mathbf{W}\} = \begin{bmatrix} E\{W_1\} \\ \vdots \\ E\{W_k\} \end{bmatrix} = \begin{bmatrix} E\{a_{11}Y_1 + \ldots + a_{1n}Y_n\} \\ \vdots \\ E\{a_{k1}Y_1 + \ldots + a_{kn}Y_n\} \end{bmatrix} = \begin{bmatrix} a_{11}E\{Y_1\} + \ldots + a_{1n}E\{Y_n\} \\ \vdots \\ a_{k1}E\{Y_1\} + \ldots + a_{kn}E\{Y_n\} \end{bmatrix} =$$

$$= \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{k1} & \cdots & a_{kn} \end{bmatrix} \begin{bmatrix} E\{Y_1\} \\ \vdots \\ E\{Y_n\} \end{bmatrix} = \mathbf{AE}\{\mathbf{Y}\}$$

$$\boldsymbol{\sigma}^2\{\mathbf{W}\} = \mathbf{E}\left\{\left[\mathbf{AY} - \mathbf{AE}\{\mathbf{Y}\}\right]\left[\mathbf{AY} - \mathbf{AE}\{\mathbf{Y}\}\right]'\right\} = \mathbf{E}\left\{\left[\mathbf{A}\left(\mathbf{Y} - \mathbf{E}\{\mathbf{Y}\}\right)\right]\left[\mathbf{A}\left(\mathbf{Y} - \mathbf{E}\{\mathbf{Y}\}\right)\right]'\right\} =$$

$$= \mathbf{E}\left\{\left[\mathbf{A}\left(\mathbf{Y} - \mathbf{E}\{\mathbf{Y}\}\right)\right]\left[\left(\mathbf{Y} - \mathbf{E}\{\mathbf{Y}\}\right)'\mathbf{A}'\right]\right\} = \mathbf{AE}\left\{\left(\mathbf{Y} - \mathbf{E}\{\mathbf{Y}\}\right)\left(\mathbf{Y} - \mathbf{E}\{\mathbf{Y}\}\right)'\right\}\mathbf{A}' = \mathbf{A}\boldsymbol{\sigma}^2\{\mathbf{Y}\}\mathbf{A}'$$

Multivariate Normal Distribution – General Case

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \qquad \boldsymbol{\mu} = \mathbf{E}\{\mathbf{Y}\} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} \qquad \boldsymbol{\Sigma} = \boldsymbol{\sigma}^2\{\mathbf{Y}\} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \cdots & \sigma_n^2 \end{bmatrix}$$

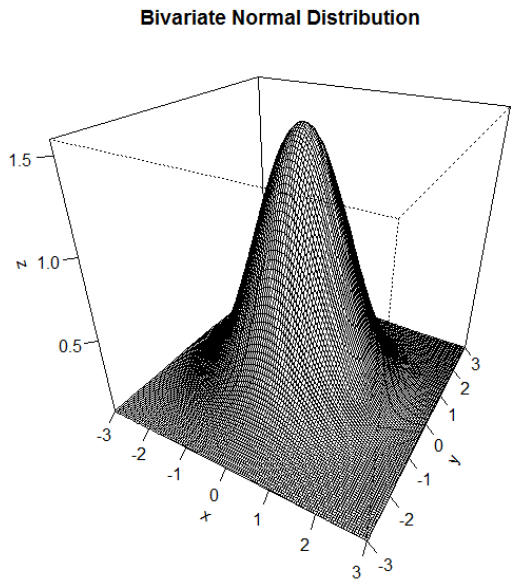
Multivariate Normal Density function:

$$f(\mathbf{Y}) = (2\pi)^{-n/2}|\boldsymbol{\Sigma}|^{-1/2}\exp\left[-\frac{1}{2}(\mathbf{Y} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{Y} - \boldsymbol{\mu})\right] \qquad \mathbf{Y} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$\Rightarrow Y_i \sim N(\mu_i, \sigma_i^2) \quad i = 1, \ldots, n \qquad \sigma\{Y_i, Y_j\} \equiv \sigma_{ij} \quad i \neq j$$

Note, if $\mathbf{A}$ is a (full rank) matrix of fixed constants:

$\mathbf{W} = \mathbf{AY} \sim N(\mathbf{A}\boldsymbol{\mu}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}')$



Bivariate Normal Density with $\mu_1=\mu_2=0$, $\sigma_1=\sigma_2=1$, $\sigma_{12}=0$.

Simple Linear Regression in Matrix Form

Simple Linear Regression Model: $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i \quad i = 1, ..., n$

$$\Rightarrow \begin{bmatrix} Y_1 = \beta_0 + \beta_1 X_1 + \varepsilon_1 \\ Y_2 = \beta_0 + \beta_1 X_2 + \varepsilon_2 \\ \vdots \\ Y_n = \beta_0 + \beta_1 X_n + \varepsilon_n \end{bmatrix}$$

Defining:

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix} \qquad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \qquad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix} \Rightarrow \quad \mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \qquad \text{since: } \mathbf{X}\boldsymbol{\beta} = \begin{bmatrix} \beta_0 + \beta_1 X_1 \\ \beta_0 + \beta_1 X_2 \\ \vdots \\ \beta_0 + \beta_1 X_n \end{bmatrix} = \mathbf{E}\{\mathbf{Y}\}$$

Assuming constant variance, and independence of error terms ε_i :

$$\boldsymbol{\sigma}^2\{\mathbf{Y}\} = \boldsymbol{\sigma}^2\{\boldsymbol{\varepsilon}\} = \begin{bmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2 \end{bmatrix} = \underset{n \times n}{\sigma^2 \mathbf{I}}$$

Further, assuming normal distribution for error terms ε_i : $\mathbf{Y} \sim N\left(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}\right)$

Least Squares Estimation in Matrix Form

Normal equations obtained from: $\frac{\partial Q}{\partial \beta_0}, \frac{\partial Q}{\partial \beta_1}$ and setting each equal to 0:

$$nb_0 + b_1 \sum_{i=1}^{n} X_i = \sum_{i=1}^{n} Y_i$$

$$b_0 \sum_{i=1}^{n} X_i + b_1 \sum_{i=1}^{n} X_i^2 = \sum_{i=1}^{n} X_i Y_i$$

Note: In matrix form: $\mathbf{X'X} = \begin{bmatrix} n & \sum_{i=1}^{n} X_i \\ \sum_{i=1}^{n} X_i & \sum_{i=1}^{n} X_i^2 \end{bmatrix}$ $\mathbf{X'Y} = \begin{bmatrix} \sum_{i=1}^{n} Y_i \\ \sum_{i=1}^{n} X_i Y_i \end{bmatrix}$ Defining $\mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$

$$\Rightarrow \quad \mathbf{X'Xb = X'Y} \quad \Rightarrow \quad \mathbf{b} = \left(\mathbf{X'X}\right)^{-1} \mathbf{X'Y}$$

Based on matrix form:

$$Q = \left(\mathbf{Y - X\beta}\right)'\left(\mathbf{Y - X\beta}\right) = \mathbf{Y'Y - Y'X\beta - \beta'X'Y + \beta'X'X\beta} =$$

$$= Y'Y - 2\left(\beta_0 \sum_{i=1}^{n} Y_i + \beta_1 \sum_{i=1}^{n} X_i Y_i\right) + n\beta_0^2 + 2\beta_0\beta_1 \sum_{i=1}^{n} X_i + \beta_1^2 \sum_{i=1}^{n} X_i^2$$

$$\frac{\partial}{\partial \boldsymbol{\beta}}(Q) = \begin{bmatrix} \frac{\partial Q}{\partial \beta_0} \\ \frac{\partial Q}{\partial \beta_1} \end{bmatrix} = \begin{bmatrix} -2\sum_{i=1}^{n} Y_i + 2n\beta_0 + 2\beta_1 \sum_{i=1}^{n} X_i \\ -2\sum_{i=1}^{n} X_i Y_i + 2\beta_0 \sum_{i=1}^{n} X_i + 2\beta_1 \sum_{i=1}^{n} X_i^2 \end{bmatrix} = -2X'Y + 2X'X\beta$$

Setting this equal to zero, and replacing β with b $\Rightarrow$ $\mathbf{X'Xb = X'Y}$

Again, the key result that we will be using repeatedly throughout multiple regression is that: $\mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'Y}$. $\mathbf{Y}$ will always be a nx1 vector of responses, and $\mathbf{X}$ will depend on what predictor variables are contained in a model, but will always have n rows.

Example: Bollywood Box Office Data

$$\left(\mathbf{X'X}\right)^{-1} = \frac{1}{55(72165.43)} \begin{bmatrix} 155976.5 & -2147 \\ -2147 & 55 \end{bmatrix} \qquad \mathbf{X'Y} = \begin{bmatrix} 2578.35 \\ 190927.5 \end{bmatrix}$$

$$\Rightarrow \quad \mathbf{b} = \frac{1}{55(72165.43)} \begin{bmatrix} 155976.5 & -2147 \\ -2147 & 55 \end{bmatrix} \begin{bmatrix} 2578.35 \\ 190927.5 \end{bmatrix}$$

$$= \frac{1}{55(72165.43)} \begin{bmatrix} 155976.5(2578.35) + (-2147)(190927.5) \\ (-2147)(2578.35) + (55)(190927.5) \end{bmatrix} = \frac{1}{3969098.65} \begin{bmatrix} -7759333.73 \\ 4965295.05 \end{bmatrix} = \begin{bmatrix} -1.9549 \\ 1.2510 \end{bmatrix}$$

Compare these estimates with those from Chapter 2.

Fitted Values and Residuals in Matrix Form

$\hat{Y}_i = b_0 + b_1 X_i \qquad e_i = Y_i - \hat{Y}_i$ In Matrix form:

$$\hat{\mathbf{Y}} = \begin{bmatrix} \hat{Y}_1 \\ \hat{Y}_2 \\ \vdots \\ \hat{Y}_n \end{bmatrix} = \begin{bmatrix} b_0 + b_1 X_1 \\ b_0 + b_1 X_2 \\ \vdots \\ b_0 + b_1 X_n \end{bmatrix} = \mathbf{Xb} = \mathbf{X}\left(\mathbf{X'X}\right)^{-1}\mathbf{X'Y} = \mathbf{HY} \qquad \mathbf{H} = \mathbf{X}\left(\mathbf{X'X}\right)^{-1}\mathbf{X'}$$

H is called the "hat" or "projection" matrix, note that **H** is idempotent (**HH = H**) and symmetric(**H = H'**):

$$\mathbf{HH} = \mathbf{X}\left(\mathbf{X'X}\right)^{-1}\mathbf{X'X}\left(\mathbf{X'X}\right)^{-1}\mathbf{X'} = \mathbf{X'I}\left(\mathbf{X'X}\right)^{-1}\mathbf{X'} = \mathbf{X}\left(\mathbf{X'X}\right)^{-1}\mathbf{X'} = \mathbf{H} \qquad \mathbf{H'} = \left(\mathbf{X}\left(\mathbf{X'X}\right)^{-1}\mathbf{X'}\right)' = \mathbf{X}\left(\mathbf{X'X}\right)^{-1}\mathbf{X'} = \mathbf{H}$$

$$\mathbf{e} = \begin{bmatrix} Y_1 - \hat{Y}_1 \\ Y_2 - \hat{Y}_2 \\ \vdots \\ Y_n - \hat{Y}_n \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} - \begin{bmatrix} \hat{Y}_1 \\ \hat{Y}_2 \\ \vdots \\ \hat{Y}_n \end{bmatrix} = \mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{Y} - \mathbf{Xb} = \mathbf{Y} - \mathbf{HY} = \mathbf{(I - H)Y}$$

Note: $\mathbf{E}\left\{\hat{\mathbf{Y}}\right\} = \mathbf{E}\left\{\mathbf{HY}\right\} = \mathbf{HE}\left\{\mathbf{Y}\right\} = \mathbf{HX\beta} = \mathbf{X}\left(\mathbf{X'X}\right)^{-1}\mathbf{X'X\beta} = \mathbf{X\beta} \qquad \boldsymbol{\sigma}^2\left\{\hat{\mathbf{Y}}\right\} = \mathbf{H}\boldsymbol{\sigma}^2\mathbf{IH'} = \sigma^2\mathbf{H}$

$\mathbf{E}\{e\} = \mathbf{E}\left\{\left(\mathbf{I}-\mathbf{H}\right)\mathbf{Y}\right\} = \left(\mathbf{I}-\mathbf{H}\right)\mathbf{E}\left\{\mathbf{Y}\right\} = \left(\mathbf{I}-\mathbf{H}\right)\mathbf{X\beta} = \mathbf{X\beta} - \mathbf{X\beta} = \mathbf{0} \qquad \boldsymbol{\sigma}^2\{\mathbf{e}\} = \left(\mathbf{I}-\mathbf{H}\right)\boldsymbol{\sigma}^2\mathbf{I}\left(\mathbf{I}-\mathbf{H}\right)' = \sigma^2\left(\mathbf{I}-\mathbf{H}\right)$

$\mathbf{s}^2\left\{\hat{\mathbf{Y}}\right\} = MSE\ \mathbf{H} \qquad \mathbf{s}^2\{\mathbf{e}\} = MSE\left(\mathbf{I}-\mathbf{H}\right)$

Note that while **H** is highly useful in matrix computations, it is *n*x*n*. Thus, we will rarely print it out (except in very small datasets), and it is very difficult to work with in EXCEL.

The important "take-away" here is that while the elements of **Y** and **ε** are assumed to be independent, the elements of $\hat{\mathbf{Y}}$ and **e** are not (**H** is not a diagonal matrix).

Analysis of Variance

Total (Corrected) Sum of Squares: $SSTO = \sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)^2 = \sum_{i=1}^{n} Y_i^2 - \dfrac{\left(\sum_{i=1}^{n} Y_i\right)^2}{n}$

Note: $\mathbf{Y'Y} = \sum_{i=1}^{n} Y_i^2$ $\qquad \dfrac{\left(\sum_{i=1}^{n} Y_i\right)^2}{n} = \left(\dfrac{1}{n}\right)\mathbf{Y'JY}$ $\qquad \underset{n\times n}{\mathbf{J}} = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}$

$\Rightarrow SSTO = \mathbf{Y'Y} - \left(\dfrac{1}{n}\right)\mathbf{Y'JY} = \mathbf{Y'}\left[\mathbf{I} - \left(\dfrac{1}{n}\right)\mathbf{J}\right]\mathbf{Y}$

$SSE = \mathbf{e'e} = \left(\mathbf{Y - Xb}\right)'\left(\mathbf{Y - Xb}\right) = \mathbf{Y'Y - Y'Xb - b'X'Y + b'X'Xb} = \mathbf{Y'Y - b'X'Y} = \mathbf{Y'}\left[\mathbf{I - H}\right]\mathbf{Y}$

since $\mathbf{b'X'Y = Y'X'}\left(\mathbf{X'X}\right)^{-1}\mathbf{X'Y = Y'HY}$

$SSR = SSTO - SSE = \mathbf{b'X'Y} - \dfrac{\left(\sum_{i=1}^{n} Y_i\right)^2}{n} = \mathbf{Y'HY} - \left(\dfrac{1}{n}\right)\mathbf{Y'JY} = \mathbf{Y'}\left[\mathbf{H} - \left(\dfrac{1}{n}\right)\mathbf{J}\right]\mathbf{Y}$

Note that *SSTO*, *SSR*, *and SSE* are all QUADRATIC FORMS: $\mathbf{Y'AY}$ for symmetric, idempotent matrices $\mathbf{A}$

Example: Bollywood Box Office Data

$\sum Y_i^2 = \mathbf{Y'IY} = 304471.84$

$\sum Y_i = 2578.3 \quad \Rightarrow \quad \dfrac{1}{n}\mathbf{Y'JY} = \left(\dfrac{2578.3^2}{55}\right) = 120866.02$

$\mathbf{b} = \begin{bmatrix} -1.9549 \\ 1.2510 \end{bmatrix} \quad \mathbf{X'Y} = \begin{bmatrix} 2578.3 \\ 190927.5 \end{bmatrix} \quad \Rightarrow \quad \mathbf{b'X'Y} = \begin{bmatrix} -1.9549 & 1.2510 \end{bmatrix}\begin{bmatrix} 2578.3 \\ 190927.5 \end{bmatrix} = 233809.98 = \mathbf{Y'HY}$

$SSTO = \mathbf{Y'}\left(\mathbf{I} - \dfrac{1}{n}\mathbf{J}\right)\mathbf{Y} = 304471.84 - 120866.02 = 183605.82$

$SSE = \mathbf{Y'}\left(\mathbf{I} - \mathbf{H}\right)\mathbf{Y} = 304471.84 - 233809.98 = 70661.86$

$SSR = \mathbf{Y'}\left(\mathbf{H} - \dfrac{1}{n}\mathbf{J}\right)\mathbf{Y} = 233809.98 - 120866.02 = 112943.96$

Inferences in Linear Regression

$$\mathbf{b} = \left(\mathbf{X'X}\right)^{-1}\mathbf{X'Y} \quad \text{Þ} \quad \mathbf{E}\left\{\mathbf{b}\right\} = \left(\mathbf{X'X}\right)^{-1}\mathbf{X'E}\left\{\mathbf{Y}\right\} = \left(\mathbf{X'X}\right)^{-1}\mathbf{X'X}\boldsymbol{\beta} = \boldsymbol{\beta}$$

$$\boldsymbol{\sigma}^2\left\{\mathbf{b}\right\} = \left(\mathbf{X'X}\right)^{-1}\mathbf{X'}\boldsymbol{\sigma}^2\left\{\mathbf{Y}\right\}\mathbf{X}\left(\mathbf{X'X}\right)^{-1} = \boldsymbol{\sigma}^2\left(\mathbf{X'X}\right)^{-1}\mathbf{X'IX}\left(\mathbf{X'X}\right)^{-1} = \boldsymbol{\sigma}^2\left(\mathbf{X'X}\right)^{-1} \qquad \mathbf{s}^2\left\{\mathbf{b}\right\} = MSE\left(\mathbf{X'X}\right)^{-1}$$

Recall: $$\left(\mathbf{X'X}\right)^{-1} = \begin{bmatrix} \frac{1}{n} + \frac{\overline{X}^2}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} & -\frac{\overline{X}}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} \\ -\frac{\overline{X}}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} & \frac{1}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} \end{bmatrix} \quad \Rightarrow$$

$$\mathbf{s}^2\left\{\mathbf{b}\right\} = \begin{bmatrix} \frac{MSE}{n} + \frac{\overline{X}^2 MSE}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} & -\frac{\overline{X}MSE}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} \\ -\frac{\overline{X}MSE}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} & \frac{MSE}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} \end{bmatrix} = MSE\begin{bmatrix} \frac{1}{n} + \frac{\overline{X}^2}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} & -\frac{\overline{X}}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} \\ -\frac{\overline{X}}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} & \frac{1}{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2} \end{bmatrix}$$

Estimated Mean Response at $X = X_h$:

$$\hat{Y}_h = b_0 + b_1 X_h = \mathbf{X_h'b} \qquad \mathbf{X_h} = \begin{bmatrix} 1 \\ X_h \end{bmatrix} \qquad s^2\left\{\hat{Y}_h\right\} = \mathbf{X_h'}\mathbf{s}^2\left\{\mathbf{b}\right\}\mathbf{X_h} = MSE\left(\mathbf{X_h'}\left(\mathbf{X'X}\right)^{-1}\mathbf{X_h}\right)$$

Predicted New Response at $X = X_h$:

$$\hat{Y}_h = b_0 + b_1 X_h = \mathbf{X_h'b} \qquad s^2\left\{\text{pred}\right\} = MSE\left(1 + \mathbf{X_h'}\left(\mathbf{X'X}\right)^{-1}\mathbf{X_h}\right)$$

Example: Bollywood Box Office Data

$$\left(\mathbf{X'X}\right)^{-1} = \frac{1}{55(72165.43)}\begin{bmatrix} 155976.5 & -2147 \\ -2147 & 55 \end{bmatrix} \qquad SSE = 70661.86 \quad \Rightarrow \quad MSE = \frac{70661.86}{55-2} = 1333.24$$

$$\Rightarrow \quad s^2\left\{\mathbf{b}\right\} = \frac{1333.24}{55(72165.43)}\begin{bmatrix} 155976.5 & -2147 \\ -2147 & 55 \end{bmatrix} = \begin{bmatrix} 52.3933 & -0.7212 \\ -0.7212 & 0.0185 \end{bmatrix}$$

$$\Rightarrow \quad s\left\{b_0\right\} = \sqrt{52.3933} = 7.2383 \quad s\left\{b_1\right\} = \sqrt{0.0185} = 0.1360$$

Compare these with the results from Chapter 2.

R Program for Matrix Computations – Bollywood Data

```
bbo <-
read.csv("http://www.stat.ufl.edu/~winner/sta4210/mydata/bollywood_boxoffice.csv",
header=T)
attach(bbo)
names(bbo)

Y <- Budget;   X <- Gross
n <- length(Y)
X0 <- rep(1,n)                  # Column of 1's for X matrix
X <- as.matrix(cbind(X0,X))     # Form the X-matrix (n=55 rows, 2 Cols)
Y <- as.matrix(Y,ncol=1)        # Form the Y-vector (n=55 rows, 1 col)

# Notes:   t(X) = transpose of X, %*% = matrix multiplication, solve(A) = A^(-1)
(XX <- t(X) %*% X)         # Obtain X'X matrix (2 rows, 2 cols)
(XY <- t(X) %*% Y)         # Obtain X'Y vector (2 rows, 1 col)
(XXI <- solve(XX))         # Obtain (X'X)^(-1) matrix (2 rows, 2 cols)
(b <- XXI %*% XY)          # Obtain b-vector (2 rows, 1 col)
Y_hat <- X %*% b           # Obtain the vector of fitted values (n=55 rows, 1 col)
e <- Y - Y_hat             # Obtain the vector of residuals (n=55 rows, 1 col)
print(cbind(Y_hat,e))
H <- X %*% XXI %*% t(X)           # Obtain the Hat matrix
J_n <- matrix(rep(1/n,n^2),ncol=n)   # Obtain the (1/n)J matrix (n=55 rows, n=55 cols)
I_n <- diag(n)                    # Obtain the identity matrix (n=55 rows, n=55 cols)
(SSTO <- t(Y) %*% (I_n - J_n) %*% Y) # Obtain Total Sum of Squares (SSTO)
# SSTO can also be computed as:   (SSTO <- (t(Y) %*% Y) - (t(Y) %*% (I_n - J_n) %*% Y))
(SSE <- t(Y) %*% (I_n - H) %*% Y)    # Obtain Error Sum of Squares (SSE)
# SSE can also be computed as:   (SSE <- (t(Y) %*% Y) - (t(b) %*% XY))
(SSR <- t(Y) %*% (H - J_n) %*% Y)    # Obtain Regression Sum of Squares (SSR)
# SSR can also be computed as:   (SSR <- (t(b) %*% XY) - (t(Y) %*% J_n %*% Y))
(MSE <- SSE/(n-2))                # Obtain MSE = s^2
(s2_b <- MSE[1,1] * XXI)          # Obtain s^2{b}, must use MSE[1,1] and * to do scalar multiplication
(X_h <- matrix(c(1,20),ncol=1))      # Create X_h vector, for case where budget=20
(Y_hat_h <- t(X_h) %*% b)         # Obtain the fitted value when budget=20
(s2_yhat_h <- t(X_h) %*% s2_b %*% X_h)       # Obtain s^2{Y_hat_h}
(s2_pred <- MSE + (t(X_h) %*% s2_b %*% X_h))  # Obtain s^2{pred}
```

R Output for Matrix Computations – Bollywood Data

```
> (XX <- t(X) %*% X)              # Obtain X'X matrix (2 rows, 2 cols)
        X0         X
X0   55.00   2578.35
X  2578.35 304471.84
>
> (XY <- t(X) %*% Y)              # Obtain X'Y vector (2 rows, 1 col)
       [,1]
X0   2147.0
X  190927.5
>
> (XXI <- solve(XX))              # Obtain (X'X)^(-1) matrix (2 rows, 2 cols)
               X0             X
X0  0.0301515111 -2.553312e-04
X  -0.0002553312  5.446590e-06
>
> (b <- XXI %*% XY)               # Obtain b-vector (2 rows, 1 col)
          [,1]
X0 15.9855609
X   0.4917075
>
> Y_hat <- X %*% b                # Obtain the vector of fitted values (n=55 rows, 1
col)
>
> e <- Y - Y_hat                  # Obtain the vector of residuals (n=55 rows, 1 col)
>
> print(cbind(Y_hat,e))
           [,1]        [,2]
 [1,]  63.01247 -27.0124706
 [2,]  43.34909  33.6509142
 …
[54,]  63.35667   1.6433342
[55,]  54.54527 -44.0452666
>
```

Continued Below

```
>
> H <- X %*% XXI %*% t(X)              # Obtain the Hat matrix
>
> J_n <- matrix(rep(1/n,n^2),ncol=n)   # Obtain the (1/n)J matrix (n=55 rows, n=55
cols)
>
> I_n <- diag(n)                       # Obtain the identity matrix (n=55 rows, n=55
cols)
>
> (SSTO <- t(Y) %*% (I_n - J_n) %*% Y) # Obtain Total Sum of Squares (SSTO)
          [,1]
[1,] 72165.43
> # SSTO can also be computed as:
> # (SSTO <- (t(Y) %*% Y) - (t(Y) %*% (I_n - J_n) %*% Y))
>
> (SSE <- t(Y) %*% (I_n - H) %*% Y)    # Obtain Error Sum of Squares (SSE)
          [,1]
[1,] 27775.02
> # SSE can also be computed as:
> # (SSE <- (t(Y) %*% Y) - (t(b) %*% XY))
>
> (SSR <- t(Y) %*% (H - J_n) %*% Y)    # Obtain Regression Sum of Squares (SSR)
         [,1]
[1,] 44390.4
> # SSR can also be computed as:
> # (SSR <- (t(b) %*% XY) - (t(Y) %*% J_n %*% Y))
>
> (MSE <- SSE/(n-2))                   # Obtain MSE = s^2
         [,1]
[1,] 524.057
>
> (s2_b <- MSE[1,1] * XXI)             # Obtain s^2{b}, must use MSE[1,1] and * to do
scalar multiplication
           X0            X
X0 15.8011116 -0.133808095
X  -0.1338081  0.002854324
>
> (X_h <- matrix(c(1,20),ncol=1))      # Create X_h vector, for case where budget=20
     [,1]
[1,]    1
[2,]   20
>
> (Y_hat_h <- t(X_h) %*% b)            # Obtain the fitted value when budget=20
          [,1]
[1,] 25.81971
>
> (s2_yhat_h <- t(X_h) %*% s2_b %*% X_h)        # Obtain s^2{Y_hat_h}
          [,1]
[1,] 11.59052
>
> (s2_pred <- MSE + (t(X_h) %*% s2_b %*% X_h))  # Obtain s^2{pred}
          [,1]
[1,] 535.6476
```

Chapter 6 – Multiple Regression I

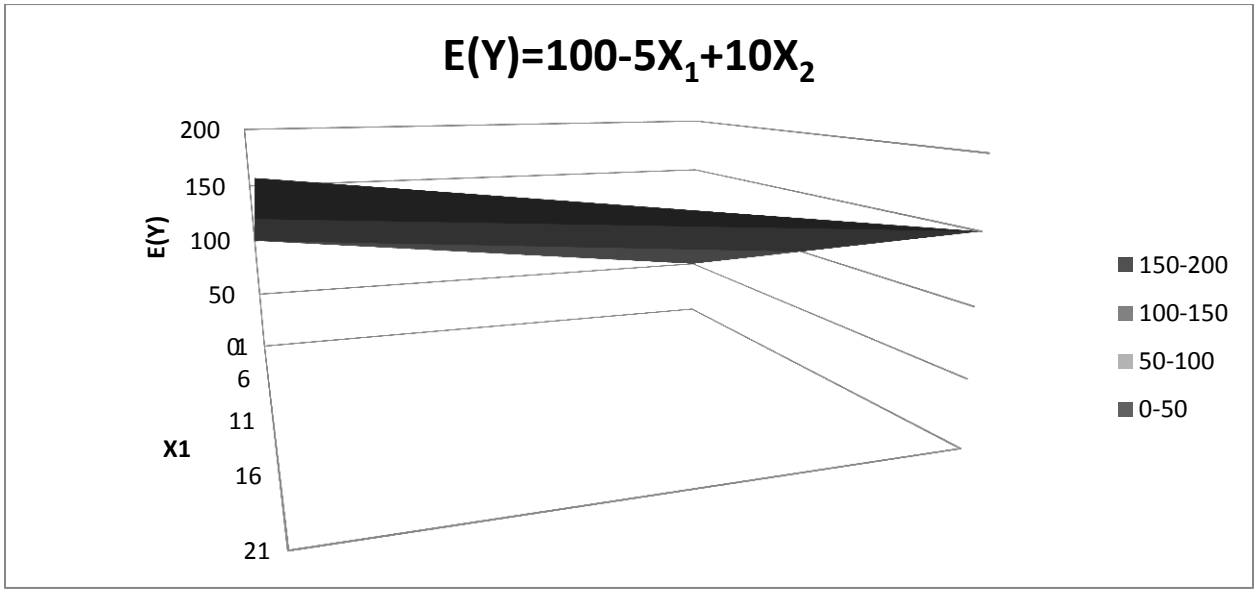
Models with Multiple Predictors

- Most Practical Problems have more than one potential predictor variable
- Goal is to determine effects (if any) of each predictor, controlling for others
- Can include polynomial terms to allow for nonlinear relations
- Can include product terms to allow for interactions when effect of one variable depends on level of another variable
- Can include "dummy" variables for categorical predictors

First-Order Model with 2 Predictors

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i$$

$$E\{\varepsilon_i\} = 0 \quad \Rightarrow \quad E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \quad \text{Plane in 3-dimensions}$$



Interpretation of Regression Coefficients

- Additive: $E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \equiv$ Mean of Y @ X_1, X_2
- $\beta_0 \equiv$ Intercept, Mean of Y when $X_1 = X_2 = 0$
- $\beta_1 \equiv$ Slope with Respect to X_1 (effect of increasing X_1 by 1 unit, while holding X_2 constant)
- $\beta_2 \equiv$ Slope with Respect to X_2 (effect of increasing X_2 by 1 unit, while holding X_1 constant)

- These can also be obtained by taking the partial derivatives of $E\{Y\}$ with respect to X_1 and X_2, respectively
- Interaction Model: $E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2$
- When $X_2 = 0$: Effect of increasing X_1 by 1: $\beta_1(1)+\beta_3(1)(0)= \beta_1$
- When $X_2 = 1$: Effect of increasing X_1 by 1: $\beta_1(1)+\beta_3(1)(1)= \beta_1+\beta_3$
- The effect of increasing X_1 depends on level of X_2, and vice versa

General Linear Regression Model with p-1 Predictor Variables

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_{p-1} X_{i,p-1} + \varepsilon_i$$

$$\Rightarrow \quad Y_i = \beta_0 + \sum_{k=1}^{p-1} \beta_k X_{ik} + \varepsilon_i$$

$$\Rightarrow \quad Y_i = \sum_{k=0}^{p-1} \beta_k X_{ik} + \varepsilon_i \qquad \text{where: } X_{i0} \equiv 1$$

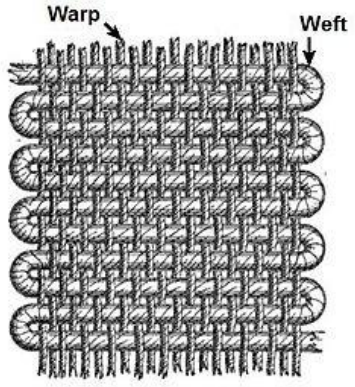
$$E\{\varepsilon_i\} = 0 \quad \Rightarrow \quad E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_{p-1} X_{p-1} \quad \text{(Hyperplane in } p\text{-dimensions)}$$

$$p - 1 = 1 \quad \Rightarrow \quad \text{Simple linear regression}$$

Normality, independence, and constant variance for errors:

$$\varepsilon_i \sim NID\left(0, \sigma^2\right) \quad \Rightarrow \quad Y_i \sim N\left(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_{p-1} X_{i,p-1}, \sigma^2\right) \quad \sigma\{Y_i, Y_j\} = 0 \quad \forall\, i \neq j$$

Example: Factors Effecting Air Permeability of Woven Fabrics



$Y \equiv$ Average air permeability $\left(\text{cm}^3/\text{s}/\text{cm}^2\right)$

$X_1 \equiv$ Warp Yarn Density (ends/cm)

$X_2 \equiv$ Weft Yarn Density (picks/cm)

$X_3 \equiv$ Mass per Unit Area $\left(\text{grams}/\text{cm}^2\right)$

Graphic Source: Wikipedia

Data Source: A. Cai, S. Vassiliadis, M. Rangoussi, I. Tarakcioglu (2007). "Prediction of the Air Permeability of Woven Fabrics Using Neural Networks," *International Journal of Clothing Science and Technology*, Vol. 19, #1, pp. 18-35

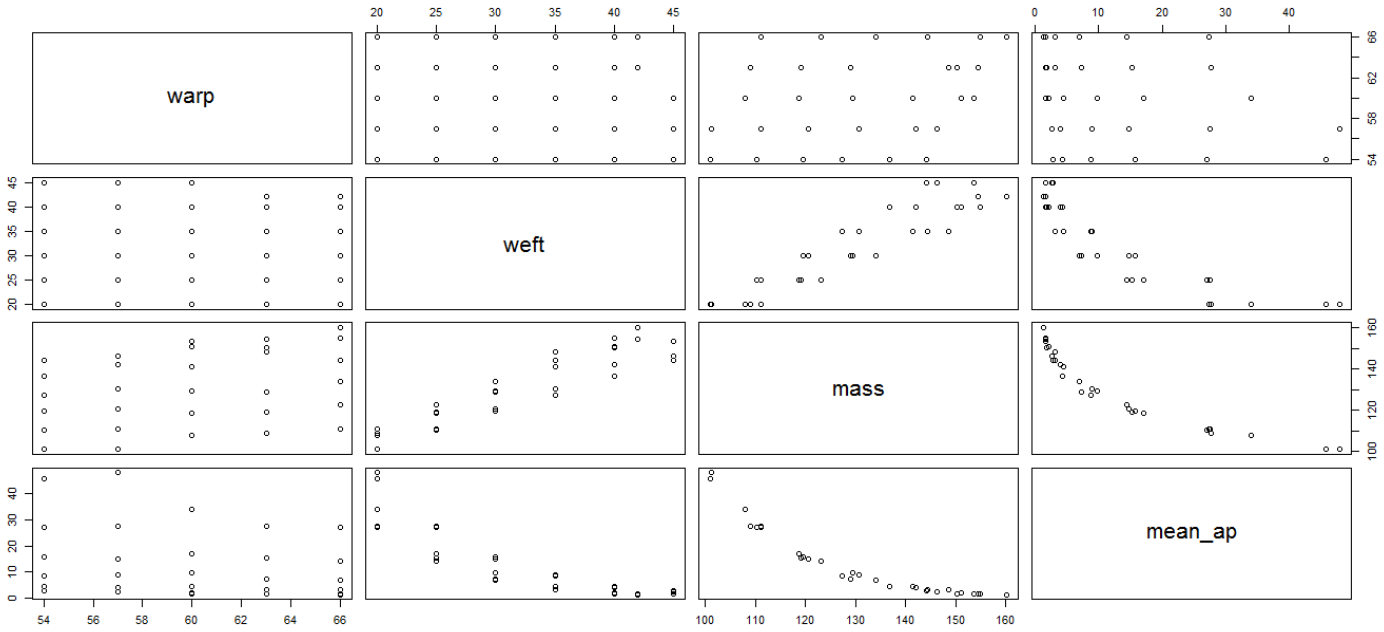
ID	warp	weft	mass	mean_ap	sd_ap
1	54	20	101.0	45.74	1.80
2	54	25	110.3	27.02	2.44
3	54	30	119.5	15.68	0.68
4	54	35	127.3	8.76	0.39
5	54	40	136.7	4.37	0.15
6	54	45	144.1	2.90	0.40
7	57	20	101.2	47.94	1.83
8	57	25	111.1	27.52	0.78
9	57	30	120.5	14.84	0.97
10	57	35	130.7	8.99	0.48
11	57	40	142.0	3.98	0.24
12	57	45	146.3	2.68	0.12
13	60	20	107.9	33.98	1.16
14	60	25	118.7	17.01	0.79
15	60	30	129.5	9.75	0.41
16	60	35	141.4	4.56	0.41
17	60	40	151.2	2.12	0.07
18	60	45	153.7	1.70	0.05
19	63	20	109.0	27.68	0.55
20	63	25	119.1	15.24	0.64
21	63	30	129.1	7.23	0.13
22	63	35	148.5	3.22	0.42
23	63	40	150.3	1.89	0.08
24	63	42	154.4	1.61	0.05
25	66	20	111.2	27.28	1.68
26	66	25	123.0	14.42	1.39
27	66	30	134.0	6.90	0.21
28	66	35	144.4	3.19	0.17
29	66	40	155.0	1.65	0.06
30	66	42	160.2	1.38	0.05

Data represent the mean of each of 5 assessments of air permeability. The standard deviation (sd_ap) is not used now, the response, Y, is mean_ap. The standard deviation will be used later for weighted least squares.

SUMMARY OUTPUT						
Regression Statistics						
Multiple R	0.916755					
R Square	0.840439					
Adjusted R Squar	0.822028					
Standard Error	5.593951					
Observations	30					
ANOVA						
	df	*SS*	*MS*	*F*	*gnificance F*	
Regression	3	4285.382	1428.461	45.64898	1.68E-10	
Residual	26	813.5993	31.29228			
Total	29	5098.981				
	Coefficients	*andard Err*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	105.0349	16.35453	6.422376	8.36E-07	71.41772	138.6521
warp	-0.39211	0.569341	-0.68871	0.497104	-1.56241	0.778185
weft	-0.67957	0.727994	-0.93349	0.359158	-2.17599	0.816841
mass	-0.35497	0.365511	-0.97117	0.340412	-1.10629	0.396344

All numbers on this table are reproduced below. The Standard Error in the Regression Statistics portion is $s = \sqrt{MSE}$

Scatterplot Matrix



Note that the bottom row plots *Y* versus each of the *X* variables. Air permeability tends to decrease as each predictor increases. We will see later that the overall model is a good fit, while individual coefficients are not significant (due to the strong correlation between weft and mass). The relationships between air permeability and weft and mass appear to be nonlinear.

Special Types of Variables/Models

- *p*-1 distinct numeric predictors (attributes)
  - Y = Sales, X_1=Advertising, X_2=Price
- Categorical Predictors – Indicator (Dummy) variables, representing *m*-1 levels of a *m* level categorical variable
  - Y = Salary, X_1=Experience, X_2=1 if College Grad, 0 if Not
- Polynomial Terms – Allow for bends in the Regression
  - Y=MPG, X_1=Speed, X_2=Speed^2
- Transformed Variables – Transformed Y variable to achieve linearity
  - $Y'=\ln(Y)$ $Y'=1/Y$
- Interaction Effects – Effect of one predictor depends on levels of other predictors
  - Y = Salary, X_1=Experience, X_2=1 if Coll Grad, 0 if Not, $X_3=X_1X_2$
  - $E\{Y\} = \beta_0 + \beta_1X_1 + \beta_2X_2 + \beta_3X_1X_2$

- Non-College Grads (X_2 =0):
- $E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2(0) + \beta_3 X_1(0) = \beta_0 + \beta_1 X_1$
- College Grads (X_2 =1):
- $E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2(1) + \beta_3 X_1(1) = (\beta_0 + \beta_2) + (\beta_1 + \beta_3) X_1$

- Response Surface Models
  - $E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2 + \beta_5 X_1 X_2$
- Note: Although the Response Surface Model has polynomial terms, it is linear with respect to the Regression parameters

Matrix Form of Multiple Regression Model

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_{p-1} X_{i,p-1} + \varepsilon_i \quad i = 1, \ldots, n$$

Matrix Form:

$$\underset{n\times 1}{\mathbf{Y}} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \qquad \underset{n\times p}{\mathbf{X}} = \begin{bmatrix} 1 & X_{11} & X_{12} & \cdots & X_{1,p-1} \\ 1 & X_{21} & X_{22} & \cdots & X_{2,p-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & X_{n1} & X_{n2} & \cdots & X_{n,p-1} \end{bmatrix}$$

$$\underset{p\times 1}{\boldsymbol{\beta}} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix} \qquad \underset{n\times 1}{\boldsymbol{\varepsilon}} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix} \qquad \mathbf{E}\left\{\underset{\mathbf{n\times 1}}{\boldsymbol{\varepsilon}}\right\} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \boldsymbol{\sigma}^{\mathbf{2}}\left\{\underset{n\times 1}{\boldsymbol{\varepsilon}}\right\} = \begin{bmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2 \end{bmatrix} = \sigma^2\mathbf{I}$$

$$\underset{n\times 1}{\mathbf{Y}} = \underset{n\times p}{\mathbf{X}}\ \underset{p\times 1}{\boldsymbol{\beta}} + \underset{n\times 1}{\boldsymbol{\varepsilon}} \quad \Rightarrow \quad \mathbf{E}\left\{\underset{\mathbf{n\times 1}}{\mathbf{Y}}\right\} = \mathbf{E}\left\{\underset{\mathbf{n\times p}}{\mathbf{X}}\ \underset{\mathbf{p\times 1}}{\boldsymbol{\beta}} + \underset{\mathbf{n\times 1}}{\boldsymbol{\varepsilon}}\right\} = \underset{n\times p}{\mathbf{X}}\ \underset{p\times 1}{\boldsymbol{\beta}} \qquad \boldsymbol{\sigma}^{\mathbf{2}}\left\{\underset{n\times 1}{\mathbf{Y}}\right\} = \sigma^2\mathbf{I}$$

Note that simple linear regression is the special case where p-1 = 1.

Least Squares Estimation of Regression Coefficients

Goal: Minimize: $Q = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} \left(Y_i - \beta_0 - \beta_1 X_{i1} - ... - \beta_{p-1} X_{i,p-1} \right)^2$

$\Rightarrow$ Obtain Estimates of $\beta_0, \beta_1, ..., \beta_{p-1}$ that minimize $Q \quad \Rightarrow \quad b_0, b_1, ..., b_{p-1}$

Normal Equations: $\underset{p\times p}{\mathbf{X'X}} \; \underset{p\times 1}{\mathbf{b}} = \underset{p\times 1}{\mathbf{X'Y}} \quad \Rightarrow \quad \underset{p\times 1}{\mathbf{b}} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{p-1} \end{bmatrix} = \left(\mathbf{X'X} \right)^{-1} \mathbf{X'Y}$

Maximum Likelihood also leads to the same estimator $\mathbf{b}$:

$$L\left(\boldsymbol{\beta}, \sigma^2 \right) = \left(2\pi\sigma^2 \right)^{-n/2} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(Y_i - \beta_0 - \beta_1 X_{i1} - ... - \beta_{p-1} X_{i,p-1} \right)^2 \right]$$

since maximizing L involves minimizing $\sum_{i=1}^{n} \left(Y_i - \beta_0 - \beta_1 X_{i1} - ... - \beta_{p-1} X_{i,p-1} \right)^2$

Example: Factors Effecting Air Permeability of Woven Fabrics

X'X					X'Y
30	1800	969	3931.3		391.23
1800	108540	58113	236587.2		23028.66
969	58113	33353	130975.2		9833.43
3931.3	236587.2	130975.2	524246		45054.13
INV(X'X)					b
8.547494	-0.21469	-0.16004	0.072775		105.0349
-0.21469	0.010359	0.011853	-0.00603		-0.39211
-0.16004	0.011853	0.016936	-0.00838		-0.67957
0.072775	-0.00603	-0.00838	0.004269		-0.35497

Note that the elements of **X'X** and **X'Y** are:

$$\mathbf{X'X} = \begin{bmatrix} n & \sum X_{i1} & \sum X_{i2} & \sum X_{i3} \\ \sum X_{i1} & \sum X_{i1}^2 & \sum X_{i1}X_{i2} & \sum X_{i1}X_{i3} \\ \sum X_{i2} & \sum X_{i1}X_{i2} & \sum X_{i2}^2 & \sum X_{i2}X_{i3} \\ \sum X_{i3} & \sum X_{i1}X_{i3} & \sum X_{i2}X_{i3} & \sum X_{i3}^2 \end{bmatrix} \qquad \mathbf{X'Y} = \begin{bmatrix} \sum Y_i \\ \sum X_{i1}Y_i \\ \sum X_{i2}Y_i \\ \sum X_{i3}Y_i \end{bmatrix}$$

Fitted Values and Residuals

Fitted Values: $\underset{n\times 1}{\hat{\mathbf{Y}}} = \begin{bmatrix} \hat{Y}_1 \\ \hat{Y}_2 \\ \vdots \\ \hat{Y}_n \end{bmatrix}$ Residuals: $\underset{n\times 1}{\mathbf{e}} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$

$$\underset{n\times 1}{\hat{\mathbf{Y}}} = \underset{n\times p}{\mathbf{X}}\ \underset{p\times 1}{\mathbf{b}} = \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y} = \mathbf{H}\mathbf{Y} \qquad \mathbf{H} = \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}' \quad \mathbf{H} = \mathbf{H}' = \mathbf{H}\mathbf{H}$$

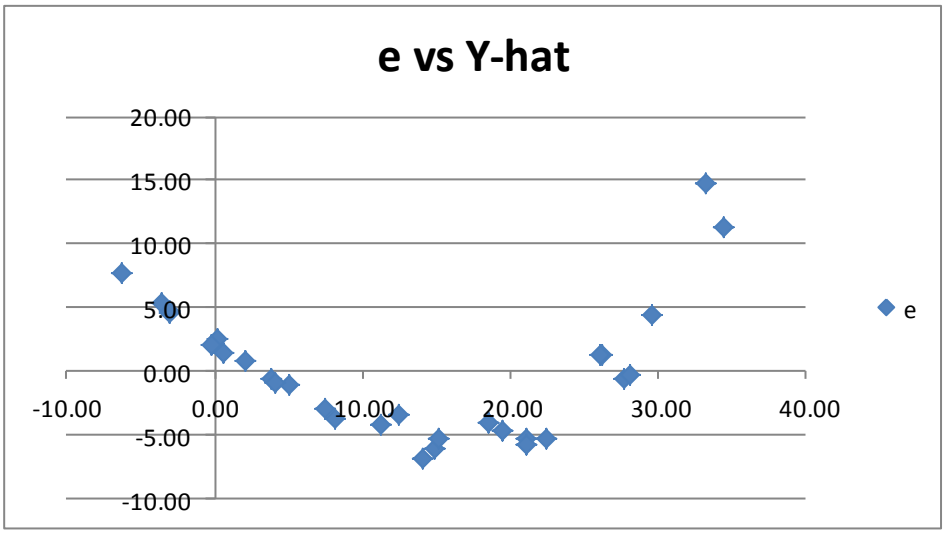
$$\underset{n\times 1}{\mathbf{e}} = \underset{n\times 1}{\mathbf{Y}} - \underset{n\times 1}{\hat{\mathbf{Y}}} = \underset{n\times 1}{\mathbf{Y}} - \underset{n\times p}{\mathbf{X}}\ \underset{p\times 1}{\mathbf{b}} = \mathbf{Y} - \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y} = \left(\mathbf{I}-\mathbf{H}\right)\mathbf{Y} \qquad \left(\mathbf{I}-\mathbf{H}\right) = \left(\mathbf{I}-\mathbf{H}\right)' = \left(\mathbf{I}-\mathbf{H}\right)\left(\mathbf{I}-\mathbf{H}\right)$$

$$\boldsymbol{\sigma}^2\left\{\hat{\mathbf{Y}}\right\} = \boldsymbol{\sigma}^2\left\{\mathbf{H}\mathbf{Y}\right\} = \mathbf{H}\boldsymbol{\sigma}^2\left\{\mathbf{Y}\right\}\mathbf{H}' = \boldsymbol{\sigma}^2\mathbf{H} \qquad \mathbf{s}^2\left\{\hat{\mathbf{Y}}\right\} = MSE\left(\mathbf{H}\right)$$

$$\boldsymbol{\sigma}^2\left\{\mathbf{e}\right\} = \boldsymbol{\sigma}^2\left\{\left(\mathbf{I}-\mathbf{H}\right)\mathbf{Y}\right\} = \left(\mathbf{I}-\mathbf{H}\right)\boldsymbol{\sigma}^2\left\{\mathbf{Y}\right\}\left(\mathbf{I}-\mathbf{H}\right)' = \boldsymbol{\sigma}^2\left(\mathbf{I}-\mathbf{H}\right) \qquad \mathbf{s}^2\left\{\mathbf{e}\right\} = MSE\left(\mathbf{I}-\mathbf{H}\right)$$

Example: Factors Effecting Air Permeability of Woven Fabrics

Y-hat	e
34.42	11.32
27.72	-0.70
21.05	-5.37
14.89	-6.13
8.15	-3.78
2.13	0.77
33.17	14.77
26.26	1.26
19.52	-4.68
12.50	-3.51
5.10	-1.12
0.17	2.51
29.62	4.36
22.38	-5.37
15.15	-5.40
7.53	-2.97
0.65	1.47
-3.63	5.33
28.05	-0.37
21.07	-5.83
14.12	-6.89
3.83	-0.61
-0.20	2.09
-3.02	4.63
26.09	1.19
18.50	-4.08
11.20	-4.30
4.11	-0.92
-3.05	4.70
-6.25	7.63



Analysis of Variance – Sums of Squares

$$\underset{n\times 1}{\mathbf{Y}}=\begin{bmatrix}Y_1\\Y_2\\\vdots\\Y_n\end{bmatrix}\qquad \underset{n\times 1}{\hat{\mathbf{Y}}}=\begin{bmatrix}\hat{Y}_1\\\hat{Y}_2\\\vdots\\\hat{Y}_n\end{bmatrix}=\mathbf{Xb}=\mathbf{HY}\qquad \underset{n\times 1}{\overline{\mathbf{Y}}}=\begin{bmatrix}\overline{Y}\\\overline{Y}\\\vdots\\\overline{Y}\end{bmatrix}=\frac{1}{n}\begin{bmatrix}1&\cdots&1\\\vdots&\ddots&\vdots\\1&\cdots&1\end{bmatrix}\begin{bmatrix}Y_1\\Y_2\\\vdots\\Y_n\end{bmatrix}=\left(\frac{1}{n}\right)\mathbf{JY}$$

$$SSTO=\sum_{i=1}^{n}\left(Y_i-\overline{Y}\right)^2=\left(\mathbf{Y}-\overline{\mathbf{Y}}\right)'\left(\mathbf{Y}-\overline{\mathbf{Y}}\right)=\mathbf{Y}'\left(\mathbf{I}-\left(\frac{1}{n}\right)\mathbf{J}\right)\mathbf{Y}$$

$$SSE=\sum_{i=1}^{n}\left(Y_i-\hat{Y}_i\right)^2=\left(\mathbf{Y}-\hat{\mathbf{Y}}\right)'\left(\mathbf{Y}-\hat{\mathbf{Y}}\right)=\mathbf{Y}'(\mathbf{I}-\mathbf{H})\mathbf{Y}=\mathbf{Y}'\mathbf{Y}-\mathbf{b}'\mathbf{X}'\mathbf{Y}\qquad MSE=\frac{SSE}{n-p}$$

$$SSR=\sum_{i=1}^{n}\left(\hat{Y}_i-\overline{Y}\right)^2=\left(\hat{\mathbf{Y}}-\overline{\mathbf{Y}}\right)'\left(\hat{\mathbf{Y}}-\overline{\mathbf{Y}}\right)=\mathbf{Y}'\left(\mathbf{H}-\left(\frac{1}{n}\right)\mathbf{J}\right)\mathbf{Y}=\mathbf{b}'\mathbf{X}'\mathbf{Y}-\mathbf{Y}'\left(\frac{1}{n}\right)\mathbf{JY}\qquad MSR=\frac{SSR}{p-1}$$

$$E\{MSE\}=\sigma^2$$

$$E\{MSR\}=\sigma^2+\sum_{k=1}^{p-1}\beta_k^2SS_{kk}+\sum_{k=1}^{p-1}\sum_{k'\neq k}\beta_k\beta_{k'}SS_{kk'}\qquad SS_{kk'}=\sum_{i=1}^{n}\left(X_{ik}-\overline{X}_k\right)\left(X_{ik'}-\overline{X}_{k'}\right)$$

$$E\{MSR\}\geq E\{MSE\}\qquad E\{MSR\}\geq E\{MSE\}\quad\Leftrightarrow\quad\beta_1=\ldots=\beta_{p-1}=0$$

Example: Factors Effecting Air Permeability of Woven Fabrics

$$\sum Y_i^2=\mathbf{Y}'\mathbf{IY}=10201.01$$

$$\sum Y_i=391.23\quad\Rightarrow\quad\frac{1}{n}\mathbf{Y}'\mathbf{JY}=\left(\frac{391.23^2}{30}\right)=5102.03$$

$$\mathbf{b}=\begin{bmatrix}104.0349\\-0.39211\\-0.67957\\-0.35497\end{bmatrix}\qquad\mathbf{X}'\mathbf{Y}=\begin{bmatrix}391.23\\23028.66\\9833.43\\45054.13\end{bmatrix}$$

$$\Rightarrow\quad\mathbf{b}'\mathbf{X}'\mathbf{Y}=\begin{bmatrix}105.0349&-0.39211&-0.67957&-0.35497\end{bmatrix}\begin{bmatrix}391.23\\23028.66\\9833.43\\45054.13\end{bmatrix}=9387.41=\mathbf{Y}'\mathbf{HY}$$

$$SSTO=\mathbf{Y}'\left(\mathbf{I}-\frac{1}{n}\mathbf{J}\right)\mathbf{Y}=10201.01-5102.03=5098.98$$

$$SSE=\mathbf{Y}'(\mathbf{I}-\mathbf{H})\mathbf{Y}=10201.01-9387.41=813.60\qquad MSE=\frac{813.60}{30-4}=31.29$$

$$SSR=\mathbf{Y}'\left(\mathbf{H}-\frac{1}{n}\mathbf{J}\right)\mathbf{Y}=9387.41-5102.03=4285.38\qquad MSR=\frac{4285.38}{3}=1428.46$$

ANOVA Table, F-test, and R^2

Analysis of Variance (ANOVA) Table

Source	df	Sum of Squares	Mean Square
Regression	$p-1$	$SSR = \mathbf{b'X'Y} - \mathbf{Y'}\left(\frac{1}{n}\right)\mathbf{JY}$	$MSR = \frac{SSR}{p-1}$
Error	$n-p$	$SSE = \mathbf{Y'Y} - \mathbf{b'X'Y}$	$MSE = \frac{SSE}{n-p}$
Total	$n-1$	$SSTO = \mathbf{Y'Y} - \mathbf{Y'}\left(\frac{1}{n}\right)\mathbf{JY}$	

Test of $H_0 : \beta_1 = ... = \beta_{p-1} = 0 \quad \left(E(Y) = \beta_0\right) \quad H_A$: Not all $\beta_k = 0$

Test Statistic: $F^* = \frac{MSR}{MSE}$ Rejection Region: $F^* \geq F(1-\alpha; p-1, n-p)$ P-value= $\Pr\left\{F(p-1, n-p) \geq F^*\right\}$

Coefficient of Multiple Determination: $R^2 = \frac{SSR}{SSTO} = 1 - \frac{SSE}{SSTO}$ Correlation: $R = \sqrt{R^2}$

Adjusted-$R^2 = 1 - \frac{\left[\frac{SSE}{n-p}\right]}{\left[\frac{SSTO}{n-1}\right]} = 1 - \left(\frac{n-1}{n-p}\right)\frac{SSE}{SSTO}$ places a "penalty" on models with extra predictors

Example: Factors Effecting Air Permeability of Woven Fabrics

$$H_0 : \beta_1 = \beta_2 = \beta_3 = 0 \quad H_A: \text{ Not all } \beta_k = 0$$

$$TS : F^* = \frac{1428.46}{31.29} = 45.65 \quad RR : F^* \geq F(.95; 3, 26) = 2.975$$

$$P - \text{value: } P\left(F(3, 26) \geq 45.65\right) = .0000$$

$$R^2 = \frac{4285.38}{5098.98} = 0.8404 \qquad R = \sqrt{.8404} = .9168$$

$$\text{Adjusted-}R^2 = 1 - \left(\frac{30-1}{30-4}\right)\left(\frac{813.60}{5098.98}\right) = 1 - .1780 = .8220$$

Inferences Regarding Regression Parameters

$\mathbf{Y} = \mathbf{X\beta} + \boldsymbol{\varepsilon} \qquad \mathbf{E}\{\boldsymbol{\varepsilon}\} = \mathbf{0} \qquad \boldsymbol{\sigma}^2\{\boldsymbol{\varepsilon}\} = \sigma^2\mathbf{I} \quad \Rightarrow \quad \mathbf{E}\{\mathbf{Y}\} = \mathbf{X\beta} \qquad \boldsymbol{\sigma}^2\{\mathbf{Y}\} = \sigma^2\mathbf{I}$

$$\mathbf{E}\{\mathbf{b}\} = \mathbf{E}\left\{\left(\mathbf{X'X}\right)^{-1}\mathbf{X'Y}\right\} = \left(\mathbf{X'X}\right)^{-1}\mathbf{X'E}\{\mathbf{Y}\} = \left(\mathbf{X'X}\right)^{-1}\mathbf{X'X\beta} = \boldsymbol{\beta}$$

$$\boldsymbol{\sigma}^2\{\mathbf{b}\} = \begin{bmatrix} \sigma^2\{b_0\} & \sigma\{b_0,b_1\} & \cdots & \sigma\{b_0,b_{p-1}\} \\ \sigma\{b_1,b_0\} & \sigma^2\{b_1\} & \cdots & \sigma\{b_1,b_{p-1}\} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma\{b_{p-1},b_0\} & \sigma\{b_{p-1},b_1\} & \cdots & \sigma^2\{b_{p-1}\} \end{bmatrix} \qquad \mathbf{s}^2\{\mathbf{b}\} = \begin{bmatrix} s^2\{b_0\} & s\{b_0,b_1\} & \cdots & s\{b_0,b_{p-1}\} \\ s\{b_1,b_0\} & s^2\{b_1\} & \cdots & s\{b_1,b_{p-1}\} \\ \vdots & \vdots & \ddots & \vdots \\ s\{b_{p-1},b_0\} & s\{b_{p-1},b_1\} & \cdots & s^2\{b_{p-1}\} \end{bmatrix}$$

$$\boldsymbol{\sigma}^2\{\mathbf{b}\} = \boldsymbol{\sigma}^2\left\{\left(\mathbf{X'X}\right)^{-1}\mathbf{X'Y}\right\} = \left(\mathbf{X'X}\right)^{-1}\mathbf{X'}\boldsymbol{\sigma}^2\{\mathbf{Y}\}\left(\left(\mathbf{X'X}\right)^{-1}\mathbf{X'}\right)' = \sigma^2\left(\mathbf{X'X}\right)^{-1}\mathbf{X'X}\left(\mathbf{X'X}\right)^{-1} = \sigma^2\left(\mathbf{X'X}\right)^{-1}$$

$$\mathbf{s}^2\{\mathbf{b}\} = MSE\left(\mathbf{X'X}\right)^{-1}$$

$$\frac{b_k - \beta_k}{s\{b_k\}} \sim t(n-p) \qquad (1-\alpha)100\%\text{ CI for } \beta_k \equiv b_k \pm t\left(1-\frac{\alpha}{2}; n-p\right)s\{b_k\}$$

Test of $H_0: \beta_k = 0 \quad H_A: \beta_k \neq 0$ Test Statistic: $t^* = \dfrac{b_k}{s\{b_k\}}$

Rejection Region: $|t^*| \geq t\left(1-\dfrac{\alpha}{2}; n-p\right)$ P-value=$2\Pr\left(t(n-p) \geq |t^*|\right)$

Simultaneous $(1-\alpha)100\%$ CI^s for $g \leq p \quad \beta^s: \quad b_k \pm t\left(1-\dfrac{\alpha}{2g}; n-p\right)s\{b_k\}$

Example: Factors Effecting Air Permeability of Woven Fabrics

$$s^2\{b\} = MSE\left(\mathbf{X'X}\right)^{-1} = 31.29\begin{bmatrix} 8.547494 & -0.21469 & -0.16004 & 0.072775 \\ -0.21469 & 0.010359 & 0.011853 & -0.00603 \\ -0.16004 & 0.011853 & 0.016936 & -0.00838 \\ 0.072775 & -0.00603 & -0.00838 & 0.004269 \end{bmatrix} \qquad t(.975, 26) = 2.056$$

Test for $\beta_1: \quad TS: t_1^* = \dfrac{-0.39211}{\sqrt{31.29(0.010359)}} = \dfrac{-0.39211}{0.56933} = -0.689$

95% CI for $\beta_1: -0.39211 \pm 2.056(0.56933) \equiv -0.39211 \pm 1.17054 \equiv (-1.563, 0.778)$

Test for $\beta_2: \quad TS: t_2^* = \dfrac{-0.67957}{\sqrt{31.29(0.016936)}} = \dfrac{-0.67957}{0.72796} = -0.934$

95% CI for $\beta_2: -0.67957 \pm 2.056(0.72796) \equiv -0.67957 \pm 1.49669 \equiv (-2.176, 0.817)$

Test for $\beta_3: \quad TS: t_3^* = \dfrac{-0.35497}{\sqrt{31.29(0.004269)}} = \dfrac{-0.35497}{0.36548} = -0.971$

95% CI for $\beta_3: -0.35497 \pm 2.056(0.36548) \equiv -0.35497 \pm 0.75143 \equiv (-1.106, 0.396)$

Note: Individually, no terms are significant, but as a group they are. We will understand why later.

Estimating Mean Response at Specific X-levels

Given set of levels of $X_1,...,X_{p-1}$: $X_{h1},...,X_{h,p-1}$

$$\underset{p\times 1}{\mathbf{X}_h}=\begin{bmatrix}1\\X_{h1}\\\vdots\\X_{h,p-1}\end{bmatrix} \qquad E\{Y_h\}=\mathbf{X}_h'\boldsymbol{\beta} \qquad \hat{Y}_h=\mathbf{X}_h'\mathbf{b}$$

$$E\left\{\hat{Y}_h\right\}=\mathbf{X}_h'\boldsymbol{\beta} \qquad \sigma^2\left\{\hat{Y}_h\right\}=\mathbf{X}_h'\boldsymbol{\sigma}^2\{\mathbf{b}\}\mathbf{X}_h=\sigma^2\mathbf{X}_h'\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}_h \qquad s^2\left\{\hat{Y}_h\right\}=MSE\left(\mathbf{X}_h'\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}_h\right)$$

$(1-\alpha)100\%$ CI for $E\left\{\hat{Y}_h\right\}$: $\hat{Y}_h\pm t\left(1-\frac{\alpha}{2};n-p\right)s\left\{\hat{Y}_h\right\}$

$(1-\alpha)100\%$ Confidence Region for Regression Surface: $\hat{Y}_h\pm Ws\left\{\hat{Y}_h\right\}$ $\qquad W^2=pF\left(1-\alpha;p,n-p\right)$

$(1-\alpha)100\%$ CI for several (g) $E\left\{\hat{Y}_h\right\}$: $\hat{Y}_h\pm Bs\left\{\hat{Y}_h\right\}$ $\qquad B=t\left(1-\frac{\alpha}{2g};n-p\right)$

Example: Factors Effecting Air Permeability of Woven Fabrics

Estimating the mean at Warp=X_1 = 60,Weft= X_2 =35, Mass=X_3 = 125

$$\mathbf{X_h}'=\begin{bmatrix}1 & 60 & 35 & 125\end{bmatrix} \qquad \hat{Y}_h=\mathbf{X_h}'\mathbf{b}=\begin{bmatrix}1 & 60 & 35 & 125\end{bmatrix}\begin{bmatrix}105.0349\\-0.39211\\-0.67957\\-0.35497\end{bmatrix}=13.3514$$

$$\mathbf{X_h}'\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X_h}=0.5862 \qquad s\left\{\hat{Y}_h\right\}=\sqrt{31.29(.5862)}=4.2828$$

95% CI for $\mathbf{X_h}'\boldsymbol{\beta}$: $13.3514\pm 2.055(4.2828)\equiv 13.3514\pm 8.8011\equiv(4.5503,22.1525)$

Predicting New Response(s) at Specific X-levels

Given set of levels of $X_1,...,X_{p-1}$: $X_{h1},...,X_{h,p-1}$

$$\underset{p\times 1}{\mathbf{X}_h}=\begin{bmatrix}1\\X_{h1}\\\vdots\\X_{h,p-1}\end{bmatrix} \qquad E\{Y_h\}=\mathbf{X}_h'\boldsymbol{\beta} \qquad \hat{Y}_h=\mathbf{X}_h'\mathbf{b}$$

$$s^2\{\text{pred}\}=MSE\left(1+\mathbf{X}_h'\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}_h\right)$$

mean of m observations (at same X-levels): $s^2\{\text{predmean}\}=MSE\left(\frac{1}{m}+\mathbf{X}_h'\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}_h\right)$

$(1-\alpha)100\%$ CI for $Y_{h(\text{new})}$: $\hat{Y}_h\pm t\left(1-\frac{\alpha}{2};n-p\right)s\{\text{pred}\}$

Scheffe: $(1-\alpha)100\%$ CI for several (g) $Y_{h(\text{new})}$: $\hat{Y}_h\pm Ss\{\text{pred}\}$ $\qquad S^2=gF(1-\alpha;g,n-p)$

Bonferroni: $(1-\alpha)100\%$ CI for several (g) $Y_{h(\text{new})}$: $\hat{Y}_h\pm Bs\{\text{pred}\}$ $\qquad B=t\left(1-\frac{\alpha}{2g};n-p\right)$

Example: Factors Effecting Air Permeability of Woven Fabrics

Predicting a new Air Permeability observation at Warp=X_1 = 60,Weft= X_2 =35, Mass=X_3 = 125

$$\mathbf{X_h'}=\begin{bmatrix}1 & 60 & 35 & 125\end{bmatrix} \qquad \hat{Y}_h=\mathbf{X_h'b}=\begin{bmatrix}1 & 60 & 35 & 125\end{bmatrix}\begin{bmatrix}105.0349\\-0.39211\\-0.67957\\-0.35497\end{bmatrix}=13.3514$$

$$\mathbf{X_h'}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X_h}=0.5862 \qquad s\{\text{pred}\}=\sqrt{31.29(1+0.5862)}=7.0450$$

95% PI for $Y_{h(new)}$: $13.3514\pm 2.055(7.0450)\equiv 13.3514\pm 14.4775\equiv(-1.1261,27.8289)$

Presumably, air permeability measurements cannot be negative, so we would interpret the prediction interval as being (0 , 27.8289)

R Program for Chapter 6 Examples – Air Permeability

```
airperm <- read.csv("E:\\blue_drive\\sta4210\\airperm_woven_reg.csv",
       header=TRUE)

attach(airperm)
names(airperm)

Y <- mean_ap
X1 <- warp
X2 <- weft
X3 <- mass
n <- length(Y)
X0 <- rep(1,n)
X <- as.matrix(cbind(X0,X1,X2,X3))    # Form the X-matrix (n=30 rows, 4 Cols)
Y <- as.matrix(Y,ncol=1)       # Form the Y-vector (n=30 rows, 1 col)
p <- ncol(X)

# Notes:   t(X) = transpose of X, %*% = matrix multiplication, solve(A) = A^(-1)

(XX <- t(X) %*% X)             # Obtain X'X matrix (4 rows, 4 cols)

(XY <- t(X) %*% Y)             # Obtain X'Y vector (4 rows, 1 col)

(XXI <- solve(XX))             # Obtain (X'X)^(-1) matrix (4 rows, 4 cols)

(b <- XXI %*% XY)              # Obtain b-vector (4 rows, 1 col)

Y_hat <- X %*% b               # Obtain the vector of fitted values (n=30 rows, 1 col)

e <- Y - Y_hat                 # Obtain the vector of residuals (n=30 rows, 1 col)

print(cbind(Y_hat,e))

H <- X %*% XXI %*% t(X)              # Obtain the Hat matrix

J_n <- matrix(rep(1/n,n^2),ncol=n)   # Obtain the (1/n)J matrix (n=30 rows, n=30 cols)

I_n <- diag(n)                       # Obtain the identity matrix (n=30 rows, n=30 cols)

(SSTO <- t(Y) %*% (I_n - J_n) %*% Y) # Obtain Total Sum of Squares (SSTO)
# SSTO can also be computed as:
# (SSTO <- (t(Y) %*% Y) - (t(Y) %*% (I_n - J_n) %*% Y))

(SSE <- t(Y) %*% (I_n - H) %*% Y)    # Obtain Error Sum of Squares (SSE)
# SSE can also be computed as:
# (SSE <- (t(Y) %*% Y) - (t(b) %*% XY))

(SSR <- t(Y) %*% (H - J_n) %*% Y)    # Obtain Regression Sum of Squares (SSR)
# SSR can also be computed as:
# (SSR <- (t(b) %*% XY) - (t(Y) %*% J_n %*% Y))

(MSE <- SSE/(n-p))                   # Obtain MSE = s^2

(s2_b <- MSE[1,1] * XXI)             # Obtain s^2{b}, must use MSE[1,1] and * to do scalar
multiplication

se_b <- sqrt(diag(s2_b))            # Obtain SE's of individual regression coefficients
```

Continued Below

```
se_b <- sqrt(diag(s2_b))            # Obtain SE's of individual regression coefficients

print(cbind((b-qt(.975,n-p)*se_b),(b+qt(.975,n-p)*se_b)))  # Print CI's for Beta coefficients

(X_h <- matrix(c(1,60,35,125),ncol=1))      # Create X_h vector, for case where
X1=60,X2=35,X3=125

(Y_hat_h <- t(X_h) %*% b)            # Obtain the fitted value when budget=20

(s2_yhat_h <- t(X_h) %*% s2_b %*% X_h)        # Obtain s^2{Y_hat_h}

(s2_pred <- MSE + (t(X_h) %*% s2_b %*% X_h))  # Obtain s^2{pred}

### Print 95% CI for Mean at X_h and 95% PI for Individual Observation

print(cbind((Y_hat_h-qt(.975,n-p)*sqrt(s2_yhat_h)),(Y_hat_h+qt(.975,n-p)*sqrt(s2_yhat_h))))

print(cbind((Y_hat_h-qt(.975,n-p)*sqrt(s2_pred)),(Y_hat_h+qt(.975,n-p)*sqrt(s2_pred))))
```

R Output

```
> (XX <- t(X) %*% X)             # Obtain X'X matrix (4 rows, 4 cols)
       X0       X1       X2       X3
X0   30.0   1800.0    969.0   3931.3
X1 1800.0 108540.0  58113.0 236587.2
X2  969.0  58113.0  33353.0 130975.2
X3 3931.3 236587.2 130975.2 524246.0
>
> (XY <- t(X) %*% Y)             # Obtain X'Y vector (4 rows, 1 col)
       [,1]
X0   391.23
X1 23028.66
X2  9833.43
X3 45054.13
>
> (XXI <- solve(XX))             # Obtain (X'X)^(-1) matrix (4 rows, 4 cols)
            X0           X1           X2           X3
X0  8.54749374 -0.214691264 -0.160042763  0.072775138
X1 -0.21469126  0.010358774  0.011852801 -0.006026103
X2 -0.16004276  0.011852801  0.016936315 -0.008380192
X3  0.07277514 -0.006026103 -0.008380192  0.004269361
>
> (b <- XXI %*% XY)              # Obtain b-vector (4 rows, 1 col)
          [,1]
X0 105.0349322
X1  -0.3921132
X2  -0.6795729
X3  -0.3549737

> (SSTO <- t(Y) %*% (I_n - J_n) %*% Y) # Obtain Total Sum of Squares (SSTO)
         [,1]
[1,] 5098.981
>
> (SSE <- t(Y) %*% (I_n - H) %*% Y)    # Obtain Error Sum of Squares (SSE)
         [,1]
[1,] 813.5993
>
> (SSR <- t(Y) %*% (H - J_n) %*% Y)    # Obtain Regression Sum of Squares (SSR)
         [,1]
[1,] 4285.382
```

Continued Below

```
> (MSE <- SSE/(n-p))                    # Obtain MSE = s^2
         [,1]
[1,] 31.29228
>
> (s2_b <- MSE[1,1] * XXI)              # Obtain s^2{b}, must use MSE[1,1] and * to do
scalar multiplication
           X0         X1         X2         X3
X0 267.470590 -6.7181797 -5.0081034  2.2773002
X1  -6.718180  0.3241497  0.3709012 -0.1885705
X2  -5.008103  0.3709012  0.5299760 -0.2622353
X3   2.277300 -0.1885705 -0.2622353  0.1335980
>
> se_b <- sqrt(diag(s2_b))             # Obtain SE's of individual regression coefficients
>
> print(cbind((b-qt(.975,n-p)*se_b),(b+qt(.975,n-p)*se_b)))  # Print CI's for Beta
coefficients
         [,1]        [,2]
X0 71.417718 138.6521462
X1 -1.562411   0.7781849
X2 -2.175987   0.8168412
X3 -1.106292   0.3963443
>
> (X_h <- matrix(c(1,60,35,125),ncol=1))      # Create X_h vector, for case where
X1=60,X2=35,X3=125
     [,1]
[1,]    1
[2,]   60
[3,]   35
[4,]  125
>
> (Y_hat_h <- t(X_h) %*% b)             # Obtain the fitted value when budget=20
         [,1]
[1,] 13.35138
>
> (s2_yhat_h <- t(X_h) %*% s2_b %*% X_h)        # Obtain s^2{Y_hat_h}
         [,1]
[1,] 18.34364
>
> (s2_pred <- MSE + (t(X_h) %*% s2_b %*% X_h))  # Obtain s^2{pred}
         [,1]
[1,] 49.63592
>
> ### Print 95% CI for Mean at X_h and 95% PI for Individual Observation
> print(cbind((Y_hat_h-qt(.975,n-p)*sqrt(s2_yhat_h)),(Y_hat_h+qt(.975,n-
p)*sqrt(s2_yhat_h))))
         [,1]    [,2]
[1,] 4.547652 22.1551
> print(cbind((Y_hat_h-qt(.975,n-p)*sqrt(s2_pred)),(Y_hat_h+qt(.975,n-p)*sqrt(s2_pred))))
          [,1]     [,2]
[1,] -1.130396 27.83315
>
```

Any differences between these output values and those within the chapter are due to the fact that I used 5 decimal places on calculations.

Chapter 7 – Multiple Regression II

Extra Sums of Squares

- For a given dataset, the total sum of squares remains the same, no matter what predictors are included (when no missing values exist among variables)
- As we include more predictors, the regression sum of squares (*SSR*) increases (technically does not decrease), and the error sum of squares (*SSE*) decreases
- $SSR + SSE = SSTO$, regardless of predictors in model
- When a model contains just X_1, denote: $SSR(X_1)$, $SSE(X_1)$
- Model Containing X_1, X_2: $SSR(X_1,X_2)$, $SSE(X_1,X_2)$
- Predictive contribution of X_2 above that of X_1:

$$SSR(X_2|X_1) = SSE(X_1) - SSE(X_1,X_2) = SSR(X_1,X_2) - SSR(X_1)$$

- Extends to any number of Predictors

Definitions and Decomposition of SSR

$$SSTO = SSR(X_1) + SSE(X_1) = SSR(X_1, X_2) + SSE(X_1, X_2) = SSR(X_1, X_2, X_3) + SSE(X_1, X_2, X_3)$$

$$SSR(X_1 \mid X_2) = SSR(X_1, X_2) - SSR(X_2) = SSE(X_2) - SSE(X_1, X_2)$$
$$SSR(X_2 \mid X_1) = SSR(X_1, X_2) - SSR(X_1) = SSE(X_1) - SSE(X_1, X_2)$$

$$SSR(X_3 \mid X_1, X_2) = SSR(X_1, X_2, X_3) - SSR(X_1, X_2) = SSE(X_1, X_2) - SSE(X_1, X_2, X_3)$$
$$SSR(X_2, X_3 \mid X_1) = SSR(X_1, X_2, X_3) - SSR(X_1) = SSE(X_1) - SSE(X_1, X_2, X_3)$$

$$SSR(X_1, X_2) = SSR(X_1) + SSR(X_2 \mid X_1) = SSR(X_2) + SSR(X_1 \mid X_2)$$

$$SSR(X_1, X_2, X_3) = SSR(X_1) + SSR(X_2 \mid X_1) + SSR(X_3 \mid X_1, X_2)$$
$$SSR(X_1, X_2, X_3) = SSR(X_2) + SSR(X_1 \mid X_2) + SSR(X_3 \mid X_1, X_2)$$
$$SSR(X_1, X_2, X_3) = SSR(X_1) + SSR(X_2, X_3 \mid X_1)$$

Example: Factors Effecting Air Permeability of Woven Fabrics

The following partial ANOVA tables are for all 7 possible models containing at least one of the 3 predictors.

Predictors	SSR	SSE	dfR	dfE
X1	366.94	4732.04	1	28
X2	3825.38	1273.60	1	28
X3	4254.74	844.24	1	28
X1,X2	4255.87	843.11	2	27
X1,X3	4258.11	840.87	2	27
X2,X3	4270.54	828.44	2	27
X1,X2,X3	4285.38	813.60	3	26

$$SSTO = SSR(X_1) + SSE(X_1) = SSR(X_1, X_2) + SSE(X_1, X_2) = SSR(X_1, X_2, X_3) + SSE(X_1, X_2, X_3)$$

$$SSR(X_1) + SSE(X_1) = 366.94 + 4732.04 = SSR(X_1, X_2) + SSE(X_1, X_2) = 4255.87 + 843.11 = 5098.98$$

$$SSR(X_1 \mid X_2) = SSR(X_1, X_2) - SSR(X_2) = 4255.87 - 3825.38 = 430.49$$

$$SSR(X_2 \mid X_1) = SSR(X_1, X_2) - SSR(X_1) = 4255.87 - 366.94 = 3888.93$$

$$SSR(X_3 \mid X_1, X_2) = SSR(X_1, X_2, X_3) - SSR(X_1, X_2) = 4285.38 - 4255.87 = 29.51$$

$$SSR(X_2, X_3 \mid X_1) = SSR(X_1, X_2, X_3) - SSR(X_1) = 4285.38 - 366.94 = 3918.44$$

$$SSR(X_1, X_2) = SSR(X_1) + SSR(X_2 \mid X_1) = SSR(X_2) + SSR(X_1 \mid X_2)$$

$$SSR(X_1) + SSR(X_2 \mid X_1) = 366.94 + 3888.93 = SSR(X_2) + SSR(X_1 \mid X_2) = 3825.38 + 430.49$$

$$SSR(X_1, X_2, X_3) = SSR(X_1) + SSR(X_2 \mid X_1) + SSR(X_3 \mid X_1, X_2) = 366.94 + 3888.93 + 29.51$$

$$SSR(X_1, X_2, X_3) = SSR(X_2) + SSR(X_1 \mid X_2) + SSR(X_3 \mid X_1, X_2) = 3825.38 + 430.49 + 29.51$$

$$SSR(X_1, X_2, X_3) = SSR(X_1) + SSR(X_2, X_3 \mid X_1) = 366.94 + 3918.41$$

Note that as the # of predictors increases, so does the ways of decomposing SSR

ANOVA – Sequential Sum of Squares

This is a partitioning of the Regression sum of squares in the full model, into its sequential sums of squares for the variables in the order of their appearance in the regression program. For the case of 3 predictors, entered in the order: X_1, X_2, X_3:

Source of Variation	SS	df	MS
Regression	SSR(X1,X2,X3)	3	MSR(X1,X2,X3)
X1	SSR(X1)	1	MSR(X1)
X2|X1	SSR(X2|X1)	1	MSR(X2|X1)
X3|X1,X2	SSR(X3|X1,X2)	1	MSR(X3|X1,X2)
Error	SSE(X1,X2,X3)	n-4	MSE(X1,X2,X3)
Total	SSTO	n-1	

$$MSR(X_1) = \frac{SSR(X_1)}{1} \qquad MSR(X_2 \mid X_1) = \frac{SSR(X_2 \mid X_1)}{1}$$

$$MSR(X_3 \mid X_1, X_2) = \frac{SSR(X_3 \mid X_1, X_2)}{1}$$

$$MSR(X_1, X_2, X_3) = \frac{SSR(X_1, X_2, X_3)}{3}$$

$$MSR(X_2, X_3 \mid X_1) = \frac{SSR(X_2, X_3 \mid X_1)}{2}$$

Example: Factors Effecting Air Permeability of Woven Fabrics

Source of Variation	SS	df	MS
Regression	4285.38	3	1428.46
X1	366.94	1	366.94
X2|X1	3888.93	1	3888.93
X3|X1,X2	29.51	1	29.51
Error	813.6	26	31.29
Total	5098.98	29	

Note: 366.94 + 3888.93 + 29.51 = 4285.38

Extra Sums of Squares & Tests of Regression Coefficients (Single β_k)

Full Model: $Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \varepsilon_i \qquad \varepsilon_i \sim NID\left(0, \sigma^2\right)$

$H_0 : \beta_3 = 0 \quad H_A : \beta_3 \neq 0 \quad \Rightarrow \quad$ Reduced Model: $Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i$

General Linear Test: $F^* = \dfrac{\left[\dfrac{SSE(R) - SSE(F)}{df_R - df_f}\right]}{\left[\dfrac{SSE(F)}{df_f}\right]}$

Full Model: $SSE(F) = SSE\left(X_1, X_2, X_3\right) \qquad df_F = n - 4$

Reduced Model: $SSE(R) = SSE\left(X_1, X_2\right) \qquad df_R = n - 3$

$SSE(R) - SSE(F) = SSE\left(X_1, X_2\right) - SSE\left(X_1, X_2, X_3\right) = SSR\left(X_3 \mid X_1, X_2\right)$

$df_R - df_F = \left(n-3\right) - \left(n-4\right) = 1$

$$\Rightarrow \quad F^* = \frac{\left[\dfrac{SSR\left(X_3 \mid X_1, X_2\right)}{1}\right]}{\left[\dfrac{SSE\left(X_1, X_2, X_3\right)}{n-4}\right]} = \frac{MSR\left(X_3 \mid X_1, X_2\right)}{MSE\left(X_1, X_2, X_3\right)} \overset{H_0}{\sim} F\left(1, n-4\right)$$

Rejection Region: $F^* \geq F\left(1-\alpha; 1, n-4\right) \qquad P - \text{value}: P\left(F\left(1, n-4\right) \geq F^*\right)$

Example: Factors Effecting Air Permeability of Woven Fabrics

$H_0 : \beta_3 = 0 \quad H_A : \beta_3 \neq 0$

$n = 30 \quad MSR\left(X_3 \mid X_1, X_2\right) = \dfrac{29.51}{1} = 29.51 \quad MSE\left(X_1, X_2, X_3\right) = \dfrac{813.6}{26} = 31.29$

$F^* = \dfrac{29.51}{31.29} = 0.943 \qquad F\left(.95; 1, 26\right) = 4.225 \qquad P - \text{value} = P\left(F\left(1, 26\right) \geq 0.943\right) = 0.3405$

This F-test gives the exact same result as the t-test from Chapter 6.

$$\left(t^*\right)^2 = F^* \qquad \left(t\left(.975, n-4\right)\right)^2 = F\left(.95; 1, n-4\right)$$

Extra Sums of Squares & Tests of Regression Coefficients (Multiple β_k)

Full Model: $Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \varepsilon_i \qquad \varepsilon_i \sim NID\left(0, \sigma^2\right)$

$H_0 : \beta_2 = \beta_3 = 0 \quad H_A : \beta_2$ and/or $\beta_3 \neq 0 \quad \Rightarrow \quad$ Reduced Model: $Y_i = \beta_0 + \beta_1 X_{i1} + \varepsilon_i$

General Linear Test: $F^* = \dfrac{\left[\dfrac{SSE(R) - SSE(F)}{df_R - df_f}\right]}{\left[\dfrac{SSE(F)}{df_f}\right]}$

Full Model: $SSE(F) = SSE\left(X_1, X_2, X_3\right) \qquad df_F = n - 4$

Reduced Model: $SSE(R) = SSE\left(X_1\right) \qquad df_R = n - 2$

$SSE(R) - SSE(F) = SSE\left(X_1\right) - SSE\left(X_1, X_2, X_3\right) = SSR\left(X_2, X_3 \mid X_1\right)$

$df_R - df_F = \left(n-2\right) - \left(n-4\right) = 2$

$$\Rightarrow \quad F^* = \frac{\left[\dfrac{SSR\left(X_2, X_3 \mid X_1\right)}{2}\right]}{\left[\dfrac{SSE\left(X_1, X_2, X_3\right)}{n-4}\right]} = \frac{MSR\left(X_2, X_3 \mid X_1\right)}{MSE\left(X_1, X_2, X_3\right)} \overset{H_0}{\sim} F\left(2, n-4\right)$$

Rejection Region: $F^* \geq F\left(1-\alpha; 2, n-4\right) \qquad P - \text{value}: P\left(F\left(2, n-4\right) \geq F^*\right)$

Example: Factors Effecting Air Permeability of Woven Fabrics

$H_0 : \beta_2 = \beta_3 = 0 \quad H_A : \beta_2$ and/or $\beta_3 \neq 0$

$n = 30 \quad MSR\left(X_2, X_3 \mid X_1\right) = \dfrac{3888.93 + 29.51}{2} = 1959.22 \quad MSE\left(X_1, X_2, X_3\right) = \dfrac{813.6}{26} = 31.29$

$F^* = \dfrac{1959.22}{31.29} = 62.615 \qquad F\left(.95; 2, 26\right) = 3.369 \qquad P - \text{value} = P\left(F\left(2, 26\right) \geq 62.615\right) = .0000$

Note that the individual *t*-tests for Wear (X_2) and Mass (X_3) were not significant, but when we test them simultaneously, they are highly significant. This is due to the fact that they are highly correlated, and both related to Air Permeability (Y). We will look at this in more detail in the section on **Multicollinearity**.

Extra Sums of Squares & Tests of Regression Coefficients (General Case)

Full Model: $Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_{p-1} X_{i,p-1} + \varepsilon_i \qquad \varepsilon_i \sim NID\left(0, \sigma^2\right)$

$H_0 : \beta_q = ... = \beta_{p-1} = 0 \quad H_A :$ At least one of β_q ... $\beta_{p-1} \neq 0$

$\Rightarrow$ Reduced Model: $Y_i = \beta_0 + \beta_1 X_{i1} ... + \beta_{q-1} X_{i,q-1} + \varepsilon_i \qquad (q < p)$

General Linear Test: $F^* = \dfrac{\left[\dfrac{SSE(R) - SSE(F)}{df_R - df_F}\right]}{\left[\dfrac{SSE(F)}{df_F}\right]}$

Full Model: $SSE(F) = SSE\left(X_1, X_2 ..., X_{p-1}\right) \qquad df_F = n - p$

Reduced Model: $SSE(R) = SSE\left(X_1, X_2 ..., X_{q-1}\right) \qquad df_R = n - q$

$SSE(R) - SSE(F) = SSE\left(X_1, X_2 ..., X_{q-1}\right) - SSE\left(X_1, X_2 ..., X_{p-1}\right) = SSR\left(X_q, ..., X_{p-1} \mid X_1, X_2 ..., X_{q-1}\right)$

$df_R - df_F = (n-q) - (n-p) = p - q$

$$\Rightarrow \quad F^* = \frac{\left[\dfrac{SSR\left(X_q, ..., X_{p-1} \mid X_1, X_2 ..., X_{q-1}\right)}{p-q}\right]}{\left[\dfrac{SSE\left(X_1, X_2 ..., X_{p-1}\right)}{n-p}\right]} = \frac{MSR\left(X_q, ..., X_{p-1} \mid X_1, X_2 ..., X_{q-1}\right)}{MSE\left(X_1, X_2 ..., X_{p-1}\right)} \overset{H_0}{\sim} F\left(p-q, n-p\right)$$

Rejection Region: $F^* \geq F\left(1-\alpha; p-q, n-p\right) \qquad P\text{-value} = P\left(F\left(p-q; n-p\right) \geq F^*\right)$

Since there are only three predictors in the Air Permeability model, we will not do this as an example.

Other Linear Tests

Suppose firm has two types of advertising:
print $\left(X_1,\right.$ in \$1000s$\left.\right)$ and internet $\left(X_2,\right.$ in \$1000s$\left.\right)$ as well as promotional expenditures $\left(X_3,\right.$ in \$1000s$\left.\right)$:
Let Sales = Y, they vary their expenditures on n periods and observe sales in each (Price is constant)

$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \varepsilon_i \qquad \varepsilon_i \sim NID\left(0, \sigma^2\right)$

Test of equal effects of increasing each input by 1 unit (say \$1000s):

$H_0 : \beta_1 = \beta_2 = \beta_3 \qquad H_A : H_0$ is False

Full Model: $Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \varepsilon_i \qquad df_F = n - 4$

Reduced Model: $Y_i = \beta_0 + \beta_1 X_{i1} + \beta_1 X_{i2} + \beta_1 X_{i3} + \varepsilon_i \quad \Rightarrow \quad Y_i = \beta_0 + \beta_1 \left(X_{i1} + X_{i2} + X_{i3}\right) + \varepsilon_i$

$\Rightarrow \quad Y_i = \beta_0 + \beta_1 W_i + \varepsilon_i \qquad W_i = X_{i1} + X_{i2} + X_{i3} \qquad df_R = n - 2$

Suppose firm has two types of advertising:
print $(X_1,$ in \$1000s$)$ and internet $(X_2,$ in \$1000s$)$ as well as promotional expenditures $(X_3,$ in \$1000s$)$:
Let Sales = Y, they vary their expenditures on n periods and observe sales in each (Price is constant)

$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \varepsilon_i \qquad \varepsilon_i \sim NID(0, \sigma^2)$

Test that Mean sales when all inputs=0 is \$10,000 and effect of increasing X_3 by 1 unit is 1:
$H_0 : \beta_0 = 10, \beta_3 = 1 \qquad H_A : H_0$ is False (all units are \$1000s)
Full Model: $Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \varepsilon_i \qquad df_F = n - 4$
Reduced Model: $Y_i = 10 + \beta_1 X_{i1} + \beta_2 X_{i2} + 1X_{i3} + \varepsilon_i \quad \Rightarrow \quad Y_i - 10 - 1X_{i3} = \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i$
$\Rightarrow \quad U_i = \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i \qquad U_i = Y_i - 10 - 1X_{i3} \qquad df_R = n - 2$ (no intercept)

Coefficients of Partial Determination-I

Proportion of Variation Explained by 1 or more variables, not explained by others

Regression of Y on X_1: $Y_i = \beta_0 + \beta_1 X_{i1} + \varepsilon_i$
Variation Explained: $SSR(X_1)$ Unexplained: $SSE(X_1) = SSTO - SSR(X_1)$

Regression of Y on X_2: $Y_i = \beta_0 + \beta_2 X_{i2} + \varepsilon_i$
Variation Explained: $SSR(X_2)$ Unexplained: $SSE(X_2) = SSTO - SSR(X_2)$

Regression of Y on X_1, X_2: $Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i$
Variation Explained: $SSR(X_1, X_2)$ Unexplained: $SSE(X_1, X_2) = SSTO - SSR(X_1, X_2)$

Proportion of Variation in Y, Not Explained by X_1, that is Explained by X_2:

$$R^2_{Y2|1} = \frac{SSE(X_1) - SSE(X_1, X_2)}{SSE(X_1)} = \frac{SSR(X_1, X_2) - SSR(X_1)}{SSE(X_1)} = \frac{SSR(X_1, X_2) - SSR(X_1)}{SSTO - SSR(X_1)} = \frac{SSR(X_2 \mid X_1)}{SSE(X_1)}$$

Proportion of Variation in Y, Not Explained by X_2, that is Explained by X_1:

$$R^2_{Y2|1} = \frac{SSE(X_2) - SSE(X_1, X_2)}{SSE(X_2)} = \frac{SSR(X_1, X_2) - SSR(X_2)}{SSE(X_2)} = \frac{SSR(X_1, X_2) - SSR(X_2)}{SSTO - SSR(X_2)} = \frac{SSR(X_1 \mid X_2)}{SSE(X_2)}$$

Coefficients of Partial Determination-II

$$R^2_{Y1|23} = \frac{SSE(X_2, X_3) - SSE(X_1, X_2, X_3)}{SSE(X_2, X_3)} = \frac{SSR(X_1, X_2, X_3) - SSR(X_2, X_3)}{SSE(X_2, X_3)} =$$

$$= \frac{SSR(X_1, X_2, X_3) - SSR(X_2, X_3)}{SSTO - SSR(X_2, X_3)} = \frac{SSR(X_1 \mid X_2, X_3)}{SSE(X_2, X_3)}$$

$$R^2_{Y2|13} = \frac{SSR(X_1, X_2, X_3) - SSR(X_1, X_3)}{SSTO - SSR(X_1, X_3)} = \frac{SSR(X_2 \mid X_1, X_3)}{SSE(X_1, X_3)}$$

$$R^2_{Y3|12} = \frac{SSR(X_1, X_2, X_3) - SSR(X_1, X_2)}{SSTO - SSR(X_1, X_2)} = \frac{SSR(X_3 \mid X_1, X_2)}{SSE(X_1, X_2)}$$

$$R^2_{Y23|1} = \frac{SSE(X_1) - SSE(X_1, X_2, X_3)}{SSE(X_1)} = \frac{SSR(X_1, X_2, X_3) - SSR(X_1)}{SSE(X_1)} =$$

$$= \frac{SSR(X_1, X_2, X_3) - SSR(X_1)}{SSTO - SSR(X_1)} = \frac{SSR(X_2, X_3 \mid X_1)}{SSE(X_1)}$$

Coefficient of Partial Correlation:

$$R_{Y2|1} = \text{sgn}\{\beta_2\}\sqrt{R^2_{Y2|1}} \qquad \text{sgn}\{\beta_2\} = \begin{cases} + \text{ if } \hat{\beta}_2 > 0 \text{ in } \hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 \\ - \text{ if } \hat{\beta}_2 < 0 \text{ in } \hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 \end{cases}$$

Example: Factors Effecting Air Permeability of Woven Fabrics

$SSR(X_1) = 366.94 \quad SSE(X_1) = 4732.04 \quad SSR(X_1, X_2) = 4255.87 \quad SSE(X_1, X_2) = 843.11$

$SSR(X_1, X_2, X_3) = 4285.38$

$SSR(X_2 \mid X_1) = 4255.87 - 366.94 = 3888.93 \quad SSE(X_1) = 4732.04 \quad \Rightarrow \quad R^2_{Y2\bullet1} = \frac{3888.93}{4732.04} = 0.8218$

$SSR(X_3 \mid X_1, X_2) = 4285.38 - 4255.87 = 29.51 \quad SSE(X_1, X_2) = 843.11 \quad \Rightarrow \quad R^2_{Y3\bullet12} = \frac{29.51}{843.11} = 0.0350$

$SSR(X_2, X_3 \mid X_1) = 4285.38 - 366.94 = 3918.44 \quad SSE(X_1) = 4732.04 \quad \Rightarrow \quad R^2_{Y2\bullet1} = \frac{3918.44}{4732.04} = 0.8281$

Standardized Regression Model

- Useful in removing round-off errors in computing $(\mathbf{X'X})^{-1}$
- Makes easier comparison of magnitude of effects of predictors measured on different measurement scales
- Coefficients represent changes in Y (in standard deviation units) as each predictor increases 1 SD (holding all others constant)
- Since all variables are centered, no intercept term

Standardized Random Variables: Scaled to have mean=0, SD=1

$$\frac{Y_i-\overline{Y}}{s_Y} \qquad s_Y=\sqrt{\frac{\sum_i\left(Y_i-\overline{Y}\right)^2}{n-1}} \qquad \frac{X_{ik}-\overline{X}_k}{s_k} \qquad s_k=\sqrt{\frac{\sum_i\left(X_{ik}-\overline{X}_k\right)^2}{n-1}} \qquad k=1,\ldots,p-1$$

Correlation Transformation:

$$Y_i^*=\frac{1}{\sqrt{n-1}}\left(\frac{Y_i-\overline{Y}}{s_Y}\right) \qquad X_{ik}^*=\frac{1}{\sqrt{n-1}}\left(\frac{X_{ik}-\overline{X}_k}{s_k}\right) \qquad k=1,\ldots,p-1$$

Standardized Regression Model:

$$Y_i^*=\beta_1^*X_{i1}^*+\ldots+\beta_{p-1}^*X_{i,p-1}^*+\varepsilon_i^*$$

Note: $\beta_k=\left(\frac{s_Y}{s_k}\right)\beta_k^* \qquad k=1,\ldots,p-1 \qquad \beta_0=\overline{Y}-\beta_1\overline{X}_1-\ldots-\beta_{p-1}\overline{X}_{p-1}$

Standardized Regression Model:

$$Y_i^*=\beta_1^*X_{i1}^*+\ldots+\beta_{p-1}^*X_{i,p-1}^*+\varepsilon_i^*$$

$$\underset{n\times(p-1)}{\mathbf{X}^*}=\begin{bmatrix}X_{11}^* & \cdots & X_{1,p-1}^*\\ \vdots & & \vdots\\ X_{n1}^* & \cdots & X_{n,p-1}^*\end{bmatrix} \qquad \underset{n\times1}{\mathbf{Y}^*}=\begin{bmatrix}Y_1^*\\ Y_1^*\\ \vdots\\ Y_1^*\end{bmatrix} \qquad \underset{(p-1)\times(p-1)}{\mathbf{X}^{*'}\mathbf{X}^*}=\begin{bmatrix}1 & r_{12} & \cdots & r_{1,p-1}\\ r_{21} & 1 & \cdots & r_{1,p-1}\\ \cdots & \cdots & \ddots & \cdots\\ r_{p-1,1} & r_{p-1,2} & \cdots & 1\end{bmatrix}=\mathbf{r}_{XX} \qquad \underset{(p-1)\times1}{\mathbf{X}^{*'}\mathbf{Y}^*}=\begin{bmatrix}r_{Y1}\\ r_{Y2}\\ \vdots\\ r_{Y,p-1}\end{bmatrix}=\mathbf{r}_{YX}$$

This results from:

$$\sum_i\left(X_{ik}^*\right)^2=\sum_i\left(\frac{1}{\sqrt{n-1}}\left(\frac{X_{ik}-\overline{X}_k}{s_k}\right)\right)^2=\left(\frac{1}{s_k^2}\right)\frac{\sum_i\left(X_{ik}-\overline{X}_k\right)^2}{n-1}=\left(\frac{1}{s^2\{X_k\}}\right)s^2\{X_k\}=1$$

$$\sum_i\left(X_{ik}^*\right)\left(X_{ik'}^*\right)=\sum_i\left(\frac{1}{\sqrt{n-1}}\left(\frac{X_{ik}-\overline{X}_k}{s_k}\right)\right)\left(\frac{1}{\sqrt{n-1}}\left(\frac{X_{ik'}-\overline{X}_{k'}}{s_{k'}}\right)\right)=\left(\frac{1}{s_ks_{k'}}\right)\frac{\sum_i\left(X_{ik}-\overline{X}_k\right)\left(X_{ik'}-\overline{X}_{k'}\right)}{n-1}=\frac{s\{X_k,X_{k'}\}}{s\{X_k\}s\{X_{k'}\}}=r_{kk'}$$

$$\sum_i\left(Y_i^*\right)\left(X_{ik}^*\right)=\sum_i\left(\frac{1}{\sqrt{n-1}}\left(\frac{Y_i-\overline{Y}}{s_Y}\right)\right)\left(\frac{1}{\sqrt{n-1}}\left(\frac{X_{ik}-\overline{X}_k}{s_k}\right)\right)=\left(\frac{1}{s_Ys_k}\right)\frac{\sum_i\left(Y_i-\overline{Y}\right)\left(X_{ik}-\overline{X}_k\right)}{n-1}=\frac{s\{Y,X_k\}}{s\{Y\}s\{X_k\}}=r_{Yk}$$

Standardized Regression Model:

$$Y_i^* = \beta_1^* X_{i1}^* + \ldots + \beta_{p-1}^* X_{i,p-1}^* + \varepsilon_i^*$$

$$\underset{n\times(p-1)}{\mathbf{X}^*} = \begin{bmatrix} X_{11}^* & \cdots & X_{1,p-1}^* \\ \vdots & & \vdots \\ X_{n1}^* & \cdots & X_{n,p-1}^* \end{bmatrix} \qquad \underset{n\times 1}{\mathbf{Y}^*} = \begin{bmatrix} Y_1^* \\ Y_1^* \\ \vdots \\ Y_1^* \end{bmatrix} \qquad \underset{(p-1)\times(p-1)}{\mathbf{X}^{*'}\mathbf{X}^*} = \begin{bmatrix} 1 & r_{12} & \cdots & r_{1,p-1} \\ r_{21} & 1 & \cdots & r_{1,p-1} \\ \cdots & \cdots & \ddots & \cdots \\ r_{p-1,1} & r_{p-1,2} & \cdots & 1 \end{bmatrix} = \mathbf{r}_{XX} \qquad \underset{(p-1)\times 1}{\mathbf{X}^{*'}\mathbf{Y}^*} = \begin{bmatrix} r_{Y1} \\ r_{Y2} \\ \vdots \\ r_{Y,p-1} \end{bmatrix} = \mathbf{r}_{YX}$$

$$\underset{(p-1)\times(p-1)}{\mathbf{X}^{*'}\mathbf{X}^*}\ \underset{(p-1)\times 1}{\mathbf{b}^*} = \underset{(p-1)\times 1}{\mathbf{X}^{*'}\mathbf{Y}^*} \quad \Rightarrow \quad \mathbf{b}^* = \left(\mathbf{X}^{*'}\mathbf{X}^*\right)^{-1}\mathbf{X}^{*'}\mathbf{Y}^* \quad \Rightarrow \quad \mathbf{b}^* = \mathbf{r}_{XX}^{-1}\mathbf{r}_{YX}$$

$$b_k = \left(\frac{s_Y}{s_k}\right) b_k^* \qquad k = 1, \ldots, p-1 \qquad b_0 = \overline{Y} - b_1\overline{X}_1 - \ldots - b_{p-1}\overline{X}_{p-1}$$

Example: Factors Effecting Air Permeability of Woven Fabrics

warp	weft	mass	airperm	warp*	weft*	mass*	airperm*
54	20	101.0	45.74	-0.2691	-0.27299	-0.33361	0.454728
54	25	110.3	27.02	-0.2691	-0.16202	-0.23034	0.194399
54	30	119.5	15.68	-0.2691	-0.05105	-0.12818	0.036699
54	35	127.3	8.76	-0.2691	0.059924	-0.04157	-0.05953
54	40	136.7	4.37	-0.2691	0.170895	0.062814	-0.12058
54	45	144.1	2.90	-0.2691	0.281866	0.144986	-0.14103
57	20	101.2	47.94	-0.13455	-0.27299	-0.33139	0.485323
57	25	111.1	27.52	-0.13455	-0.16202	-0.22146	0.201352
57	30	120.5	14.84	-0.13455	-0.05105	-0.11708	0.025018
57	35	130.7	8.99	-0.13455	0.059924	-0.00381	-0.05634
57	40	142.0	3.98	-0.13455	0.170895	0.121667	-0.12601
57	45	146.3	2.68	-0.13455	0.281866	0.169415	-0.14409
60	20	107.9	33.98	0	-0.27299	-0.25699	0.291188
60	25	118.7	17.01	0	-0.16202	-0.13706	0.055195
60	30	129.5	9.75	0	-0.05105	-0.01714	-0.04577
60	35	141.4	4.56	0	0.059924	0.115004	-0.11794
60	40	151.2	2.12	0	0.170895	0.223827	-0.15187
60	45	153.7	1.70	0	0.281866	0.251587	-0.15771
63	20	109.0	27.68	0.134549	-0.27299	-0.24478	0.203577
63	25	119.1	15.24	0.134549	-0.16202	-0.13262	0.03058
63	30	129.1	7.23	0.134549	-0.05105	-0.02158	-0.08081
63	35	148.5	3.22	0.134549	0.059924	0.193845	-0.13658
63	40	150.3	1.89	0.134549	0.170895	0.213833	-0.15507
63	42	154.4	1.61	0.134549	0.215284	0.259361	-0.15897
66	20	111.2	27.28	0.269098	-0.27299	-0.22035	0.198015
66	25	123.0	14.42	0.269098	-0.16202	-0.08932	0.019177
66	30	134.0	6.90	0.269098	-0.05105	0.032832	-0.0854
66	35	144.4	3.19	0.269098	0.059924	0.148317	-0.13699
66	40	155.0	1.65	0.269098	0.170895	0.266023	-0.15841
66	42	160.2	1.38	0.269098	0.215284	0.323766	-0.16216

X*'X*				X*'Y*		s_y	13.25998	b0	105.03
1	-0.02564	0.320361		-0.26826		s_1	4.31517	b1	-0.39211
-0.02564	1	0.925054		-0.86616		s_2	8.416527	b2	-0.67957
0.320361	0.925054	1		-0.91347		s_3	17.6902	b3	-0.35497
						ybar	13.04		
INV(X*'X*)				b*		xbar1	60		
5.593738	12.48389	-13.3403		-0.1276		xbar2	32.3		
12.48389	34.79227	-36.1841		-0.43135		xbar3	131.0		
-13.3403	-36.1841	38.74596		-0.47357					

Note that the correlation between Weft and Mass is 0.925054. That leads us to Multicollinearity.

Multicollinearity

- Consider model with 2 Predictors (this generalizes to any number of predictors)
  $Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i$
- When X_1 and X_2 are uncorrelated, the regression coefficients b_1 and b_2 are the same whether we fit simple regressions or a multiple regression, and: $SSR(X_1) = SSR(X_1|X_2)$ $SSR(X_2) = SSR(X_2|X_1)$
- When X_1 and X_2 are highly correlated, their regression coefficients become unstable, and their standard errors become larger (smaller t-statistics, wider CI's), leading to strange inferences when comparing simple and partial effects of each predictor
- Estimated means and Predicted values are not affected

Example: Factors Effecting Air Permeability of Woven Fabrics

Warp, Weft				
	Coefficients	*tandard Err*	*t Stat*	*P-value*
Intercept	111.0858	15.1053	7.3541	0.0000
warp	-0.8932	0.2406	-3.7129	0.0009
weft	-1.3763	0.1233	-11.1597	0.0000

Warp, Mass				
	Coefficients	*tandard Err*	*t Stat*	*P-value*
Intercept	98.6132	14.8019	6.6622	0.0000
warp	0.0835	0.2535	0.3293	0.7445
mass	-0.6912	0.0618	-11.1778	0.0000

Weft, Mass				
	Coefficients	*tandard Err*	*t Stat*	*P-value*
Intercept	96.9082	11.2132	8.6423	0.0000
weft	-0.2309	0.3218	-0.7177	0.4791
mass	-0.5831	0.1531	-3.8090	0.0007

Warp, Weft, Mass				
	Coefficients	*tandard Err*	*t Stat*	*P-value*
Intercept	105.0349	16.3545	6.4224	0.0000
warp	-0.3921	0.5693	-0.6887	0.4971
weft	-0.6796	0.7280	-0.9335	0.3592
mass	-0.3550	0.3655	-0.9712	0.3404

Compare the standard errors of the Weft coefficient: changing from 0.1233 to 0.7280 depending on whether Mass is in the model (a 6-fold increase when Mass is included). Similarly, standard error for the Mass coefficient changes from 0.0618 without Weft to 0.3655 with Weft.

R Program for Chapter 7 Examples – Air Permeability

```
airperm <- read.csv("E:\\blue_drive\\sta4210\\airperm_woven_reg.csv",
       header=TRUE)

attach(airperm)
names(airperm)

plot(airperm[,2:5])   ### Scatterplot Matrix of X1,X2,X3,Y

ssto <- sum((mean_ap-mean(mean_ap))^2)  ### Total Sum of Squares

########## Fit all 7 possible Models
ap.mod123 <- lm(mean_ap ~ warp + weft + mass)
summary(ap.mod123)
anova(ap.mod123)

ap.mod12 <- lm(mean_ap ~ warp + weft)
#summary(ap.mod12)
#anova(ap.mod12)

ap.mod13 <- lm(mean_ap ~ warp + mass)
#summary(ap.mod13)
#anova(ap.mod13)

ap.mod23 <- lm(mean_ap ~ weft + mass)
#summary(ap.mod23)
#anova(ap.mod23)

ap.mod1 <- lm(mean_ap ~ warp)
#summary(ap.mod1)
#anova(ap.mod1)

ap.mod2 <- lm(mean_ap ~ weft)
#summary(ap.mod2)
#anova(ap.mod2)

ap.mod3 <- lm(mean_ap ~ mass)
#summary(ap.mod3)
#anova(ap.mod3)
####### Obtain SSE and SSR for each model (deviance=SSE)
sse.x1 <- deviance(ap.mod1); ssr.x1 <- ssto-sse.x1
sse.x2 <- deviance(ap.mod2); ssr.x2 <- ssto-sse.x2
sse.x3 <- deviance(ap.mod3); ssr.x3 <- ssto-sse.x3
sse.x1x2 <- deviance(ap.mod12); ssr.x1x2 <- ssto-sse.x1x2
sse.x1x3 <- deviance(ap.mod13); ssr.x1x3 <- ssto-sse.x1x3
sse.x2x3 <- deviance(ap.mod23); ssr.x2x3 <- ssto-sse.x2x3
sse.x1x2x3 <- deviance(ap.mod123); ssr.x1x2x3 <- ssto-sse.x1x2x3
#### Compute Sequential Sums of Squares
ssr.x1x2-ssr.x1             ### SSR(X2|X1)
ssr.x1x2-ssr.x2             ### SSR(X2|X1)
ssr.x1x2x3-ssr.x1x2         ### SSR(X3|X1,X2)

#### Test H0: B2=B3=0
anova(ap.mod1,ap.mod123)

### Compue Coefficients of Partial Determination
(ssr.x1x2-ssr.x1)/sse.x1         ### R2(Y2|1)
(ssr.x1x2x3-ssr.x1x2)/sse.x1x2    ### R2(Y3|12)
```

Continued Below

```
#### Correlation Transformation and Standardized Regression Coefficients
y.corr <- (mean_ap-mean(mean_ap))/(sqrt(29)*sd(mean_ap))
x1.corr <- (warp-mean(warp))/(sqrt(29)*sd(warp))
x2.corr <- (weft-mean(weft))/(sqrt(29)*sd(weft))
x3.corr <- (mass-mean(mass))/(sqrt(29)*sd(mass))

xstar <- matrix(cbind(x1.corr,x2.corr,x3.corr),ncol=3)
ystar <- matrix(y.corr)


bstar <- solve(t(xstar) %*% xstar) %*% t(xstar) %*% ystar
bstar

##### Regression coefficient estimates for all 2 and 3 variable models
summary(ap.mod12)
summary(ap.mod13)
summary(ap.mod23)
summary(ap.mod123)
```

R Output

```
> ap.mod123 <- lm(mean_ap ~ warp + weft + mass)
> summary(ap.mod123)

Call:
lm(formula = mean_ap ~ warp + weft + mass)

Residuals:
    Min      1Q  Median      3Q     Max
-6.8875 -4.2474 -0.6555  2.4051 14.7703

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 105.0349    16.3545   6.422 8.36e-07 ***
warp         -0.3921     0.5693  -0.689    0.497
weft         -0.6796     0.7280  -0.933    0.359
mass         -0.3550     0.3655  -0.971    0.340

Residual standard error: 5.594 on 26 degrees of freedom
Multiple R-squared:  0.8404,    Adjusted R-squared:  0.822
F-statistic: 45.65 on 3 and 26 DF,  p-value: 1.678e-10



> anova(ap.mod123)

Analysis of Variance Table

Response: mean_ap
          Df Sum Sq Mean Sq  F value    Pr(>F)
warp       1  366.9   366.9  11.7263  0.002054 **
weft       1 3888.9  3888.9 124.2774 2.106e-11 ***
mass       1   29.5    29.5   0.9432  0.340412
Residuals 26  813.6    31.3
---
```

Continued Below

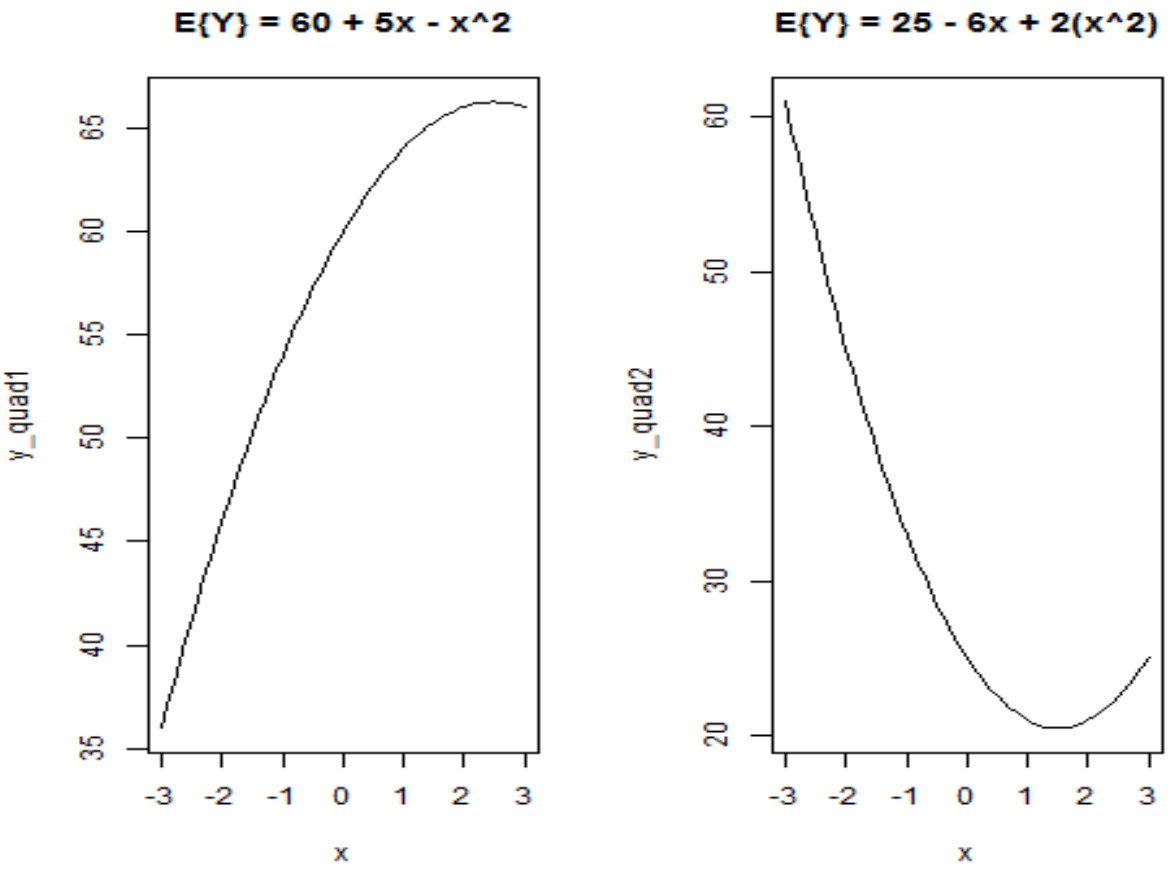
```
> ssr.x1x2-ssr.x1             ### SSR(X2|X1)
[1] 3888.924
> ssr.x1x2-ssr.x2             ### SSR(X1|X2)
[1] 430.4843
> ssr.x1x2x3-ssr.x1x2         ### SSR(X3|X1,X2)
[1] 29.51409
>
> anova(ap.mod1,ap.mod123)
Analysis of Variance Table

Model 1: mean_ap ~ warp
Model 2: mean_ap ~ warp + weft + mass
  Res.Df    RSS Df Sum of Sq     F    Pr(>F)
1     28 4732.0
2     26  813.6  2    3918.4 62.61 1.147e-10 ***

> (ssr.x1x2-ssr.x1)/sse.x1         ### R2(Y2|1)
[1] 0.8218287
>
> (ssr.x1x2x3-ssr.x1x2)/sse.x1x2    ### R2(Y3|12)
[1] 0.03500607

> bstar <- solve(t(xstar) %*% xstar) %*% t(xstar) %*% ystar
> bstar
           [,1]
[1,] -0.1276047
[2,] -0.4313465
[3,] -0.4735722

> summary(ap.mod12)
lm(formula = mean_ap ~ warp + weft)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 111.0858    15.1053   7.354 6.54e-08 ***
warp         -0.8932     0.2406  -3.713 0.000941 ***
weft         -1.3763     0.1233 -11.160 1.28e-11 ***
---

> summary(ap.mod13)
lm(formula = mean_ap ~ warp + mass)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 98.61319   14.80193   6.662 3.77e-07 ***
warp         0.08348    0.25351   0.329    0.744
mass        -0.69123    0.06184 -11.178 1.24e-11 ***
---
> summary(ap.mod23)
lm(formula = mean_ap ~ weft + mass)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  96.9082    11.2132   8.642 2.95e-09 ***
weft         -0.2309     0.3218  -0.718 0.479137
mass         -0.5831     0.1531  -3.809 0.000732 ***
---
> summary(ap.mod123)
lm(formula = mean_ap ~ warp + weft + mass)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 105.0349    16.3545   6.422 8.36e-07 ***
warp         -0.3921     0.5693  -0.689    0.497
weft         -0.6796     0.7280  -0.933    0.359
mass         -0.3550     0.3655  -0.971    0.340
```

Chapter 8 – Models for Quantitative and Qualitative Predictors

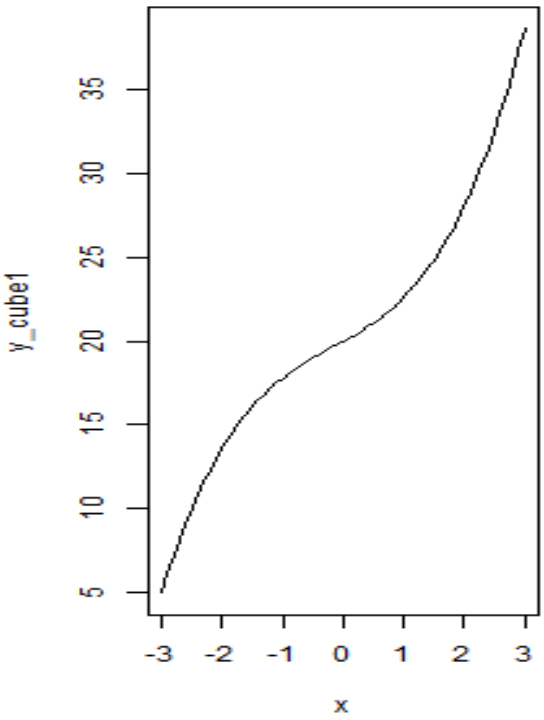
Polynomial Regression Models

- Useful in 2 Settings:
  - True relation between response and predictor is polynomial
  - True relation is complex nonlinear function that can be approximated by polynomial in specific range of *X*-levels
- Models with 1 Predictor: Including *p* polynomial terms in model, creates *p*-1 "bends"
  - 2nd order Model: $E\{Y\} = \beta_0 + \beta_1 x + \beta_2 x^2$ (x = centered X)
  - 3rd order Model: $E\{Y\} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$
- Response Surfaces with 2 (or more) predictors
  - 2nd order model with 2 Predictors:

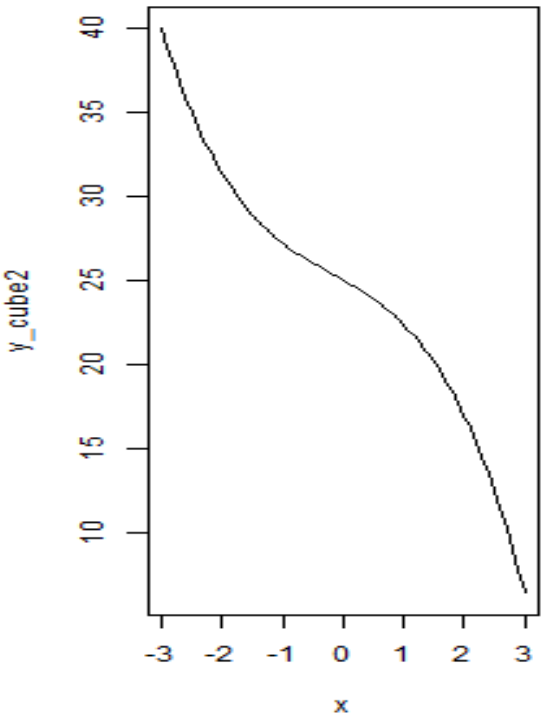
$$E\{Y\} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 \qquad x_1 = X_1 - \overline{X}_1 \quad x_2 = X_2 - \overline{X}_2$$



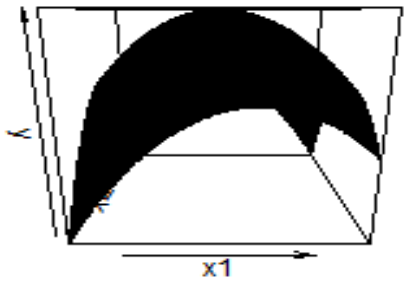
E{Y} = 20+2*x+0.2*x^2+0.4*x^3



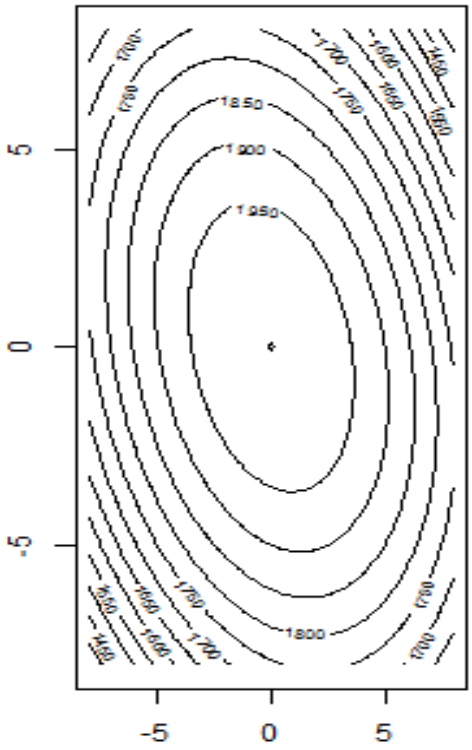
E{Y} = 25-2*x-0.2*x^2-0.4*x^3



E{Y} = 2000-4x1^2-4x2^2-2x1x2



E{Y} = 2000-4x1^2-4x2^2-2x1x2



Modeling Strategies

- Use Extra Sums of Squares and General Linear Tests to compare models of increasing complexity (higher order)
- Use coding in fitting models (centered/scaled) predictors to reduce multicollinearity when conducting testing. Keep lower order terms whenever higher order polynomials and interactions are included in a model, even if not significant.
- Fit models in original units, or back-transform for plotting on original scale* (see below for quadratic)
- For Response Surfaces, include multiple replicates at center point for goodness-of-fit tests

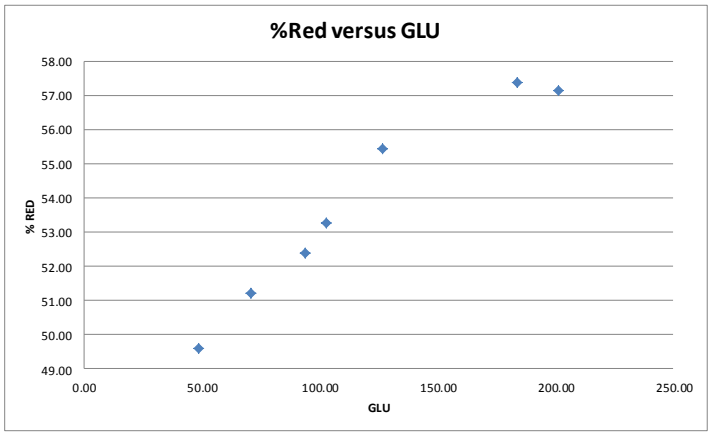
Centered variables: $\hat{Y} = b_0 + b_1 x + b_{11} x^2 = b_0 + b_1\left(X - \overline{X}\right) + b_{11}\left(X - \overline{X}\right)^2$

$= b_0 + b_1 X - b_1 \overline{X} + b_{11} X^2 - 2b_{11} X \overline{X} + b_{11} \overline{X}^2 = \left(b_0 - b_1 \overline{X} + b_{11} \overline{X}^2\right) + \left(b_1 - 2b_{11} \overline{X}\right) X + b_{11} \overline{X}^2 = b_o' + b_1' X + b_2' X^2$

Example: Relationship Between Wine Color (% Red) and Anthocyanin Glucosides (GLU)

A study related various types of color parameters and phenolic compounds in three varieties of wine (Tempranillo, Graciano, and Cabernet Sauvignon) over a 26 month storage period. The following plot is of the percent red color (Y) versus Anthocyanin Glucosides (X).

Source: M. Monagas, P. J. Martın-Alvarez,, B. Bartolome · C. Gomez-Cordoves (2006). "Statistical interpretation of the color parameters of red wines in function of their phenolic composition during aging in bottle," *European Food Research Technology*, Vol. 222, pp. 702-709.



GLU	GLU2	GLUC	GLUC2	%Red
201.25	40501.56	82.97	6884.50	57.15
183.97	33844.96	65.69	4315.55	57.36
126.69	16050.36	8.41	70.78	55.43
102.78	10563.73	-15.50	240.16	53.25
93.85	8807.82	-24.43	596.69	52.39
70.76	5006.98	-47.52	2257.88	51.21
48.64	2365.85	-69.64	4849.33	49.59

GLUC and GLUC2 are the centered, and squared centered values of GLU

Fitting the Regression Model:

$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i \qquad \varepsilon_i \sim NID\left(0, \sigma^2\right) \qquad x_i = X_i - \overline{X}$$

	Coefficients	Standard Err	t Stat	P-value	Lower 95%	Upper 95%
Intercept	54.42328	0.277486	196.1302	4.05E-09	53.65286	55.19371
GLUC	0.056116	0.003692	15.19824	0.000109	0.045864	0.066367
GLUC2	-0.00024	7.87E-05	-3.0312	0.038735	-0.00046	-2E-05

Note that all coefficients are significant. Also, the intercept represents the fitted value at x=0, or equivalently, the fitted value at the mean GLU level. The fitted equation, and back-transformed coefficients are:

$$\hat{Y} = 54.4233 + 0.0561x - 0.00024x^2 \qquad \overline{X} = 118.2771$$

$$b_0' = 54.4233 - 0.0561(118.2771) + (-0.00024)(118.2771)^2 = 54.4233 - 6.6353 - 3.3575 = 44.4305$$

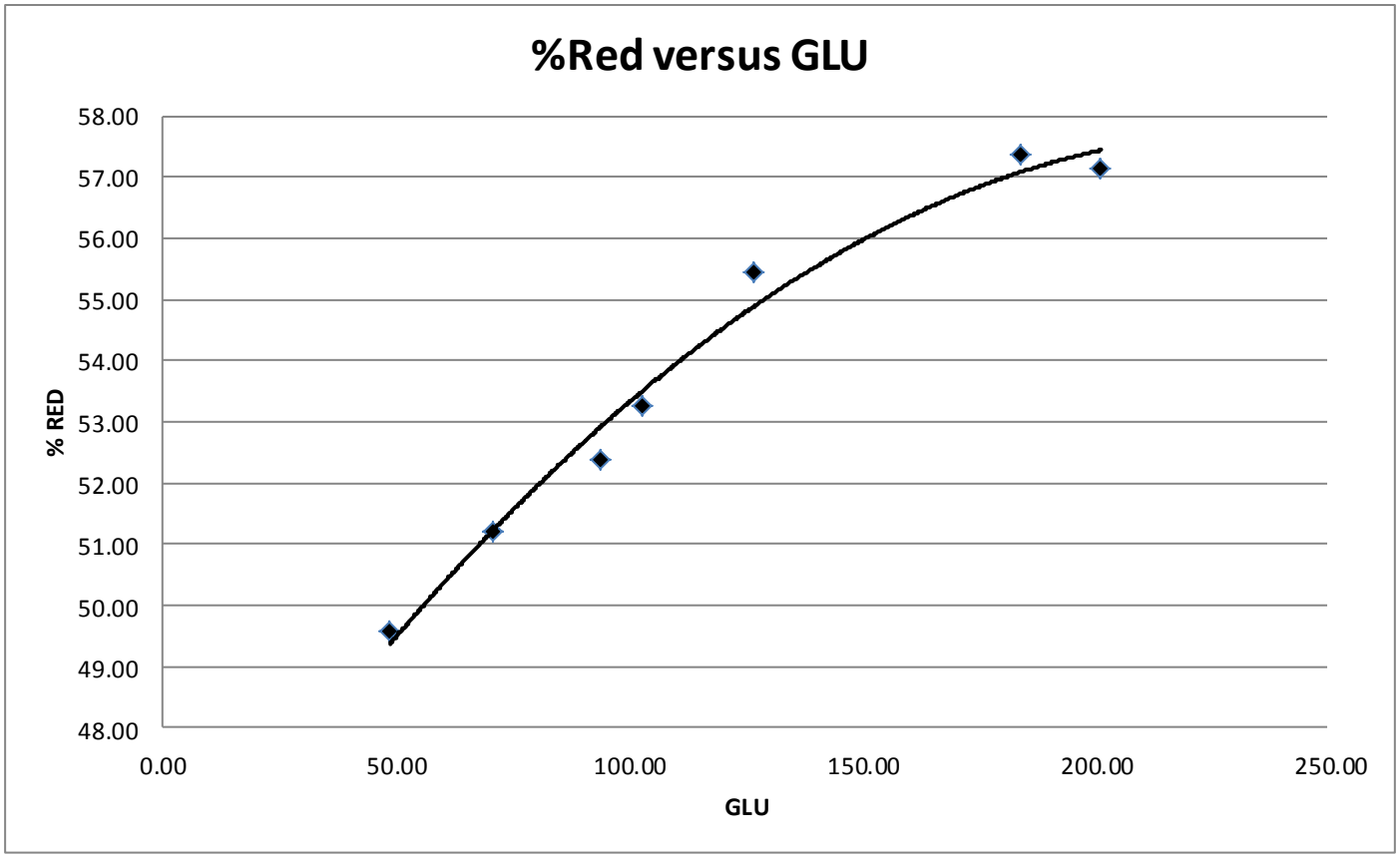
$$b_1' = 0.0561 - 2(-0.00024)(118.2771) = 0.0561 + 0.0568 = 0.1129$$

$$b_2' = -0.00024$$

$$\Rightarrow \quad \hat{Y} = 44.4302 + 0.1129X - 0.00024X^2$$

Below is the model fit with the un-centered (original) GLU values (differences are due to rounding):

	Coefficients	Standard Err	t Stat	P-value	Lower 95%	Upper 95%
Intercept	44.44944	1.157544	38.39979	2.75E-06	41.23558	47.6633
GLU	0.112537	0.02049	5.49216	0.005356	0.055646	0.169427
GLU2	-0.00024	7.87E-05	-3.0312	0.038735	-0.00046	-2E-05



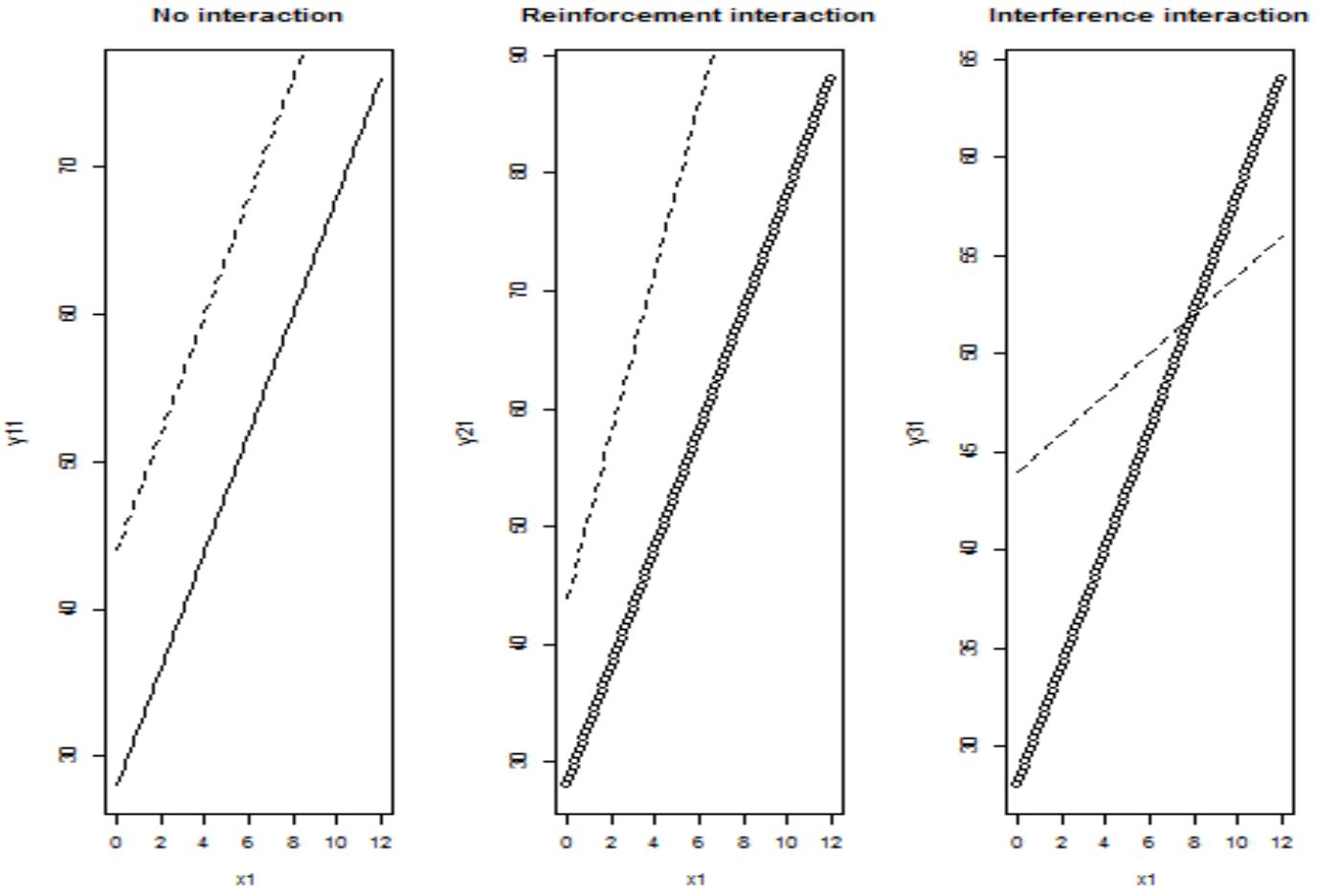
Regression Models with Interaction Term(s)

- Interaction $\Rightarrow$ Effect (Slope) of one predictor variable depends on the level other predictor variable(s)
- Formulated by including cross-product term(s) among predictor variables
- 2 Variable Models: $E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2$

$$X_2 = 0 \Rightarrow E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 (0) + \beta_3 (X_1(0)) = \beta_0 + \beta_1 X_1$$

$$X_2 = 10 \Rightarrow E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 (10) + \beta_3 (X_1(10)) = \beta_0 + \beta_1 X_1 + 10\beta_2 + 10\beta_3 X_1 =$$
$$= (\beta_0 + 10\beta_2) + (\beta_1 + 10\beta_3) X_1$$

Testing Hypothesis of no interaction: $H_0 : \beta_3 = 0 \quad H_A : \beta_3 \neq 0$



Example – Response Surface Relating 3 Factors to Color Intensity Indigo Dye Applied to Cotton

Source: M.B. Ticha, N. Meksi, N. Driri, M. Kechida, and M.F. Mhenni (2013). "A promising route to dye cotton by indigo with an ecological exhaustion process: A dyeing process optimization based on a response surface methodology," *Industrial Crops and Products*, Vol. 46, pp. 350-358.

An experiment was conducted relating X_1 = Temperature (Celsius, Levels=35,60,100), X_2 = Time (Minutes, Levels=30,60,90,120), and X_3 = Cationizing Agent (Percentage, Levels=0,4,10,15,20) to Y = Color Yield (K/S ratio). The experiment was made up of n = 60 runs. To obtain the same results as the authors, we will use the original data levels, not centered or scaled levels. The second order response surface is of the form:

$$E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_{11} X_1^2 + \beta_{22} X_2^2 + \beta_{33} X_3^2 + \beta_{12} X_1 X_2 + \beta_{13} X_1 X_3 + \beta_{23} X_2 X_3$$

EXCEL Output:

Regression Statistics						
Multiple R	0.9919					
R Square	0.9838					
Adjusted R Square	0.9808					
Standard Error	0.6612					
Observations	60					
ANOVA						
	df	*SS*	*MS*	*F*	*gnificance F*	
Regression	9	1324.97	147.22	336.73	0.0000	
Residual	50	21.86	0.44			
Total	59	1346.83				
	Coefficients	*andard Err*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	-20.503	1.087422	-18.8547	0.0000	-22.6872	-18.3188
Temp	0.704351	0.026628	26.4512	0.0000	0.650867	0.757836
Time	0.064974	0.016093	4.0374	0.0002	0.03265	0.097299
Cation	0.409912	0.058253	7.0367	0.0000	0.292907	0.526918
Temp2	-0.00396	0.000183	-21.6810	0.0000	-0.00433	-0.00359
Time2	-0.00024	9.48E-05	-2.4894	0.0162	-0.00043	-4.6E-05
Cation2	-0.01758	0.002096	-8.3877	0.0000	-0.02179	-0.01337
TempTime	-0.00023	9.51E-05	-2.3990	0.0202	-0.00042	-3.7E-05
TempCat	0.000892	0.000442	2.0198	0.0488	4.96E-06	0.001779

Note that all regression coefficients are significant, implying a very complex surface in four dimensions. To observe the surface, we will use the **rsm** package in R.

R Program

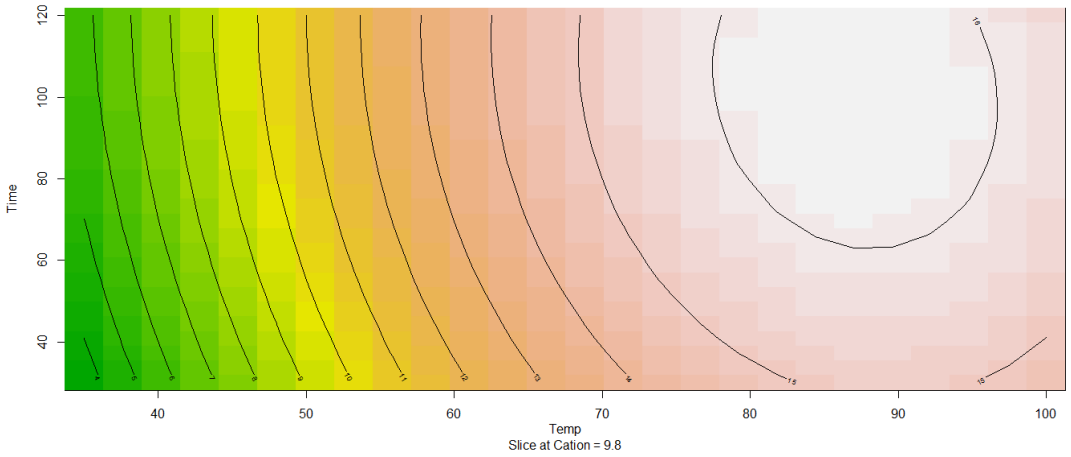
```
dye <- read.csv("E:\\blue_drive\\sta4210\\dye_cotton_rsm.csv", header=TRUE)
attach(dye); names(dye)

install.packages("rsm")
library(rsm)

dye.rsm1 <- rsm(KS ~ SO(Temp,Time,Cation))
summary(dye.rsm1)
drop1(dye.rsm1)

contour(dye.rsm1, ~ Temp + Time, image=TRUE)
contour(dye.rsm1, ~ Temp + Cation, image=TRUE)
contour(dye.rsm1, ~ Time + Cation, image=TRUE)
```

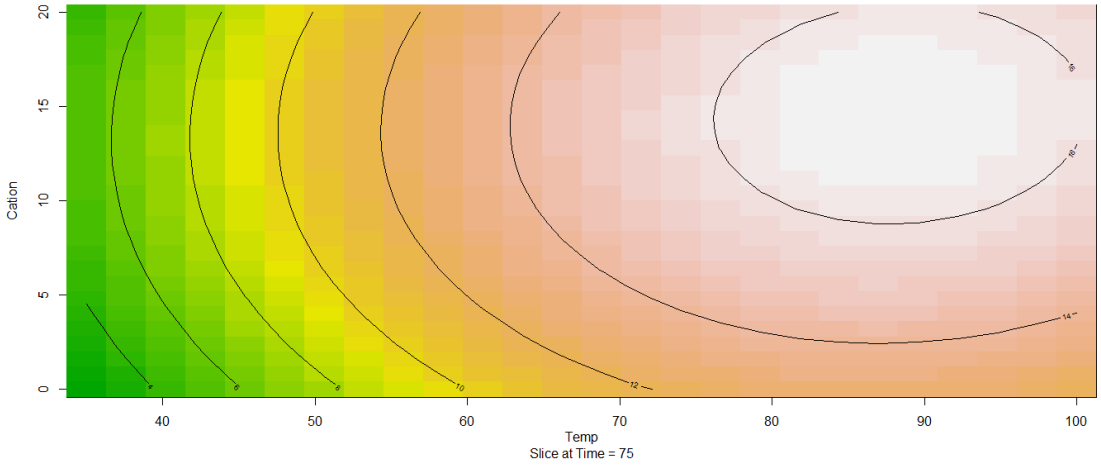
R Text Output

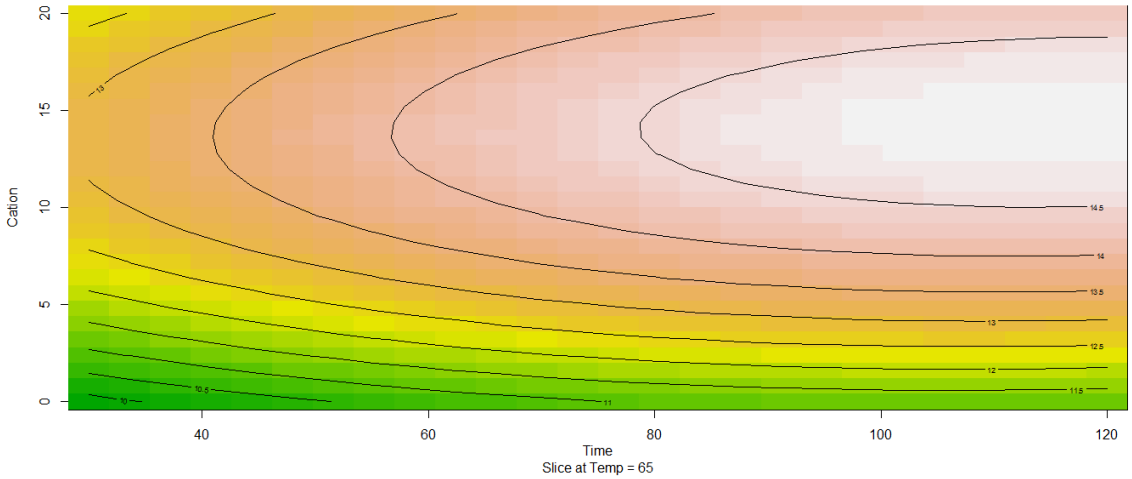
```
> dye.rsm1 <- rsm(KS ~ SO(Temp,Time,Cation))
> summary(dye.rsm1)

Call:
rsm(formula = KS ~ SO(Temp, Time, Cation))

               Estimate  Std. Error  t value  Pr(>|t|)
(Intercept) -2.0503e+01  1.0874e+00 -18.8547 < 2.2e-16 ***
Temp         7.0435e-01  2.6628e-02  26.4512 < 2.2e-16 ***
Time         6.4974e-02  1.6093e-02   4.0374 0.0001856 ***
Cation       4.0991e-01  5.8253e-02   7.0367 5.257e-09 ***
Temp:Time   -2.2806e-04  9.5067e-05  -2.3990 0.0202118 *
Temp:Cation  8.9176e-04  4.4151e-04   2.0198 0.0487793 *
Time:Cation  3.3488e-04  3.5239e-04   0.9503 0.3465153
Temp^2      -3.9607e-03  1.8268e-04 -21.6810 < 2.2e-16 ***
Time^2      -2.3611e-04  9.4847e-05  -2.4894 0.0161676 *
Cation^2    -1.7583e-02  2.0963e-03  -8.3877 4.207e-11 **

---

Multiple R-squared:  0.9838,    Adjusted R-squared:  0.9808
F-statistic: 336.7 on 9 and 50 DF,  p-value: < 2.2e-16

Analysis of Variance Table

Response: KS
                        Df  Sum Sq Mean Sq  F value  Pr(>F)
FO(Temp, Time, Cation)   3 1081.30  360.43 824.4098 < 2e-16
TWI(Temp, Time, Cation)  3    4.69    1.56   3.5792 0.02014
PQ(Temp, Time, Cation)   3  238.98   79.66 182.2053 < 2e-16
Residuals               50   21.86    0.44
Lack of fit             50   21.86    0.44
Pure error               0    0.00

Stationary point of response surface:
     Temp      Time    Cation
 87.54535 105.86823  14.88488

> drop1(dye.rsm1)
                        Df Sum of Sq    RSS     AIC
<none>                               21.86 -40.581
FO(Temp, Time, Cation)   3    316.89 338.75 117.854
TWI(Temp, Time, Cation)  3      4.69  26.55 -34.909
PQ(Temp, Time, Cation)   3    238.98 260.84 102.174
```

Note that the first ANOVA table gives the sequential sums of squares, the second one (drop1) gives the partial sums of squares, each group given all the others. In this case, sequential makes more sense due to the hierarchy with main effects (FO = First order terms tested first). Higher order terms are (TWI = Two-Way Interactions and PQ = Polynomial Quadratic). Stationary point gives the best levels for each variable to maximize *Y*.

Contour Plots for all Pairs of Variables (at Best Level of Third Variable):







Data (Split into 3 Groups of Columns for Readability)

Temp	Time	Cation	KS	Temp	Time	Cation	KS	Temp	Time	Cation	KS
35	30	0	1.27	60	30	0	8.32	100	30	0	11.02
35	30	4	2.29	60	30	4	11.66	100	30	4	13.63
35	30	10	2.40	60	30	10	12.24	100	30	10	14.40
35	30	15	2.43	60	30	15	12.78	100	30	15	14.45
35	30	20	2.44	60	30	20	12.78	100	30	20	14.50
35	60	0	1.66	60	60	0	9.25	100	60	0	11.26
35	60	4	4.30	60	60	4	11.75	100	60	4	14.32
35	60	10	4.67	60	60	10	12.65	100	60	10	15.35
35	60	15	3.80	60	60	15	12.88	100	60	15	15.60
35	60	20	3.83	60	60	20	12.90	100	60	20	15.07
35	90	0	2.22	60	90	0	9.82	100	90	0	11.64
35	90	4	5.01	60	90	4	12.31	100	90	4	15.08
35	90	10	5.86	60	90	10	12.78	100	90	10	15.90
35	90	15	5.89	60	90	15	12.90	100	90	15	15.92
35	90	20	5.90	60	90	20	12.94	100	90	20	15.93
35	120	0	2.21	60	120	0	9.81	100	120	0	11.65
35	120	4	5.02	60	120	4	12.30	100	120	4	15.07
35	120	10	5.85	60	120	10	12.79	100	120	10	15.89
35	120	15	5.89	60	120	15	12.90	100	120	15	15.91
35	120	20	5.89	60	120	20	12.93	100	120	20	15.92

Qualitative Predictors

- Often, we wish to include categorical variables as predictors (e.g. gender, region of country, …)
- Trick: Create dummy (indicator) variable(s) to represent effects of levels of the categorical variables on response
- Problem: If variable has *c* categories, and we create *c* dummy variables, the model is not full rank when we include intercept
- Solution: Create $c - 1$ dummy variables, leaving one level as the control/baseline/reference category
- Interactions can be generated between qualitative and quantitative predictors
- Many models will contain multiple qualitative predictors.

Example – Salary vs Experience by Region – Why create *c*-1 Dummy Variables (not *c*)

State has $c = 3$ regions, sample 2 people from each region, Y = Salary, X_1 = Experience.

$$X_2 = \begin{cases} 1 \text{ if Region 1} \\ 0 \text{ otherwise} \end{cases} \quad X_3 = \begin{cases} 1 \text{ if Region 2} \\ 0 \text{ otherwise} \end{cases} \quad X_4 = \begin{cases} 1 \text{ if Region 3} \\ 0 \text{ otherwise} \end{cases}$$

$$\mathbf{X} = \begin{bmatrix} 1 & X_{11} & 1 & 0 & 0 \\ 1 & X_{21} & 1 & 0 & 0 \\ 1 & X_{31} & 0 & 1 & 0 \\ 1 & X_{41} & 0 & 1 & 0 \\ 1 & X_{51} & 0 & 0 & 1 \\ 1 & X_{61} & 0 & 0 & 1 \end{bmatrix} \Rightarrow \mathbf{X'X} = \begin{bmatrix} 6 & \sum_i X_{i1} & 2 & 2 & 2 \\ \sum_i X_{i1} & \sum_i X_{i1}^2 & X_{11}+X_{21} & X_{31}+X_{41} & X_{51}+X_{61} \\ 2 & X_{11}+X_{21} & 2 & 0 & 0 \\ 2 & X_{31}+X_{41} & 0 & 2 & 0 \\ 2 & X_{51}+X_{61} & 0 & 0 & 2 \end{bmatrix}$$

Problem: Last 3 columns (and rows) of **X'X** add up to first column (row), thus it is not full rank, and $(\mathbf{X'X})^{-1}$ does not exist, and there is not a unique estimator of $\boldsymbol{\beta}$. Trick: Delete either one of last 3 columns of **X** (and thus the same row and column of **X'X**). Note that internally, R will delete the first of those columns, making the first level of a qualitative (aka factor) variable the baseline (reference) level.

Solution to the problem:

3 regions, Y=salary, X_1 = experience $\quad X_2 = \begin{cases} 1 \text{ if Region 2} \\ 0 \text{ otherwise} \end{cases} \quad X_3 = \begin{cases} 1 \text{ if Region 3} \\ 0 \text{ otherwise} \end{cases}$

$$E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$$

Region 1: $X_2 = 0, X_3 = 0 \Rightarrow E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2(0) + \beta_3(0) = \beta_0 + \beta_1 X_1$

Region 2: $X_2 = 1, X_3 = 0 \Rightarrow E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2(1) + \beta_3(0) = (\beta_0 + \beta_2) + \beta_1 X_1$

Region 3: $X_2 = 0, X_3 = 1 \Rightarrow E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2(0) + \beta_3(1) = (\beta_0 + \beta_3) + \beta_1 X_1$

$\beta_2 \equiv$ Difference between Regions 2 and 1, controlling for experience

$\beta_3 \equiv$ Difference between Regions 3 and 1, controlling for experience

$\beta_2 - \beta_3 \equiv$ Difference between Regions 2 and 3, controlling for experience

$\beta_2 = \beta_3 = 0 \Rightarrow$ No differences among Regions 1,2,3 wrt Salary, Controlling for Experience

Example – Organic Soil Subsidence as Function of Water Table Level and Crop Type in Everglades

Source: S.F. Shih and W.F.P. Shih (1978). "Use of Dummy Variables in Water Resources Studies," *Journal of Hydrology*, Vol. 38, pp. 289-298.

Data were collected, relating annual organic soil subsidence (Y, in cm) in the Everglades to Water Table Level (X_1, in cm) for $c = 3$ crops (Pasture, Truck Crop ($X_2 = 1$), and Sugarcane ($X_3 = 1$)), each observed over 8 years (all observations assumed independent for this analysis).

Data:

Crop	subsid	watertab	truck	sugar
1	1.90	30.5	0	0
1	2.88	43.0	0	0
1	4.08	56.7	0	0
1	4.16	57.6	0	0
1	5.23	71.9	0	0
1	5.15	77.7	0	0
1	5.56	80.8	0	0
1	5.44	80.8	0	0
2	1.94	32.1	1	0
2	2.76	46.3	1	0
2	4.00	58.2	1	0
2	4.12	60.4	1	0
2	5.03	70.4	1	0
2	4.98	78.9	1	0
2	5.64	78.9	1	0
2	4.98	79.9	1	0
3	1.48	35.7	0	1
3	2.35	51.5	0	1
3	3.12	62.5	0	1
3	2.97	67.4	0	1
3	3.50	78.3	0	1
3	4.49	83.8	0	1
3	4.00	86.9	0	1
3	3.91	86.0	0	1

Y = subsid

X_1 = watertab

X_2 = truck

X_3 = sugar

Testing for crop effect, controlling for water table level:

$H_0 : \beta_2 = \beta_3 = 0$ (No Crop Effect) $\quad H_A : \beta_2$ and/or $\beta_3 \neq 0$

Full Model: $E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$

Reduced Model: $E\{Y\} = \beta_0 + \beta_1 X_1$

Full Model

	df	*SS*
Regression	3	33.83
Residual	20	1.85
Total	23	35.69

	Coefficients	*Standard Err*	*t Stat*	*P-value*
Intercept	0.3153	0.2521	1.2507	0.2255
watertab	0.0639	0.0037	17.4760	0.0000
truck	-0.1675	0.1522	-1.1000	0.2844
sugar	-1.4965	0.1541	-9.7095	0.0000

Reduced Model

	df	*SS*
Regression	1	23.39
Residual	22	12.29
Total	23	35.69

	Coefficients	*Standard Err*	*t Stat*	*P-value*
Intercept	0.1930	0.5933	0.3254	0.7480
watertab	0.0572	0.0088	6.4703	0.0000

Full Model: $\hat{Y}_F = 0.3153 + 0.0639X_1 - 0.1675X_2 - 1.4965X_3$

Reduced Model: $\hat{Y}_R = 0.1930 + 0.0572X_1$

$SSE(F) = 1.85 \quad df_F = 24 - 4 = 20 \quad SSE(R) = 12.29 \quad df_R = 24 - 2 = 22$

$$TS: \quad F^* = \frac{\left[\frac{SSE(R) - SSE(F)}{df_R - df_F}\right]}{\left[\frac{SSE(F)}{df_F}\right]} = \frac{\left[\frac{12.29 - 1.85}{22 - 20}\right]}{\left[\frac{1.85}{20}\right]} = \frac{5.22}{0.0925} = 56.43 \qquad F^* \geq F\left(0.95; 2, 20\right) = 3.493$$

$$P - \text{value} = \quad P\left(F\left(2, 20\right) \geq 56.43\right) = .0000$$

There is strong evidence of crop effect differences on organic soil subsidence, controlling for water table levels.

Interactions Between Qualitative and Quantitative Predictors

- We can allow the slope with respect to a Quantitative Predictor to differ across levels of the Categorical Predictor
- Trick: Create cross-product terms between Quantitative Predictor and each of the *c*-1 dummy variables
- Can conduct General Linear Test to determine whether slopes differ (or t-test when qualitative predictor has *c*=2 levels)
- These models generalize to any number of quantitative and qualitative predictors

Salary (Y), Expediture (X_1), and regions (X_2, X_3):

Additive Model: $E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$

Interaction Model: $E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_1 X_2 + \beta_5 X_1 X_3$

Region 1: $\left(X_2 = 0, X_3 = 0\right): E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2(0) + \beta_3(0) + \beta_4 X_1(0) + \beta_5 X_1(0) = \beta_0 + \beta_1 X_1$

Region 2: $\left(X_2 = 1, X_3 = 0\right): E\{Y\} = \left(\beta_0 + \beta_2\right) + \left(\beta_1 + \beta_4\right) X_1$

Region 3: $\left(X_2 = 0, X_3 = 1\right): E\{Y\} = \left(\beta_0 + \beta_3\right) + \left(\beta_1 + \beta_5\right) X_1$

Example – Organic Soil Subsidence as Function of Water Table Level and Crop Type in Everglades

Model fit with interactions between watertable*truck and watertable*sugar:

$H_0 : \beta_4 = \beta_5 = 0$ (No Water/Crop Interaction) $\quad H_A : \beta_4$ and/or $\beta_5 \neq 0$

Full Model: $E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_1 X_2 + \beta_5 X_1 X_3$

Reduced Model: $E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$

Interaction Model

	df	*SS*
Regressio	5	34.43
Residual	18	1.26
Total	23	35.69

	Coefficients	*Standard Err*	*t Stat*	*P-value*
Intercept	-0.0781	0.3463	-0.2256	0.8240
watertab	0.0702	0.0053	13.1303	0.0000
truck	-0.2213	0.5109	-0.4331	0.6701
sugar	-0.2195	0.5188	-0.4231	0.6773
wt*truck	0.0008	0.0079	0.0987	0.9224
wt*sugar	-0.0191	0.0076	-2.5079	0.0219

Full Model: $\hat{Y}_F = -0.0781 + 0.0702 X_1 - 0.2213 X_2 - 0.2195 X_3 + 0.0008 X_1 X_2 - 0.0191 X_1 X_3$

Reduced Model: $\hat{Y}_R = 0.3153 + 0.0639 X_1 - 0.1675 X_2 - 1.4965 X_3$

$SSE(F) = 1.26 \quad df_F = 24 - 6 = 18 \quad SSE(R) = 1.85 \quad df_R = 24 - 4 = 20$

$$TS: \quad F^* = \frac{\left[\frac{SSE(R) - SSE(F)}{df_R - df_F}\right]}{\left[\frac{SSE(F)}{df_F}\right]} = \frac{\left[\frac{1.85 - 1.26}{20 - 18}\right]}{\left[\frac{1.26}{18}\right]} = \frac{0.295}{0.070} = 4.214 \qquad F^* \geq F\left(0.95; 2, 18\right) = 3.555$$

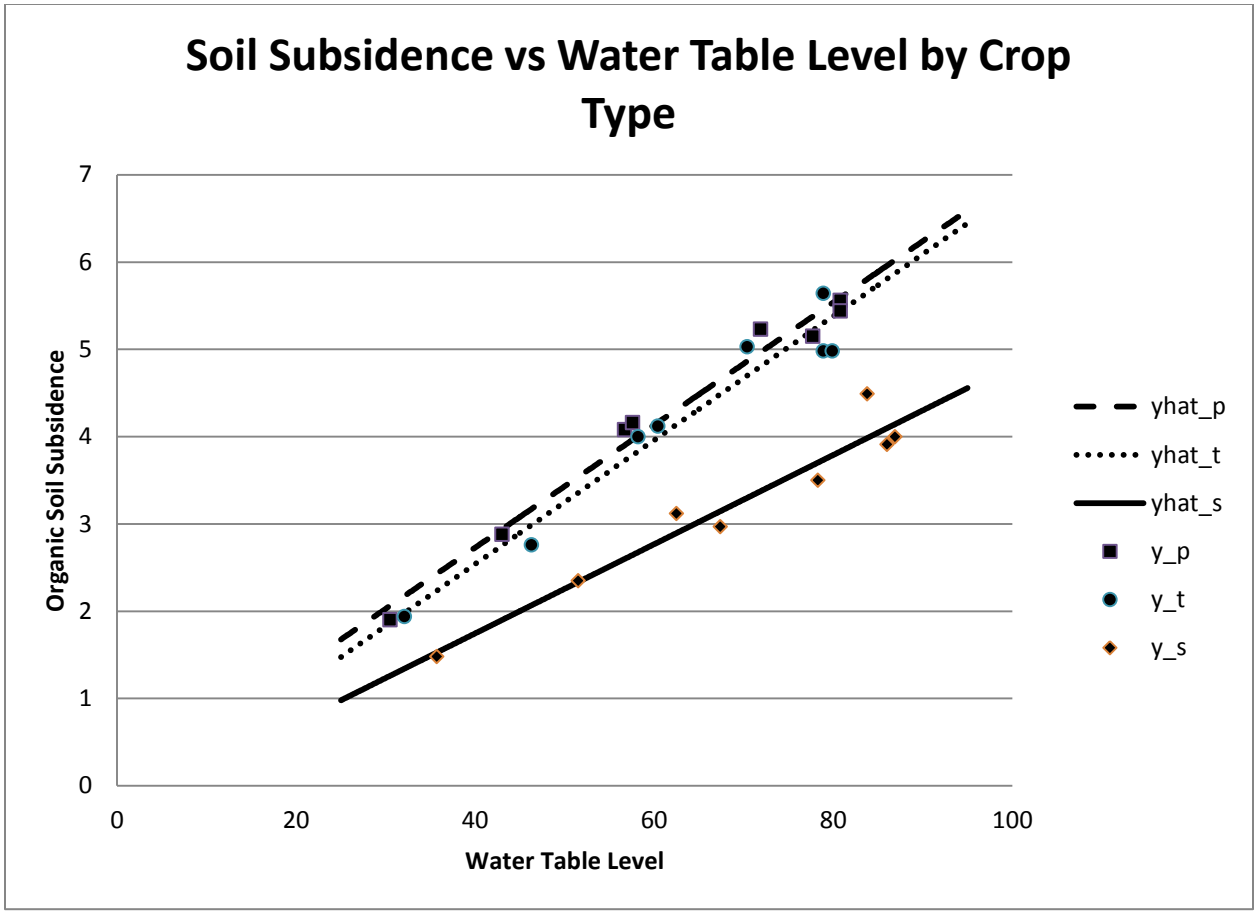
$P - \text{value} = \quad P\left(F\left(2, 18\right) \geq 4.214\right) = .0315$

Fitted Equation for each Crop Type:

Pasture: $X_2 = 0, X_3 = 0 : \hat{Y}_P = -0.0781 + 0.0702 X_1$

Truck Crop: $X_2 = 1, X_3 = 0 : \hat{Y}_T = \left(-0.0781 - 0.2213\right) + \left(0.0702 + 0.0008\right) X_1 = -0.2994 + 0.0710 X$

Sugar: $X_2 = 0, X_3 = 1 : \hat{Y}_T = \left(-0.0781 - 0.2195\right) + \left(0.0702 - 0.0191\right) X_1 = -0.2976 + 0.0511 X$



R Program

```
water <- read.csv("E:\\blue_drive\\sta4210\\water_resource_dumvarreg.csv", header=TRUE)
attach(water); names(water)

water.mod1 <- lm(subsid ~ watertab)
summary(water.mod1)
anova(water.mod1)

water.mod2 <- lm(subsid ~ watertab + truck + sugar)
summary(water.mod2)
anova(water.mod2)

anova(water.mod1,water.mod2)

water.mod3 <- lm(subsid ~ watertab + truck + sugar + I(watertab*truck) + I(watertab*sugar))
summary(water.mod3)
anova(water.mod3)

anova(water.mod2,water.mod3)
```

R Output

```
> summary(water.mod1)

Call:
lm(formula = subsid ~ watertab)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.193047   0.593322   0.325    0.748
watertab    0.057214   0.008843   6.470 1.65e-06 ***


Residual standard error: 0.7475 on 22 degrees of freedom
Multiple R-squared:  0.6555,    Adjusted R-squared:  0.6399
F-statistic: 41.87 on 1 and 22 DF,  p-value: 1.649e-06

> anova(water.mod1)
Analysis of Variance Table

Response: subsid
          Df Sum Sq Mean Sq F value    Pr(>F)
watertab   1 23.393 23.3931  41.865 1.649e-06 ***
Residuals 22 12.293  0.5588

> summary(water.mod2)

Call: lm(formula = subsid ~ watertab + truck + sugar)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.315346   0.252133   1.251    0.225
watertab     0.063882   0.003655  17.476 1.39e-13 ***
truck       -0.167460   0.152234  -1.100    0.284
sugar       -1.496518   0.154130  -9.709 5.18e-09 ***


Residual standard error: 0.3044 on 20 degrees of freedom
Multiple R-squared:  0.9481,    Adjusted R-squared:  0.9403
F-statistic: 121.7 on 3 and 20 DF,  p-value: 5.157e-13

> anova(water.mod2)
Analysis of Variance Table

Response: subsid
          Df  Sum Sq Mean Sq F value    Pr(>F)
watertab   1 23.3931 23.3931 252.436 8.292e-13 ***
truck      1  1.7033  1.7033  18.381 0.0003594 ***
sugar      1  8.7363  8.7363  94.274 5.182e-09 ***
Residuals 20  1.8534  0.0927
>
> anova(water.mod1,water.mod2)
Analysis of Variance Table

Model 1: subsid ~ watertab
Model 2: subsid ~ watertab + truck + sugar
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1     22 12.2930
2     20  1.8534  2     10.44 56.327 6.068e-09 ***
```

Continued Below

```
> summary(water.mod3)

Call:
lm(formula = subsid ~ watertab + truck + sugar + I(watertabtruck) + I(watertab * sugar))

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         -0.078139   0.346318  -0.226   0.8240
watertab             0.070191   0.005346  13.130 1.17e-10 ***
truck               -0.221266   0.510937  -0.433   0.6701
sugar               -0.219489   0.518799  -0.423   0.6773
I(watertab * truck)  0.000776   0.007860   0.099   0.9224
I(watertab * sugar) -0.019111   0.007620  -2.508   0.0219 *


Residual standard error: 0.2647 on 18 degrees of freedom
Multiple R-squared:  0.9647,    Adjusted R-squared:  0.9549
F-statistic:  98.3 on 5 and 18 DF,  p-value: 2.014e-12

> anova(water.mod3)
Analysis of Variance Table

Response: subsid
                    Df  Sum Sq Mean Sq  F value    Pr(>F)
watertab             1 23.3931 23.3931 333.9966 4.541e-13 ***
truck                1  1.7033  1.7033  24.3195 0.0001077 ***
sugar                1  8.7363  8.7363 124.7328 1.589e-09 ***
I(watertab * truck)  1  0.1521  0.1521   2.1722 0.1578035
I(watertab * sugar)  1  0.4405  0.4405   6.2897 0.0219434 *
Residuals           18  1.2607  0.0700

>
> anova(water.mod2,water.mod3)
Analysis of Variance Table

Model 1: subsid ~ watertab + truck + sugar
Model 2: subsid ~ watertab + truck + sugar + I(watertab * truck) + I(watertab *
    sugar)
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     20 1.8534
2     18 1.2607  2   0.59267 4.2309 0.03118 *
```

Chapter 9 – Model Building and Validation

Data Collection Strategies

- Controlled Experiments – Subjects (Experimental Units) assigned to *X*-levels by Experimenter
  - Purely Controlled Experiments – Researcher only uses predictors that were assigned to units
  - Controlled Experiments with Covariates – Researcher has information (additional predictors) associated with units

- Observational Studies – Subjects (Units) have *X*-levels associated with them (not assigned by researcher)
  - Confirmatory Studies – New (primary) predictor(s) believed to be associated with *Y*, controlling for (control) predictor(s), known to be associated with *Y*
  - Exploratory Studies – Set of potential predictors believed that some or all are associated with *Y*

Reduction of Explanatory Variables

- Controlled Experiments
  - Purely Controlled Experiments – Rarely any need or desire to reduce number of explanatory variables
  - Controlled Experiments with Covariates – Remove any covariates that do not reduce the error variance

- Observational Studies
  - Confirmatory Studies – Must keep in all control variables to compare with previous research, should keep all primary variables as well
  - Exploratory Studies – Often have many potential predictors (and polynomials and interactions). Want to fit parsimonious model that explains much of the variation in Y, while keeping model as basic as possible. Caution: do not make decisions based on single variable t-tests, make use of Complete/Reduced models for testing multiple predictors

Model Selection Criteria – All Possible Regressions

$P-1$ predictors $\Rightarrow$ 2^{P-1} potential models (each variable can be in or out of model)

R_p^2 or SSE_p criterion (Goal: find p so that $\max\left(R_p^2\right)$ or $\min\left(SSE_p\right)$ "flattens out"):

$R_p^2 = \dfrac{SSR_p}{SSTO} = 1 - \dfrac{SSE_p}{SSTO}$ $\quad p = \#$ of parameters in current model

$R_{a,p}^2$ or MSE_p criterion (Goal: find model that maximizes (or close to) $R_{a,p}^2$ and minimizes MSE_p):

$$R_{a,p}^2 = 1 - \left(\frac{n-1}{n-p}\right)\frac{SSE_p}{SSTO} = 1 - \frac{\left(SSE_p/(n-p)\right)}{\left(SSTO/(n-1)\right)} = 1 - \frac{MSE_p}{\left(SSTO/(n-1)\right)}$$

Mallow's C_p criterion (Goal: find model with smallest p so that $C_p \le p$):

$$C_p = \frac{SSE_p}{MSE\left(X_1,...,X_{P-1}\right)} - (n-2p)$$

AIC_p and SBC_p criteria (Goal: choose model that minimize these values):

$AIC_p = n\ln\left(SSE_p\right) - n\ln(n) + 2p$ $\qquad$ $SBC_p = n\ln\left(SSE_p\right) - n\ln(n) + \left[\ln(n)\right]p$

$PRESS_p$ criterion (Goal: Small values):

$PRESS_p = \sum_{i=1}^{n}\left(Y_i - \hat{Y}_{i(i)}\right)^2$ $\qquad$ $\hat{Y}_{i(i)} \equiv$ fitted value for i^{th} case when it was not used in fitting model

This Can be obtained without re-fitting the model n times: $PRESS_p = \sum_{i=1}^{n}\left(\dfrac{e_i}{1-h_{ii}}\right)^2$

Example: Construction Cost of Hong Kong Office Buildings

Source: H. Li, Q.P. Shen, P.E.D. Love (2005). "Cost Modelling of Office Buildings in Hong Kong: An Exploratory Study," *Facilities*, Vol. 23, #9/10, pp. 438-452.

Response variable: Adjusted Construction Cost (100 Millions of HK$).

Potential Predictors: Average Floor Area (X_1, in m^2), Total Floor Area (X_2, in m^2), Average Storey Height (X_3, in m), and Steel Dummy Variable (X_4=1 if Steel, 0 if Reinforced Concrete). There are 4 one variable models, 6 two variable models, 4 three variable models, and 1 four variable model. There are n = 37 buildings in the sample, and all models contain an intercept. Results for each model are given below:

$$SSTO = \sum_i \left(Y_i - \overline{Y}\right)^2 = 873.3359$$

Predictors	p	SSE	R^2	Adj-R^2	C_p	AIC	SBC
X1	2	41.2140	0.9528	0.9541	373.61	7.99	11.21
X2	2	24.9003	0.9715	0.9723	212.66	-10.65	-7.43
X3	2	415.4651	0.5243	0.5375	4065.86	93.48	96.71
X4	2	867.5796	0.0066	0.0342	8526.29	120.73	123.95
X1,X2	3	4.1598	0.9952	0.9955	10.04	-74.86	-70.03
X1,X3	3	41.1994	0.9528	0.9554	375.46	9.98	14.81
X1,X4	3	41.0281	0.9530	0.9556	373.77	9.82	14.66
X2,X3	3	17.6364	0.9798	0.9809	143.00	-21.42	-16.58
X2,X4	3	23.5145	0.9731	0.9746	200.99	-10.77	-5.94
X3,X4	3	412.2314	0.5280	0.5542	4035.95	95.19	100.03
X1,X2,X3	4	3.5714	0.9959	0.9963	6.23	-78.50	-72.06
X1,X2,X4	4	3.8178	0.9956	0.9960	8.66	-76.04	-69.59
X1,X3,X4	4	41.0097	0.9530	0.9570	375.59	11.81	18.25
X2,X3,X4	4	16.6652	0.9809	0.9825	135.41	-21.51	-15.07
X1,X2,X3,X4	5	3.2436	0.9963	0.9967	5.00	-80.07	-72.01

The calculations for the first model (X_1) are given here:

$$R_p^2\left(X_1\right) = 1 - \frac{SSE\left(X_1\right)}{SSTO} = 1 - \frac{41.2140}{873.3359} = 0.9528$$

$$R_{a,p}^2\left(X_1\right) = 1 - \left(\frac{n-1}{n-p}\right)\left(\frac{SSE\left(X_1\right)}{SSTO}\right) = 1 - \left(\frac{37-1}{37-2}\right)\left(\frac{41.2140}{873.3359}\right) = 0.9515$$

$$MSE\left(X_1, X_2, X_3, X_4\right) = \frac{SSE\left(X_1, X_2, X_3, X_4\right)}{n-P} = \frac{3.2436}{37-5} = 0.1013625$$

$$\Rightarrow \quad C_p\left(X_1\right) = \frac{SSE\left(X_1\right)}{MSE\left(X_1, X_2, X_3, X_4\right)} - \left(n-p\right) = \frac{41.2140}{0.1013625} - \left(37 - 2(2)\right) = 373.61$$

$$AIC_p\left(X_1\right) = \ln\left[SSE\left(X_1\right)\right] - n\ln(n) + 2p = \ln(41.2140) - 37\ln(37) + 2(2) = 7.99$$

$$SBC_p\left(X_1\right) = \ln\left[SSE\left(X_1\right)\right] - n\ln(n) + \ln(n)p = \ln(41.2140) - 37\ln(37) + \ln(37)(2) = 11.21$$

All methods, with the exception of SBC favor the full model, with all four predictors. SBC, which places a higher penalty on extra parameters, when $\ln(n) > 2$. SBC chooses the model with Average Floor Area, Total Floor Area, and Average Storey Height. It eliminates the Steel dummy variable. Either model explains virtually all of the variation in Construction costs. Note that the model with the two Floor Area measures also has R^2=.9952.

R Program and Output – All Possible Regressions

```
hkbuild <- read.csv("E:\\blue_drive\\sta4210\\hk_build_cost.csv", header=TRUE)
attach(hkbuild); names(hkbuild)

install.packages("leaps")     # Must have "set mirror" in R
library(leaps)

allpossreg <- regsubsets(cost100m ~ avearea+totarea+aveht+steel,nbest=6,data=hkbuild)
aprout <- summary(allpossreg)

with(aprout,round(cbind(which,rsq,adjr2,cp,bic),3))  ### AIC is not an option




####### Output

> allpossreg <- regsubsets(cost100m ~ avearea+totarea+aveht+steel,nbest=6,data=hkbuild)
> aprout <- summary(allpossreg)
>
> with(aprout,round(cbind(which,rsq,adjr2,cp,bic),3))  ### AIC is not an option
  (Intercept) avearea totarea aveht steel   rsq  adjr2       cp      bic
1           1       0       1     0     0 0.971  0.971  212.659 -124.403
1           1       1       0     0     0 0.953  0.951  373.606 -105.759
1           1       0       0     1     0 0.524  0.511 4065.857  -20.266
1           1       0       0     0     1 0.007 -0.022 8526.285    6.977
2           1       1       1     0     0 0.995  0.995   10.039 -187.001
2           1       0       1     1     0 0.980  0.979  142.995 -133.554
2           1       0       1     0     1 0.973  0.971  200.987 -122.911
2           1       1       0     0     1 0.953  0.950  373.771 -102.316
2           1       1       0     1     0 0.953  0.950  375.461 -102.161
2           1       0       0     1     1 0.528  0.500 4035.954  -16.944
3           1       1       1     1     0 0.996  0.996    6.235 -189.032
3           1       1       1     0     1 0.996  0.995    8.665 -186.565
3           1       0       1     1     1 0.981  0.979  135.414 -132.039
3           1       1       0     1     1 0.953  0.949  375.590  -98.721
4           1       1       1     1     1 0.996  0.996    5.000 -188.984
```

Comments

- Makes use of the **leaps** R package, which must be downloaded.
- The first column gives the number of predictors (p-1), not the number of parameters (p).
- The first row corresponds to the model with an intercept and totarea (X_2)
- `nbest = 6` tells the program to print out the best 6 models for each possible number of parameters. For p-1=4, this prints out all models, as p-1 gets larger, you don't want to print out all cases.
- The `with(aprout,round(cbind(which,rsq,adjr2,cp,bic),3))` command prints out the output in readable form, with 3 decimal places.

R Program – PRESS Statistic

```
hkbuild <- read.csv("E:\\blue_drive\\sta4210\\hk_build_cost.csv",
       header=TRUE)
attach(hkbuild); names(hkbuild)

hkb.X1 <- lm(cost100m ~ avearea)
PRESS.statistic <- sum( (resid(hkb.X1)/(1-hatvalues(hkb.X1)))^2 )
print(paste("X1 PRESS statistic= ", PRESS.statistic))

hkb.X2 <- lm(cost100m ~ totarea)
PRESS.statistic <- sum( (resid(hkb.X2)/(1-hatvalues(hkb.X2)))^2 )
print(paste("X2 PRESS statistic= ", PRESS.statistic))

hkb.X3 <- lm(cost100m ~ aveht)
PRESS.statistic <- sum( (resid(hkb.X3)/(1-hatvalues(hkb.X3)))^2 )
print(paste("X3 PRESS statistic= ", PRESS.statistic))

hkb.X4 <- lm(cost100m ~ steel)
PRESS.statistic <- sum( (resid(hkb.X4)/(1-hatvalues(hkb.X4)))^2 )
print(paste("X4 PRESS statistic= ", PRESS.statistic))

hkb.X1X2 <- lm(cost100m ~ avearea + totarea)
PRESS.statistic <- sum( (resid(hkb.X1X2)/(1-hatvalues(hkb.X1X2)))^2 )
print(paste("X1X2 PRESS statistic= ", PRESS.statistic))

hkb.X1X3 <- lm(cost100m ~ avearea + aveht)
PRESS.statistic <- sum( (resid(hkb.X1X3)/(1-hatvalues(hkb.X1X3)))^2 )
print(paste("X1X3 PRESS statistic= ", PRESS.statistic))

hkb.X1X4 <- lm(cost100m ~ avearea + steel)
PRESS.statistic <- sum( (resid(hkb.X1X4)/(1-hatvalues(hkb.X1X4)))^2 )
print(paste("X1X4 PRESS statistic= ", PRESS.statistic))

hkb.X2X3 <- lm(cost100m ~ totarea + aveht)
PRESS.statistic <- sum( (resid(hkb.X2X3)/(1-hatvalues(hkb.X2X3)))^2 )
print(paste("X2X3 PRESS statistic= ", PRESS.statistic))

hkb.X2X4 <- lm(cost100m ~ totarea + steel)
PRESS.statistic <- sum( (resid(hkb.X2X4)/(1-hatvalues(hkb.X2X4)))^2 )
print(paste("X2X4 PRESS statistic= ", PRESS.statistic))

hkb.X3X4 <- lm(cost100m ~ aveht + steel)
PRESS.statistic <- sum( (resid(hkb.X3X4)/(1-hatvalues(hkb.X3X4)))^2 )
print(paste("X3X4 PRESS statistic= ", PRESS.statistic))

hkb.X1X2X3 <- lm(cost100m ~ avearea + totarea + aveht)
PRESS.statistic <- sum( (resid(hkb.X1X2X3)/(1-hatvalues(hkb.X1X2X3)))^2 )
print(paste("X1X2X3 PRESS statistic= ", PRESS.statistic))

hkb.X1X2X4 <- lm(cost100m ~ avearea + totarea + steel)
PRESS.statistic <- sum( (resid(hkb.X1X2X4)/(1-hatvalues(hkb.X1X2X4)))^2 )
print(paste("X1X2X4 PRESS statistic= ", PRESS.statistic))

hkb.X1X3X4 <- lm(cost100m ~ avearea + aveht + steel)
PRESS.statistic <- sum( (resid(hkb.X1X3X4)/(1-hatvalues(hkb.X1X3X4)))^2 )
print(paste("X1X3X4 PRESS statistic= ", PRESS.statistic))

hkb.X2X3X4 <- lm(cost100m ~ totarea + aveht + steel)
PRESS.statistic <- sum( (resid(hkb.X2X3X4)/(1-hatvalues(hkb.X2X3X4)))^2 )
print(paste("X2X3X4 PRESS statistic= ", PRESS.statistic))

hkb.X1X2X3X4 <- lm(cost100m ~ avearea + totarea + aveht + steel)
PRESS.statistic <- sum( (resid(hkb.X1X2X3X4)/(1-hatvalues(hkb.X1X2X3X4)))^2 )
print(paste("X1X2X3X4 PRESS statistic= ", PRESS.statistic))
```

R Output – PRESS Statistic

```
[1] "X1 PRESS statistic=  58.9285847132438"
>
[1] "X2 PRESS statistic=  34.3068459264399"
>
[1] "X3 PRESS statistic=  480.232597571876"
>
[1] "X4 PRESS statistic=  974.535465360164"
>
[1] "X1X2 PRESS statistic=  5.36692153722167"
>
[1] "X1X3 PRESS statistic=  61.5334300336164"
>
[1] "X1X4 PRESS statistic=  62.0853136828898"
>
[1] "X2X3 PRESS statistic=  28.8866429223542"
>
[1] "X2X4 PRESS statistic=  33.9044260157386"
>
[1] "X3X4 PRESS statistic=  502.052047289793"
>
[1] "X1X2X3 PRESS statistic=  4.81054783566282"
>
[1] "X1X2X4 PRESS statistic=  5.16226362282806"
>
[1] "X1X3X4 PRESS statistic=  64.8572417164229"
>
[1] "X2X3X4 PRESS statistic=  28.3305394700148"
>
[1] "X1X2X3X4 PRESS statistic=  4.57715937848567"
```

Comments

- The PRESS statistic is minimized for the full model (X_1, X_2, X_3, X_4) with PRESS = 4.5772
- Also relatively small for: (X_1, X_2, X_3), (X_1, X_2, X_4), and (X_1, X_2)
- PRESS can also be used for Model Validation (see below)

Regression Model Building

- Setting: Possibly a large set of predictor variables (including interactions and polynomials).
- Goal: Fit a parsimonious model that explains variation in Y with a small set of predictors
- Automated Procedures and all possible regressions:
  - Backward Elimination (Top down approach)
  - Forward Selection (Bottom up approach)
  - Stepwise Regression (Combines Forward/Backward)

Backward Elimination - Traditional Approach

- Select a significance level to stay in the model (e.g. SLS=0.20, generally .05 is too low, causing too many variables to be removed)
- Fit the full model with all possible predictors
- Consider the predictor with lowest *t*-statistic (highest *P*-value).
    - If $P >$ SLS, remove the predictor and fit model without this variable (must re-fit model here because partial regression coefficients change)
    - If $P \leq$ SLS, stop and keep current model
- Continue until all predictors have *P*-values below SLS
- Note: R uses model based criteria: AIC, SBC instead, which does not require choosing a level for SLS:
    - Fit the full model with all possible predictors.
    - Fit all models with one predictor removed.
    - Consider the predictor with the LOWEST AIC.
        - If AIC is lower when the predictor has been removed than when it is in model, remove it.
        - If AIC is higher when the predictor has been removed than when it is in model, stop and keep the current model.
    - Continue until no variables can be eliminated.

Forward Selection – Traditional Approach

- Choose a significance level to enter the model (e.g. SLE=0.20, generally .05 is too low, causing too few variables to be entered)
- Fit all simple regression models.
- Consider the predictor with the highest *t*-statistic (lowest *P*-value)
    - If $P \leq$ SLE, keep this variable and fit all two variable models that include this predictor
    - If $P >$ SLE, stop and keep previous model
- Continue until no new predictors have $P \leq$ SLE
- Note: R uses model based criteria: AIC, SBC instead, which does not require choosing a level for SLE:
    - Obtain AIC (or SBC) for the null model, with no predictors *(SSE = SSTO*)
    - Fit all models with one variable
    - Consider the model with the LOWEST AIC (or SBC, assuming AIC below, same if SBC used)

- If AIC is lower when the predictor has been added than when it is not in model, keep it.
- If AIC is higher when the predictor has been added than when it is not in model, stop and keep the current model (without this predictor)

- Continue until no variables can be added.

Stepwise Regression – Traditional Approach

- Select SLS and SLE (SLE<SLS)
- Starts like Forward Selection (Bottom up process)
- New variables must have $P \leq$ SLE to enter
- Re-tests all "old variables" that have already been entered, must have $P \leq$ SLS to stay in model
- Continues until no new variables can be entered and no old variables need to be removed
- Note: R uses model based criteria: AIC, SBC instead, and does not need selection of SLS or SLE. It follows a similar algorithm as above, keeps checking variables that have previously been entered to see if they still contribute to the model.

Example: Aroma scores for Lager Beer Related to 3 higher alcohol levels and 4 estery compound levels

Source: I. Techakriengkrai, A. Paterson, B. Taidi, and J.R. Piggott (2006). "Relationships of Overall Estery Aroma Character in Lagers with Volatile Headspace Congener Concentrations," Journal of the Institute of Brewing, Vol. 112, #1, pp. 41-49.

Response Variable: Y = Estery Aroma Score
Predictor Variables (Concentrations): X_1 = Ethyl Acetate , X_2 = Propanol, X_3 = Isobutanol,
X_4 = Isoamyl Acetate, X_5 = Methyl Butanol , X_6 = Ethyl Caproate, X_7 = Ethyl Caprylate
Note: There are only n = 23 brands of beer in study, so this is more of a demonstration than decisive analysis.

R Program

```
aroma <- read.csv("E:\\blue_drive\\sta4210\\aroma_beer.csv",
       header=TRUE)
attach(aroma); names(aroma)

# These Stepwise Methods are based on Model Criteria, not individual regression coefficients
# direction="both" begins like backward and works down
# Criteria:  k=2 uses AIC (default)   k=log(length(y))  uses BIC

reg.full <- lm(Y ~ X1+X2+X3+X4+X5+X6+X7)
reg.null <- lm(Y ~ 1)
summary(reg.full); anova(reg.full); drop1(reg.full)

backward.reg <- step(reg.full,direction="backward",k=log(length(Y)))
summary(backward.reg)

forward.reg <- step(reg.null,direction="forward",scope=list(upper=reg.full,lower=reg.null))
summary(forward.reg)

stepwise.reg <- step(reg.full,direction="both")
summary(stepwise.reg)
```

R Output – Part 1 – Summary of Full Model

```
> summary(reg.full)

Call:
lm(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7)

Residuals:
    Min      1Q  Median      3Q     Max
-8.9722 -2.6885 -0.4587  3.5230 10.5647

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   53.2318     7.9678   6.681 7.34e-06 ***
X1            -0.1490     0.6501  -0.229    0.822
X2           -10.0862     6.7913  -1.485    0.158
X3             0.7951     0.6256   1.271    0.223
X4             2.1770     4.0779   0.534    0.601
X5            -0.3536     0.3499  -1.011    0.328
X6            98.4047    75.9735   1.295    0.215
X7          -383.5664   235.0144  -1.632    0.123
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.349 on 15 degrees of freedom
Multiple R-squared:  0.3884,    Adjusted R-squared:  0.103
F-statistic: 1.361 on 7 and 15 DF,  p-value: 0.2905

> anova(reg.full)
Analysis of Variance Table

Response: Y
          Df Sum Sq Mean Sq F value Pr(>F)
X1         1  72.32  72.318  1.7942 0.2004
X2         1  69.33  69.332  1.7201 0.2094
X3         1  62.19  62.193  1.5430 0.2332
X4         1  63.67  63.675  1.5798 0.2280
X5         1   8.87   8.866  0.2200 0.6458
X6         1   0.21   0.213  0.0053 0.9430
X7         1 107.37 107.366  2.6637 0.1235
Residuals 15 604.59  40.306
> drop1(reg.full)
Single term deletions

Model:
Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7
       Df Sum of Sq    RSS    AIC
<none>              604.59 91.188
X1      1     2.118 606.71 89.269
X2      1    88.904 693.50 92.344
X3      1    65.097 669.69 91.540
X4      1    11.487 616.08 89.621
X5      1    41.174 645.77 90.704
X6      1    67.621 672.22 91.627
X7      1   107.366 711.96 92.948
>
```

Note that overall, the model is not significant (P-value for the F-test is 0.2905). There are only n-p = 23-8 = 15 error degrees of freedom. Note that **anova(reg.full)** gives the sequential sums of squares: $SSR(X_1)$, $SSR(X_2|X_1)$,… and **drop1(reg.full)** gives the partial sums of squares: $SSR(X_1|X_2,\ldots,X_7)$,…. By definition the two types of sums of squares are the same for X_7.

R Output – Part 2 – Summary of Backward Elimination ("AIC" = SBC)

```
> backward.reg <- step(reg.full,direction="backward",k=log(length(Y)))
Start:  AIC=100.27
Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7

       Df Sum of Sq    RSS     AIC
- X1    1     2.118 606.71  97.217
- X4    1    11.487 616.08  97.570
- X5    1    41.174 645.77  98.652
- X3    1    65.097 669.69  99.489
- X6    1    67.621 672.22  99.575
<none>              604.59 100.272
- X2    1    88.904 693.50 100.292
- X7    1   107.366 711.96 100.897

Step:  AIC=97.22
Y ~ X2 + X3 + X4 + X5 + X6 + X7

       Df Sum of Sq    RSS    AIC
- X4    1    20.274 626.99 94.838
- X5    1    60.703 667.42 96.275
- X6    1    76.938 683.65 96.828
- X3    1    83.591 690.30 97.051
- X2    1    87.952 694.66 97.195
<none>              606.71 97.217
- X7    1   125.104 731.82 98.394

Step:  AIC=94.84
Y ~ X2 + X3 + X5 + X6 + X7

       Df Sum of Sq    RSS    AIC
- X5    1    45.835 672.82 93.325
- X3    1    77.629 704.62 94.387
<none>              626.99 94.838
- X2    1   111.808 738.79 95.477
- X7    1   144.368 771.35 96.469
- X6    1   152.121 779.11 96.699

Step:  AIC=93.33
Y ~ X2 + X3 + X6 + X7

       Df Sum of Sq    RSS    AIC
- X3    1    37.055 709.88 91.423
<none>              672.82 93.325
- X2    1   104.035 776.86 93.496
- X7    1   107.393 780.22 93.596
- X6    1   109.503 782.33 93.658

Step:  AIC=91.42
Y ~ X2 + X6 + X7

       Df Sum of Sq    RSS    AIC
- X2    1    74.976 784.85 90.597
<none>              709.88 91.423
- X7    1   148.696 858.57 92.661
- X6    1   226.241 936.12 94.650
```

Continued Below

```
Step:  AIC=90.6
Y ~ X6 + X7

       Df Sum of Sq    RSS    AIC
<none>               784.85 90.597
- X7    1    165.00 949.85 91.850
- X6    1    190.94 975.80 92.469
> summary(backward.reg)

Call:
lm(formula = Y ~ X6 + X7)

Residuals:
    Min      1Q  Median      3Q     Max
-9.7330 -4.1652 -0.9828  4.0525 13.9520

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   46.577      3.791  12.286 8.96e-11 ***
X6           112.868     51.168   2.206   0.0393 *
X7          -413.080    201.452  -2.051   0.0537 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.264 on 20 degrees of freedom
Multiple R-squared:  0.2061,    Adjusted R-squared:  0.1267
F-statistic: 2.595 on 2 and 20 DF,  p-value: 0.09951
```

Description of Output (Labelled AIC is actually SBC based on the “k” option)

- At the first step for the Full Model (<None> Removed), SBC = 100.272. All models, where one variable is removed, are fit. The smallest SBC is when X_1 is removed, with SBC = 97.217 < 100.272
- At the second step for the Full Model (<None> Removed), SBC = 97.217 (the value when X_1 is removed). All models, where one variable is removed, are fit. The smallest SBC is when X_4 is removed, with SBC = 94.838 < 97.217
- At the third step for the Full Model (<None> Removed), SBC = 94.838 (the value when X_1 and X_4 are removed). All models, where one variable is removed, are fit. The smallest SBC is when X_5 is removed, with SBC = 93.325 < 94.838
- At the fourth step for the Full Model (<None> Removed), SBC = 93.325 (the value when X_1, X_4, and X_5 are removed). All models, where one variable is removed, are fit. The smallest SBC is when X_3 is removed, with SBC = 91.423 < 93.325
- At the fifth step for the Full Model (<None> Removed), SBC = 91.423 (the value when X_1, X_3, X_4, and X_5 are removed). All models, where one variable is removed, are fit. The smallest SBC is when X_2 is removed, with SBC = 90.597 < 91.423
- At the fifth step for the Full Model (<None> Removed), SBC = 91.423 (the value when X_1, X_2, X_3, X_4, and X_5 are removed). All models, where one variable is removed, are fit. The smallest SBC is when X_7 is removed, with SBC = 91.850 > 90.597. Thus, neither X_6 or X_7 are removed.
- The results for the final regression model , containing X_6 = Ethyl Caproate, and X_7 = Ethyl Caprylate. Note that level of X_6 = Ethyl Caproate is positively related to aroma score, and X_7 = Ethyl Caprylate is negatively related to aroma score.

R Output – Part 3 – Summary of Forward Selection Using AIC (Default choice)

```
> forward.reg <- step(reg.null,direction="forward",scope=list(upper=reg.full,lower=reg.null))
Start:  AIC=88.5
Y ~ 1

       Df Sum of Sq    RSS    AIC
+ X4    1   118.072 870.48 87.572
<none>              988.56 88.497
+ X1    1    72.318 916.24 88.750
+ X3    1    61.628 926.93 89.017
+ X2    1    49.852 938.71 89.307
+ X5    1    47.238 941.32 89.371
+ X6    1    38.705 949.85 89.579
+ X7    1    12.762 975.80 90.198

Step:  AIC=87.57
Y ~ X4

       Df Sum of Sq    RSS    AIC
+ X2    1    76.671 793.81 87.451
<none>              870.48 87.572
+ X7    1    67.702 802.78 87.710
+ X1    1    31.633 838.85 88.720
+ X3    1     3.632 866.85 89.476
+ X6    1     3.161 867.32 89.488
+ X5    1     0.316 870.17 89.563

Step:  AIC=87.45
Y ~ X4 + X2

       Df Sum of Sq    RSS    AIC
<none>              793.81 87.451
+ X7    1    38.566 755.25 88.306
+ X3    1    36.897 756.92 88.356
+ X1    1    31.708 762.11 88.514
+ X5    1     7.900 785.91 89.221
+ X6    1     0.883 792.93 89.426
> summary(forward.reg)

Call:
lm(formula = Y ~ X4 + X2)

Residuals:
     Min       1Q   Median       3Q      Max
-10.3444  -4.5607  -0.0758   5.5401  10.3339

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   52.879      3.349  15.792 9.29e-13 ***
X4             2.206      1.155   1.911   0.0705 .
X2            -8.218      5.913  -1.390   0.1798
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.3 on 20 degrees of freedom
Multiple R-squared:  0.197,     Adjusted R-squared:  0.1167
F-statistic: 2.453 on 2 and 20 DF,  p-value: 0.1115
```

Description of Output

- At the first step for the Null Model (<None> Entered), AIC = 88.497. When each model with one variable is fit, the minimum AIC is when X_4 is entered, with AIC = 87.572 < 88.497

- At the second for the Null Model (<None> Entered), AIC = 87.572, (the value when X_4 was entered). When each model with one new variable is fit, the minimum AIC is when X_2 is entered, with AIC = 87.451 < 87.572.
- At the second for the Null Model (<None> Entered), AIC = 87.451, (the value when X_2 and X_4 were entered). When each model with one new variable is fit, the minimum AIC is when X_7 is entered, with AIC = 88.306 > 87.451. Thus no new variables are entered.

- The results for the final regression model , containing X_2 = Propanol, and X_4 = Isoamyl Acetate. Note that level of X_4 = Isoamyl Acetate is positively related to aroma score, and X_2 = Propanol is negatively related to aroma score.
- Note that with this small dataset (and also using different criteria (SBC and AIC, respectively), backward elimination and forward selection give quite different results.

R Output – Part 4 – Summary of Stepwise Regression Using AIC (Default choice)

```
>
> stepwise.reg <- step(reg.full,direction="both")
Start:  AIC=91.19
Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7

       Df Sum of Sq    RSS    AIC
- X1    1     2.118 606.71 89.269
- X4    1    11.487 616.08 89.621
- X5    1    41.174 645.77 90.704
<none>              604.59 91.188
- X3    1    65.097 669.69 91.540
- X6    1    67.621 672.22 91.627
- X2    1    88.904 693.50 92.344
- X7    1   107.366 711.96 92.948

Step:  AIC=89.27
Y ~ X2 + X3 + X4 + X5 + X6 + X7

       Df Sum of Sq    RSS    AIC
- X4    1    20.274 626.99 88.025
<none>              606.71 89.269
- X5    1    60.703 667.42 89.462
- X6    1    76.938 683.65 90.015
- X3    1    83.591 690.30 90.238
- X2    1    87.952 694.66 90.382
+ X1    1     2.118 604.59 91.188
- X7    1   125.104 731.82 91.581

Step:  AIC=88.02
Y ~ X2 + X3 + X5 + X6 + X7

       Df Sum of Sq    RSS    AIC
- X5    1    45.835 672.82 87.648
<none>              626.99 88.025
- X3    1    77.629 704.62 88.710
+ X4    1    20.274 606.71 89.269
+ X1    1    10.906 616.08 89.621
- X2    1   111.808 738.79 89.799
- X7    1   144.368 771.35 90.791
- X6    1   152.121 779.11 91.021
```

Continued Below

```
Step:  AIC=87.65
Y ~ X2 + X3 + X6 + X7

       Df Sum of Sq    RSS    AIC
- X3    1    37.055 709.88 86.881
<none>               672.82 87.648
+ X5    1    45.835 626.99 88.025
- X2    1   104.035 776.86 88.955
- X7    1   107.393 780.22 89.054
- X6    1   109.503 782.33 89.116
+ X4    1     5.406 667.42 89.462
+ X1    1     0.007 672.81 89.647

Step:  AIC=86.88
Y ~ X2 + X6 + X7

       Df Sum of Sq    RSS    AIC
<none>               709.88 86.881
- X2    1    74.976 784.85 87.190
+ X3    1    37.055 672.82 87.648
+ X4    1    19.465 690.41 88.241
+ X5    1     5.261 704.62 88.710
+ X1    1     4.575 705.30 88.732
- X7    1   148.696 858.57 89.255
- X6    1   226.241 936.12 91.244
> summary(stepwise.reg)

Call:
lm(formula = Y ~ X2 + X6 + X7)

Residuals:
   Min     1Q Median     3Q    Max
-8.669 -3.695 -1.177  3.298 11.435

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   49.255      4.154  11.856 3.17e-10 ***
X2            -8.506      6.004  -1.417   0.1728
X6           124.512     50.599   2.461   0.0236 *
X7          -393.145    197.069  -1.995   0.0606 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.112 on 19 degrees of freedom
Multiple R-squared:  0.2819,    Adjusted R-squared:  0.1685
F-statistic: 2.486 on 3 and 19 DF,  p-value: 0.09168
```

Description of Output

- At the first step for the Full Model (<None> Removed), AIC = 91.188. All models, where one variable is removed, are fit. The smallest AIC is when X_1 is removed, with AIC = 89.269 < 91.188
- At the second step for the Full Model (<None> Removed), AIC = 89.269 (the value when X_1 is removed). All models, where one variable is removed are fit, further, a model adding back X_1 is included. The smallest AIC is when X_4 is removed, with AIC = 88.025 < 89.269
- At the third step for the Full Model (<None> Removed), AIC = 88.025 (the value when X_1 and X_4 are removed). All models, where one variable is removed are fit, further, models adding back X_1 and X_4 are included. The smallest AIC is when X_5 is removed, with AIC = 87.648 < 88.025.

- At the fourth step for the Full Model (<None> Removed), AIC = 87.648 (the value when X_1, X_4, and X_5 are removed). All models, where one variable is removed are fit, further models adding back X_1, X_4, and X_5 are included. The smallest AIC is when X_3 is removed, with AIC = 86.881 < 87.648.
- At the fourth step for the Full Model (<None> Removed), AIC = 87.648 (the value when X_1, X_3, X_4, and X_5 are removed). All models, where one variable is removed are fit, further, models adding back X_1, X_3, X_4, and X_5 are included. The smallest AIC is when X_2 is removed, with AIC = 87.190 > 86.881.
- The results for the final regression model , containing X_2 = Propanol, X_6 = Ethyl Caproate, and X_7 = Ethyl Caprylate. Note that level of X_2 = Propanol is negatively related to aroma score, X_6 = Ethyl Caproate is positively related to aroma score, and X_7 = Ethyl Caprylate is negatively related to aroma score.
- The difference between Stepwise Regression and Backward Elimination is that we were using AIC instead of SBC. AIC puts less of a penalty on adding predictors to the model than SBC when $\ln(n) > 2$. Note that no predictors were "brought back" into the regression equation.

Model Validation

- When we have a lot of data, we would like to see how well a model fit on one set of data (training sample) compares to one fit on a new set of data (validation sample), and how the training model fits the new data.
- Want data sets to be similar wrt levels of the predictors
- Training set should have at least 6-10 times as many observations than potential predictors.
- When this is not possible, the *PRESS* statistic is often used, as it is based on "leave-one-out" errors. If *PRESS* and *SSE* are similar, then $s^2 = MSE$ is considered a reasonable estimate of model's quality in prediction.
- Models should give similar model fits based on SSE_p, $PRESS_p$, C_p, and MSE_p and regression coefficients
- Mean Square Prediction Error when training model is applied to validation sample:

$$MSPR = \frac{\sum_{i=1}^{n^*}\left(Y_i - \hat{Y}_i\right)^2}{n^*} \qquad \hat{Y}_i = b_0^T + b_1^T X_{i1}^V + \ldots + b_{p-1}^T X_{i,p-1}^V$$

Example: Predicting Weight from Height, Arm Length, and Hand Length in Potential NFL Athletes

Source: 2014 NFL Combine Data @ nfl.com

Sample: n = 335 NFL prospects at NFL Combine in Indianapolis. Feb. 2014

Response Variable: Y = Weight (lbs)

Predictor Variables: X_1 = Height (inches), X_2 = Arm Length (inches), X_3 = Hand Length (inches)

Procedure:

- Split overall sample into Training Sample and Validation Sample at random. Will put $n_T = 168$ in training sample, and the remaining $n_V = 167$ in validation sample.
- Fit the regression model for the Training sample
- Apply the fitted equation from the Training Sample to the Heights, Arm Lengths, and Hand Lengths in the Validation Sample.
- Obtain the prediction errors from the validation sample, and compute *MSPR*
- Fit model for Validation Sample.
- Compare SSE_p, MSE_p , Regression Coefficients and their standard errors for the two datasets.
- Will construct this in EXCEL, then give R Program for same samples.

For the training sample, we get MSE = 875. For the validation sample, we obtain:

$$MSPR = \frac{\sum_{i=1}^{n^*}\left(Y_i - \hat{Y}_i\right)^2}{n^*} = \frac{171037.6}{167} = 1024.18 \qquad \hat{Y}_i = b_0^T + b_1^T X_{i1}^V + \ldots + b_{p-1}^T X_{i,p-1}^V$$

Thus *MSPR/MSE* = 1.17, and MSPR is about 17% larger than $s^2 = MSE$. This is reasonably close (much closer than the example in the textbook). Still, we might think of the square root of *MSPR* (32.0) as a better measure of out-of-sample prediction error the square root of *MSE* (29.6) based on the training sample. Notice that the ratio of the standard deviations is closer to 1 than the ratio of the variances.

Next, we obtain the two regression models, one for the Training Sample, one for the Validation sample, and compare the results.

SUMMARY OUTPUT						
Training Sample						
Regression Statistics						
Multiple R	0.7277					
R Square	0.5295					
Adjusted R Square	0.5209					
Standard Error	29.5769					
Observations	168					
ANOVA						
	df	*SS*	*MS*	*F*	*nificance F*	
Regression	3	161463	53821	61.52	0.0000	
Residual	164	143466	875			
Total	167	304929				
	b	*SE{b}*	*t Stat*	*P-value*	*LB(95%)*	*UB(95%)*
Intercept	-639.75	65.25	-9.804	0.0000	-768.59	-510.91
Height	5.92	1.26	4.706	0.0000	3.44	8.41
ArmLng	8.06	2.59	3.115	0.0022	2.95	13.17
HandLng	19.19	5.03	3.814	0.0002	9.26	29.13

SUMMARY OUTPUT						
Validation Sample						
Regression Statistics						
Multiple R	0.7645					
R Square	0.5844					
Adjusted R Square	0.5768					
Standard Error	31.6040					
Observations	167					
ANOVA						
	df	*SS*	*MS*	*F*	*nificance F*	
Regression	3	228939	76313	76.40	0.0000	
Residual	163	162806	999			
Total	166	391745				
	b	*SE{b}*	*t Stat*	*P-value*	*LB(95%)*	*UB(95%)*
Intercept	-755.20	66.33	-11.385	0.0000	-886.18	-624.21
Height	9.31	1.35	6.914	0.0000	6.65	11.96
ArmLng	5.61	2.75	2.037	0.0433	0.17	11.05
HandLng	13.58	5.13	2.649	0.0089	3.46	23.71

Note that while the point estimates of the regression coefficients differ for the two samples (as they always will), there is a large amount of overlap for their 95% Confidence Intervals. The Model Validation holds well for these data.

R Program/Output for These Specific Samples (Obtained in EXCEL w/ Random Seed of 1234 for U(0,1))

```
nflcomb <- read.csv("E:\\blue_drive\\sta4210\\nfl_combine.csv",
       header=TRUE)
attach(nflcomb); names(nflcomb)

#### Obtain the Training and Validation Samples, generated in EXCEL
#### Selects all rows where Sample is 1 or 2, and all columns
#### nflcomb[rows,cols]

nfl.train <- nflcomb[Sample==1,]
nfl.valid <- nflcomb[Sample==2,]

#### Fit the Regression model for the Training and Validation Samples

train.mod <- lm(Weight ~ Height + ArmLng + HandLng, data=nfl.train)
valid.mod <- lm(Weight ~ Height + ArmLng + HandLng, data=nfl.valid)
summary(train.mod)
summary(valid.mod)

#### Compute the fitted values for Validation sample from Model fitted
####   to Training Sample (Note that X1, X2, X3 are in Columns 4, 5, 6)
####   and Y is in Column 7
####  Compute MSEP

yhat.val <- predict(train.mod,nfl.valid[,4:6])
e.val <- nfl.valid[,7] - yhat.val
(MSEP <- sum(e.val^2)/nrow(nfl.valid))

############## Output ####################################################

> summary(train.mod)

Call: lm(formula = Weight ~ Height + ArmLng + HandLng, data = nfl.train)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -639.748     65.251  -9.804  < 2e-16 ***
Height         5.924      1.259   4.706 5.33e-06 ***
ArmLng         8.063      2.588   3.115 0.002172 **
HandLng       19.194      5.033   3.814 0.000193 ***

Residual standard error: 29.58 on 164 degrees of freedom
Multiple R-squared:  0.5295,    Adjusted R-squared:  0.5209
F-statistic: 61.52 on 3 and 164 DF,  p-value: < 2.2e-16

> summary(valid.mod)

Call: lm(formula = Weight ~ Height + ArmLng + HandLng, data = nfl.valid)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -755.198     66.335 -11.385  < 2e-16 ***
Height         9.306      1.346   6.914 1.01e-10 ***
ArmLng         5.609      2.754   2.037  0.04332 *
HandLng       13.585      5.129   2.649  0.00888 **

Residual standard error: 31.6 on 163 degrees of freedom
Multiple R-squared:  0.5844,    Adjusted R-squared:  0.5768
F-statistic:  76.4 on 3 and 163 DF,  p-value: < 2.2e-16

> (MSEP <- sum(e.val^2)/nrow(nfl.valid))
[1] 1024.177
```

Note that if we had been fitting "sub-models" with fewer predictors, we would compare their fits for the two samples. In this case, the full model has all predictors being significant. We could have also included interactions and quadratic terms.

R Program for General Case – Using Sample Function to Obtain Samples Internally in R

```
nflcomb <- read.csv("E:\\blue_drive\\sta4210\\nfl_combine.csv",
       header=TRUE)
attach(nflcomb); names(nflcomb)

#### Obtain the Training and Validation Samples
#### nfl.samp obtains a sample of 168 integers from 1:335 w/out replacement
#### Selects all rows where id is either in nfl.samp (train) or not (valid)
#### Selects all columns:     nflcomb[rows,cols]

set.seed(9876)   #### Will produce same sample in future runs
nfl.samp <- sample(1:length(Weight),168,replace=FALSE)
nfl.train <- nflcomb[nfl.samp,]
nfl.valid <- nflcomb[-nfl.samp,]

#### Fit the Regression model for the Training and Validation Samples

train.mod <- lm(Weight ~ Height + ArmLng + HandLng, data=nfl.train)
valid.mod <- lm(Weight ~ Height + ArmLng + HandLng, data=nfl.valid)
summary(train.mod)
summary(valid.mod)

#### Compute the fitted values for Validation sample from Model fitted
####   to Training Sample (Note that X1, X2, X3 are in Columns 4, 5, 6)
####   and Y is in Column 7
####  Compute MSEP

yhat.val <- predict(train.mod,nfl.valid[,4:6])
e.val <- nfl.valid[,7] - yhat.val
(MSEP <- sum(e.val^2)/nrow(nfl.valid))

#### Output
Training Sample:
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -662.395     68.565  -9.661  < 2e-16 ***
Height         8.179      1.449   5.646 7.06e-08 ***
ArmLng         4.274      2.824   1.514 0.132069
HandLng       17.116      5.077   3.371 0.000933 ***

Residual standard error: 31.64 on 164 degrees of freedom
Multiple R-squared:  0.5208,    Adjusted R-squared:  0.512
F-statistic: 59.41 on 3 and 164 DF,  p-value: < 2.2e-16

Validation Sample:
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -737.652     63.668 -11.586  < 2e-16 ***
Height         7.203      1.177   6.120 6.73e-09 ***
ArmLng         9.372      2.539   3.691 0.000304 ***
HandLng       15.161      5.113   2.965 0.003477 **

Residual standard error: 29.69 on 163 degrees of freedom
Multiple R-squared:  0.5957,    Adjusted R-squared:  0.5882
F-statistic: 80.04 on 3 and 163 DF,  p-value: < 2.2e-16

> (MSEP <- sum(e.val^2)/nrow(nfl.valid))
[1] 887.3897
```

Chapter 10 – Regression Model Building - Diagnostics

Model Adequacy for Predictors – Added Variable Plot

- Graphical way to determine partial relation between response and a given predictor, after controlling for other predictors – shows form of relation between new X and Y
- May not be helpful when other predictor(s) enter model with polynomial or interaction terms that are not controlled for
- Algorithm (assume plot for X_3, given X_1, X_2):
  - Fit regression of Y on X_1,X_2, obtain residuals = $e_i(Y|X_1,X_2)$
  - Fit regression of X_3 on X_1,X_2, obtain residuals = $e_i(X_3|X_1,X_2)$
  - Plot $e_i(Y|X_1,X_2)$ (vertical axis) versus $e_i(X_3|X_1,X_2)$ (horizontal axis)
- Slope of the regression through the origin of $e_i(Y|X_1,X_2)$ on $e_i(X_3|X_1,X_2)$ is the partial regression coefficient for X_3

Example: Factors Effecting Air Permeability of Woven Fabrics

$Y \equiv$ Average air permeability $\left(\text{cm}^3/\text{s}/\text{cm}^2\right)$

$X_1 \equiv$ Warp Yarn Density (ends/cm)

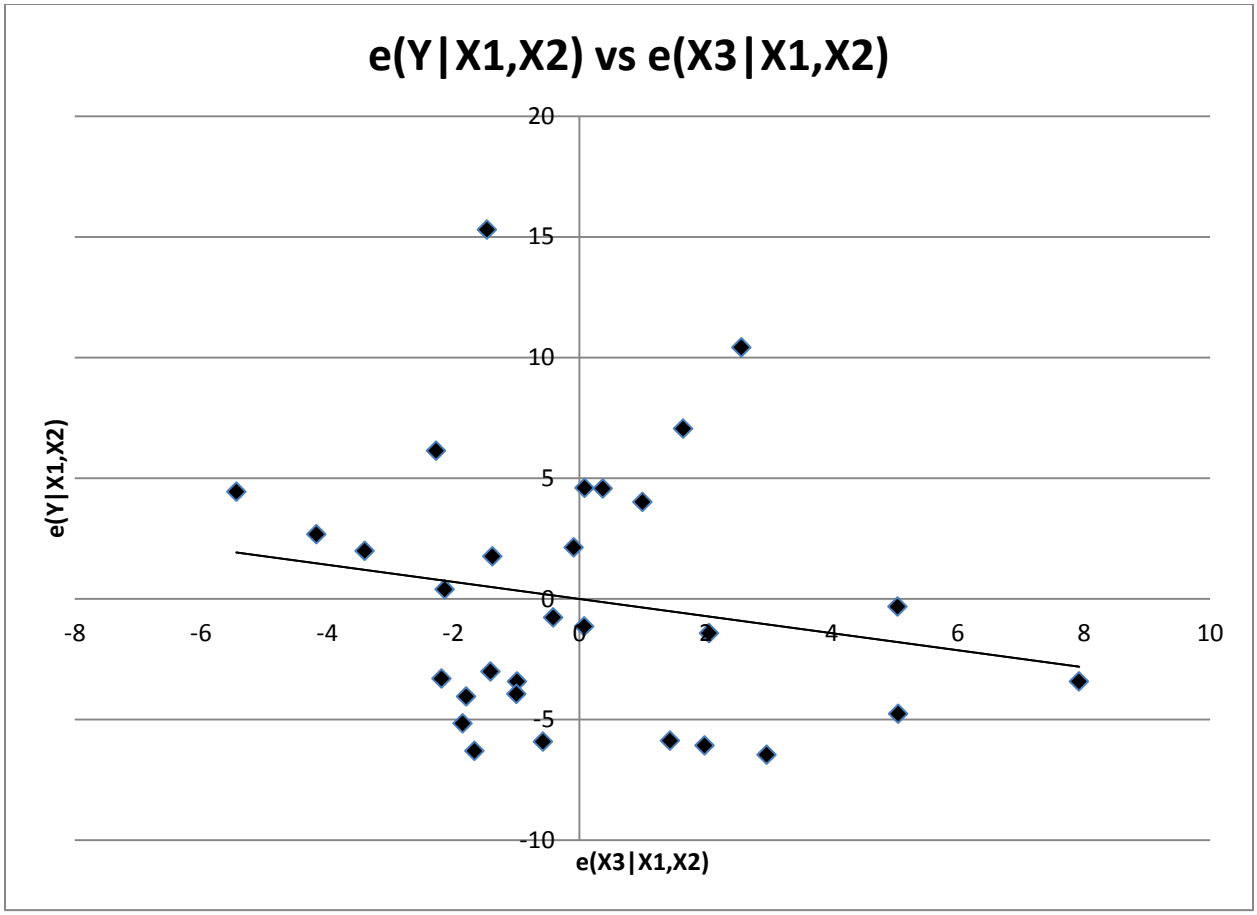
$X_2 \equiv$ Weft Yarn Density (picks/cm)

$X_3 \equiv$ Mass per Unit Area $\left(\text{grams}/\text{cm}^2\right)$

Procedure:

- Regress Y (air permeability) on X_1 and X_2 (warp and weft), obtain residuals $e_i(Y|X_1,X_2)$
- Regress X_3 (mass) on X_1 and X_2 (warp and weft), obtain residuals $e_i(X_3|X_1,X_2)$
- Plot $e_i(Y|X_1,X_2)$ (vertical axis) versus $e_i(X_3|X_1,X_2)$ (horizontal axis)
- Slope of the regression through the origin of $e_i(Y|X_1,X_2)$ on $e_i(X_3|X_1,X_2)$ is the partial regression coefficient for X_3

Slope of the regression (through the origin) of $e_i(Y|X_1,X_2)$ on $e_i(X_3|X_1,X_2)$ is $b_1 = -0.35497$, the same as the partial regression coefficient for Mass in the full model (See Chapter 6).



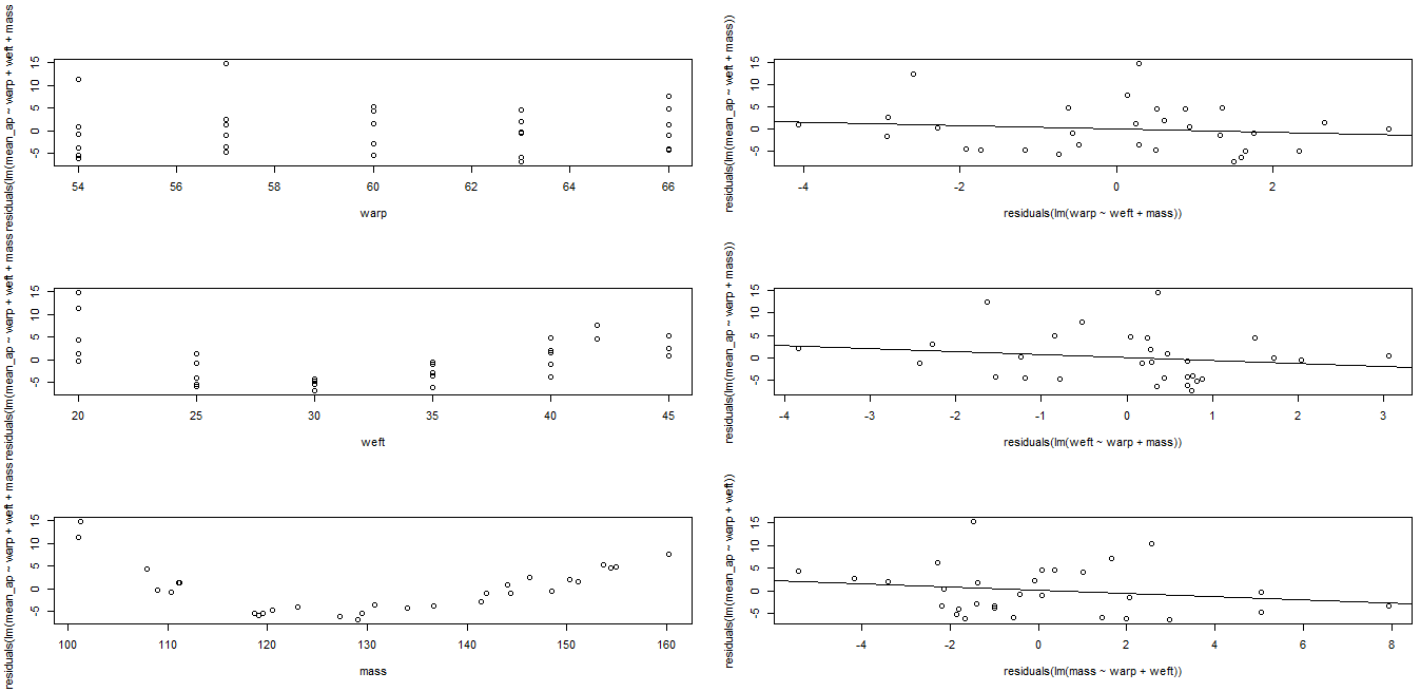
R Program for each plot of Residuals vs X and Added Variable Plots

```
airperm <- read.csv("E:\\blue_drive\\sta4210\\airperm_woven_reg.csv",
       header=TRUE)
attach(airperm); names(airperm)


par(mfrow=c(3,2))
plot(warp,residuals(lm(mean_ap~warp+weft+mass)))
plot(residuals(lm(warp~weft+mass)),residuals(lm(mean_ap~weft+mass)))
abline(lm(residuals(lm(mean_ap~weft+mass)) ~ -1+residuals(lm(warp~weft+mass))))

plot(weft,residuals(lm(mean_ap~warp+weft+mass)))
plot(residuals(lm(weft~warp+mass)),residuals(lm(mean_ap~warp+mass)))
abline(lm(residuals(lm(mean_ap~warp+mass)) ~ -1+residuals(lm(weft~warp+mass))))

plot(mass,residuals(lm(mean_ap~warp+weft+mass)))
plot(residuals(lm(mass~warp+weft)),residuals(lm(mean_ap~warp+weft)))
abline(lm(residuals(lm(mean_ap~warp+weft)) ~ -1+residuals(lm(mass~warp+weft))))
```

Within each of the 3 "groups" of commands, we have (using X_3 = mass for simplicity of notation):

- Plot $e_i(Y|X_1,X_2,X_3)$ (vertical axis) versus X_3 (horizontal axis)
- Plot $e_i(Y|X_1,X_2)$ (vertical axis) versus $e_i(X_3|X_1,X_2)$ (horizontal axis)
- Add regression line of $e_i(Y|X_1,X_2)$ vs $e_i(X_3|X_1,X_2)$ through the origin.

R Graphics Output



The plots of residuals from the full model versus weft and mass (bottom two, left side) clearly show a "U-shaped" pattern, implying a nonlinear relation. The added variable plots (bottom two, right side, show that the shape of the nonlinear pattern is decreasing air permeability as weft and mass increase. We will fit a second-order model in weft and mass, where both Weft and Mass have been centered.

Full Model:

ANOVA				
	df	*SS*	*MS*	*F*
Regression	5	5017.28	1003.46	295
Residual	24	81.70	3.40	
Total	29	5099		
	Coefficients	*andard Err*	*t Stat*	*P-value*
Intercept	6.9657	0.6936	10.0435	0.0000
cweft	0.0081	0.1247	0.0650	0.9487
cmass	-0.6631	0.0573	-11.5823	0.0000
cweft.sq	0.0236	0.0396	0.5956	0.5570
cmass.sq	0.0242	0.0098	2.4712	0.0210
cweft.mass	-0.0215	0.0386	-0.5568	0.5828

The overall model is highly significant

($F^* = 295$, $R^2 = 5017/5099 = 0.9920$)

Mass and Mass^2 have significant t-statistics.

Weft, Weft^2, Mass*Weft do not

Test H_0: $\beta_W = \beta_{W.SQ} = \beta_{W.M} = 0$

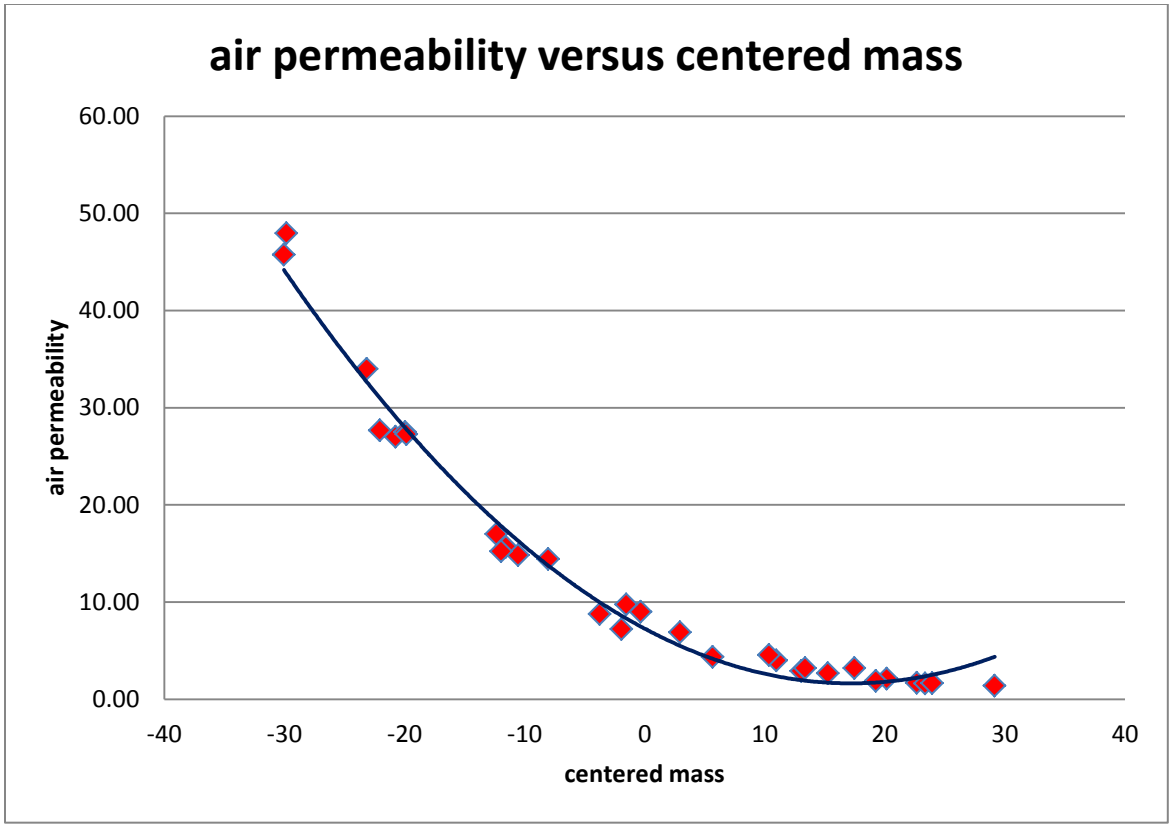
Reduced Model:

ANOVA				
	df	*SS*	*MS*	*F*
Regression	2	5015.58	2507.79	812
Residual	27	83.40	3.09	
Total	29	5099		
	Coefficients	*andard Err*	*t Stat*	*P-value*
Intercept	7.2706	0.4880	14.8984	0.0000
cmass	-0.6553	0.0185	-35.3357	0.0000
cmass.sq	0.0191	0.0012	15.6941	0.0000

$$F^* = \frac{\left[\frac{83.40 - 81.70}{27 - 24}\right]}{\left[\frac{81.70}{24}\right]} = \frac{0.57}{3.40} = 0.17$$

$$F\left(0.95; 3, 24\right) = 3.009$$

$$P - \text{value} \;=\; P\left(F\left(3.24\right) \geq 0.17\right) = 0.9166$$



Note that while the quadratic function gives a reasonable approximation to the air permeability scores, the upward bend at higher levels of centered mass is problematic. A nonlinear regression model with an asymptote would probably be a more reasonable model.

Outlying Y Observations – Studentized Residuals

Model Errors (unobserved):

$$\varepsilon_i = Y_i - \left(\beta_0 + \beta_1 X_{i1} + \ldots + \beta_{p-1} X_{i,p-1}\right) \qquad E\{\varepsilon_i\} = 0 \qquad \sigma^2\{\varepsilon_i\} = \sigma^2 \qquad \sigma\{\varepsilon_i, \varepsilon_j\} = 0 \quad \forall i \neq j$$

Residuals (observed) where $h_{ij} = (i, j)^{th}$ element of $\underset{n\times n}{\mathbf{H}} = \mathbf{X(X'X)^{-1}X'}$:

$$e_i = Y_i - \hat{Y}_i = Y_i - \left(b_0 + b_1 X_{i1} + \ldots + b_{p-1} X_{i,p-1}\right)$$

$$E\{e_i\} = 0 \qquad \sigma^2\{e_i\} = \sigma^2\left(1 - h_{ii}\right) \qquad \sigma\{e_i, e_j\} = -h_{ij}\sigma^2 \quad \forall i \neq j$$

$$s^2\{e_i\} = MSE\left(1 - h_{ii}\right) \qquad s\{e_i, e_j\} = -h_{ij} MSE \quad \forall i \neq j$$

Semi-Studentized Residual (Residual divided by estimate of σ, trivial to compute in spreadsheet):

$$e_i^* = \frac{e_i}{\sqrt{MSE}}$$

Studentized Residual (Residual divided by its standard error, messier to compute in spreadsheet):

$$r_i = \frac{e_i}{\sqrt{MSE\left(1 - h_{ii}\right)}}$$

Example – Wheat Condition in Ohio as Related to Meteorological Factors (1893-1917)

Source: T.A. Blair (1919). "A Statistical Study of Weather Factors Affecting the Yield of Winter Wheat in Ohio," *Monthly Weather Review*, Vol. 47, #12, pp. 841.847.

Response Variable: Y = Wheat condition index (percentage of "normal," can be > 100).

Predictors: X_1 = Temp in Oct/Nov (F), X_2 = Rainfall in Sep (in), X_3 = Rainfall in Oct/Nov (in), X_4 = Sunshine in Oct/Nov (percent).

Note: Y for 1897 was not reported, but X values were. This analysis removes 1897, while original paper used 1897 X values in obtaining means in hand computed least squares estimates, giving slightly different results. The following tables gives the data, fitted values, Semi-Studentized, and Studentized Residuals; as well as matrix computations of **X'X**, $(\mathbf{X'X})^{-1}$, **X'Y**, **b**, and *MSE*. Note that while the **H** matrix is difficult to obtain in EXCEL (you have to highlight a *n*x*n* (24x24, in this case) range of cells, then type in commands), it is not TOO difficult to obtain its diagonal elements h_{ii} if you include the intercept among the columns of *X* variables:

$$h_{ii} = \mathbf{x_i}'\left(\mathbf{X'X}\right)^{-1}\mathbf{x_i}$$

$$\Rightarrow \quad h_{11} = \begin{bmatrix}1 & 46 & 1.6 & 6.3 & 44\end{bmatrix}\begin{bmatrix} 2.6227 & -0.3369 & -0.0696 & -0.3020 & -0.0736 \\ -0.3369 & 0.0086 & -0.0017 & -0.0009 & -0.0006 \\ -0.0696 & -0.0017 & 0.0322 & -0.0015 & 0.0014 \\ -0.3020 & -0.0009 & -0.0015 & 0.0300 & 0.0041 \\ -0.0736 & -0.0006 & 0.0014 & 0.0041 & 0.0016 \end{bmatrix}\begin{bmatrix}1\\46\\1.6\\6.3\\44\end{bmatrix} = 0.1498$$

year	whtcnd.y	x0	tempon.x1	rains.x2	rainon.x3	sunon.x4	Y-hat_i	e_i	h_ii	e_i*	r_i
1893	92	1	46	1.6	6.3	44	88.3637	3.6363	0.1498	0.4297	0.4660
1894	93	1	46	3.3	4.2	54	84.7485	8.2515	0.1149	0.9750	1.0364
1895	74	1	44	1.7	5.3	52	82.0147	-8.0147	0.2021	-0.9470	-1.0602
1896	101	1	47	5.1	3.8	46	90.7864	10.2136	0.2776	1.2068	1.4199
1898	102	1	46	2.6	6.9	40	92.9093	9.0907	0.1487	1.0742	1.1642
1899	83	1	50	2.7	3.8	47	88.7796	-5.7796	0.1717	-0.6829	-0.7504
1900	86	1	51	1.8	6.1	48	93.1572	-7.1572	0.2433	-0.8457	-0.9722
1901	75	1	46	2.9	2.2	52	79.6347	-4.6347	0.2196	-0.5476	-0.6199
1902	98	1	52	4.6	4.9	42	99.0161	-1.0161	0.3880	-0.1201	-0.1535
1903	80	1	46	1.5	4.7	51	82.7120	-2.7120	0.1037	-0.3205	-0.3385
1904	76	1	46	2	1.9	46	78.3875	-2.3875	0.4020	-0.2821	-0.3648
1905	98	1	46	2.9	6.2	46	90.4897	7.5103	0.0876	0.8874	0.9290
1906	97	1	47	2.9	5.8	39	92.3631	4.6369	0.1370	0.5479	0.5898
1907	84	1	44	3.9	4.7	52	85.3999	-1.3999	0.2256	-0.1654	-0.1880
1908	62	1	48	0.6	2.3	60	75.2653	-13.2653	0.2850	-1.5674	-1.8537
1909	95	1	49	1.8	4.8	54	86.3572	8.6428	0.1231	1.0212	1.0906
1910	91	1	46	4	6.1	44	93.1284	-2.1284	0.1159	-0.2515	-0.2675
1911	83	1	46	4.9	7.9	38	100.7598	-17.7598	0.3086	-2.0985	-2.5238
1912	95	1	48	3.1	3.5	58	84.0236	10.9764	0.1489	1.2970	1.4059
1913	99	1	49	2.4	6.9	41	95.7024	3.2976	0.1825	0.3896	0.4309
1914	94	1	50	1.4	4.9	51	87.5877	6.4123	0.1727	0.7577	0.8330
1915	85	1	50	4.5	5.1	56	93.6363	-8.6363	0.2998	-1.0205	-1.2196
1916	87	1	48	2.6	4.2	61	83.8714	3.1286	0.2290	0.3697	0.4210
1917	83	1	43	1.9	5.3	41	83.9056	-0.9056	0.2629	-0.1070	-0.1246

X'X						X'Y
24	1134	66.7	117.8	1163		2113
1134	53702	3155.5	5562.7	55003		99967
66.7	3155.5	219.05	335.75	3184.2		5981.5
117.8	5562.7	335.75	631.4	5565.6		10545
1163	55003	3184.2	5565.6	57391		101765
INV(X'X)						b
22.62274	-0.36693	-0.06961	-0.30203	-0.07362		27.30306
-0.36693	0.008594	-0.00167	-0.00087	-0.00062		1.155569
-0.06961	-0.00167	0.03221	-0.00149	0.001372		2.181775
-0.30203	-0.00087	-0.00149	0.030044	0.004123		2.357811
-0.07362	-0.00062	0.001372	0.004123	0.001631		-0.23729
SSE	dfE	MSE				
1360.838	19	71.62305				

Note that as the studentized residuals are like t-statistics, we are interested in any extreme ones outside the range of say -3 to +3 (will be more specific below). Only one year (1911, with r_i = -2.5238 is outside the range of -2 to +2. There is no evidence of extreme outliers in this example.

Outlying Y Observations – Studentized Deleted Residuals

Deleted Residual (Observed value minus fitted value when regression is fit on the other n-1 cases):

$d_i = Y_i - \hat{Y}_{i(i)} \qquad \hat{Y}_{i(i)} = b_{0(i)} + b_{1(i)}X_{i1} + ... + b_{p-1(i)}X_{i,p-1}$

$b_{k(i)} \equiv$ regression coefficient of X_k when case i is deleted

Studentized Deleted Residual (makes use of having predicted i^{th} response from regression based on other n-1 cases):

$$t_i = \frac{d_i}{s\{d_i\}} \sim t(n-p-1)$$

$$s^2\{d_i\} = s^2\{\text{pred}_i\} = MSE_{(i)}\left[1 + \mathbf{X_i}'\left(\mathbf{X_{(i)}}'\mathbf{X_{(i)}}\right)^{-1}\mathbf{X_i}\right] \qquad \mathbf{X_i}' = \begin{bmatrix} 1 & X_{i1} & \cdots & X_{i,p-1} \end{bmatrix}$$

Note: $$SSE = (n-p)MSE = (n-p-1)MSE_{(i)} + \frac{e_i^2}{1-h_{ii}}$$

$$\Rightarrow \quad t_i = e_i\left[\frac{n-p-1}{SSE(1-h_{ii}) - e_i^2}\right]^{1/2}$$ Computed without re-fitting n regressions

Test for outliers (Bonferroni adjustment): Outlier if $|t_i| \geq t\left(1-\left(\frac{\alpha}{2n}\right), n-p-1\right)$

Note: for the Ohio Wheat data: n = 24, p=5, $t\left(1-\left(\frac{05}{2n}\right), n-p-1\right) = t\left(.9990, 18\right) = 3.6105$

year	e_i	h_ii	t_i
1893	3.6363	0.1498	0.4562
1894	8.2515	0.1149	1.0385
1895	-8.0147	0.2021	-1.0638
1896	10.2136	0.2776	1.4618
1898	9.0907	0.1487	1.1759
1899	-5.7796	0.1717	-0.7414
1900	-7.1572	0.2433	-0.9707
1901	-4.6347	0.2196	-0.6096
1902	-1.0161	0.3880	-0.1495
1903	-2.7120	0.1037	-0.3305
1904	-2.3875	0.4020	-0.3563
1905	7.5103	0.0876	0.9255
1906	4.6369	0.1370	0.5794
1907	-1.3999	0.2256	-0.1831
1908	-13.2653	0.2850	-1.9936
1909	8.6428	0.1231	1.0964
1910	-2.1284	0.1159	-0.2608
1911	-17.7598	0.3086	-3.0128
1912	10.9764	0.1489	1.4457
1913	3.2976	0.1825	0.4215
1914	6.4123	0.1727	0.8260
1915	-8.6363	0.2998	-1.2364
1916	3.1286	0.2290	0.4117
1917	-0.9056	0.2629	-0.1214

The largest studentized deleted residual is for 1911 (case 18), with t_i = -3.0128 < 3.6105. So, it is not too extreme, given the sample size.

Outlying *X*-Cases – Hat Matrix Leverage Values

$$\mathbf{H} = \mathbf{X}\left(\mathbf{X'X}\right)^{-1}\mathbf{X'} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n1} & h_{n2} & \cdots & h_{nn} \end{bmatrix} \qquad h_{ij} = \mathbf{x_i'}\left(\mathbf{X'X}\right)^{-1}\mathbf{x_j} \qquad \mathbf{x_i} = \begin{bmatrix} 1 \\ x_{i1} \\ \vdots \\ x_{i,p-1} \end{bmatrix}$$

Notes: $0 \le h_{ii} \le 1$ $\qquad \sum_{i=1}^{n} h_{ii} = trace\left(\mathbf{H}\right) = trace\left(\mathbf{X}\left(\mathbf{X'X}\right)^{-1}\mathbf{X'}\right) = trace\left(\mathbf{X'X}\left(\mathbf{X'X}\right)^{-1}\right) = trace\left(\mathbf{I_p}\right) = p$

Cases with *X*-levels close to the "center" of the sampled *X*-levels will have small leverages.

Cases with "extreme" levels have large leverages, and have the **potential** to "pull" the regression equation toward their observed *Y*-values. Large leverage values are considered to be > 2*p*/*n* (2 times larger than the mean of the leverage values). Note that leverage values cannot exceed 1.

Example – Wheat Condition in Ohio as Related to Meteorological Factors (1893-1917)

There are p-1 = 4 predictor variables, so that p = 5, and n = 24 observations, so that 2p/n = 10/24 = 0.4167. No years exceed 0.4167 (the closest is 1904 with h_{ii} = 0.4020). Thus no years had particularly extreme weather conditions that may cause it to have a large impact on least squares estimates.

$$\hat{\mathbf{Y}} = \mathbf{HY} \quad \Rightarrow \quad \hat{Y}_i = \sum_{j=1}^{n} h_{ij} Y_j = \sum_{j=1}^{i-1} h_{ij} Y_j + h_{ii} Y_{ii} + \sum_{j=i+1}^{n} h_{ij} Y_j$$

with: $\sum_{j=1}^{n} h_{ij} = 1$

Leverage values for new observations: $h_{\text{new,new}} = \mathbf{X}'_{\text{new}} \left(\mathbf{X'X}\right)^{-1} \mathbf{X}_{\text{new}}$

New cases with leverage values larger than those in original dataset are extrapolations, and may not have reasonable predicted values as the functional form of the regression is only observed within the observed *X* levels in the sample.

Identifying Influential Cases I – Fitted Values

Influential Cases in Terms of Their Own Fitted Values - DFFITS:

$DFFITS_i = \dfrac{\hat{Y}_i - \hat{Y}_{i(i)}}{\sqrt{MSE_{(i)} h_{ii}}} \equiv$ # of standard errors own fitted value is shifted when case included vs excluded

Computational Formula (avoids fitting all deleted models):

$$DFFITS_i = e_i \left[\frac{n-p-1}{SSE\left(1-h_{ii}\right) - e_i^2}\right]^{1/2} \left(\frac{h_{ii}}{1-h_{ii}}\right)^{1/2} = t_i \left(\frac{h_{ii}}{1-h_{ii}}\right)^{1/2}$$

Problem cases are (in absolute value) >1 for small to medium sized datasets, $> 2\sqrt{\frac{p}{n}}$ for larger ones

Influential Cases in Terms of All Fitted Values - Cook's Distance:

$$D_i = \frac{\sum_{j=1}^{n} \left(\hat{Y}_j - \hat{Y}_{j(i)}\right)^2}{pMSE} = \frac{\left(\hat{\mathbf{Y}} - \hat{\mathbf{Y}}_{(i)}\right)' \left(\hat{\mathbf{Y}} - \hat{\mathbf{Y}}_{(i)}\right)}{pMSE} = \frac{e_i^2}{pMSE}\left[\frac{h_{ii}}{\left(1-h_{ii}\right)^2}\right]$$

Problem cases are $> F\left(0.50; p, n-p\right)$

Example – Wheat Condition in Ohio as Related to Meteorological Factors (1893-1917)

$$n = 24 \quad p = 5 \quad SSE = 1360.838 \quad 2\sqrt{\frac{p}{n}} = 2\sqrt{\frac{5}{24}} = 0.913 \quad F\left(0.50;5,19\right) = 0.895$$

year	e_i	h_ii	DFFITS_i	D_i
1893	3.6363	0.1498	0.1915	0.0076
1894	8.2515	0.1149	0.3742	0.0279
1895	-8.0147	0.2021	-0.5354	0.0569
1896	10.2136	0.2776	0.9062	0.1550
1898	9.0907	0.1487	0.4914	0.0473
1899	-5.7796	0.1717	-0.3375	0.0233
1900	-7.1572	0.2433	-0.5504	0.0608
1901	-4.6347	0.2196	-0.3233	0.0216
1902	-1.0161	0.3880	-0.1190	0.0030
1903	-2.7120	0.1037	-0.1124	0.0027
1904	-2.3875	0.4020	-0.2922	0.0179
1905	7.5103	0.0876	0.2867	0.0166
1906	4.6369	0.1370	0.2309	0.0110
1907	-1.3999	0.2256	-0.0988	0.0021
1908	-13.2653	0.2850	-1.2588	0.2740
1909	8.6428	0.1231	0.4108	0.0334
1910	-2.1284	0.1159	-0.0944	0.0019
1911	-17.7598	0.3086	-2.0128	0.5686
1912	10.9764	0.1489	0.6048	0.0692
1913	3.2976	0.1825	0.1991	0.0083
1914	6.4123	0.1727	0.3773	0.0290
1915	-8.6363	0.2998	-0.8091	0.1274
1916	3.1286	0.2290	0.2244	0.0105
1917	-0.9056	0.2629	-0.0725	0.0011

Years 1908 and 1911 appear to be influential with respect to their own fitted values:
($|DFFITS_i| > 0.913$)

None appear to be influential with respect to the overall vector of fitted values:
(all $D_i < 0.895$). The highest is 1911 (0.5686)

Influential Cases II – Regression Coefficients

Influential Cases in Terms of Regression Coefficients (One for each case for each coefficient):

$$\left(DFBETAS\right)_{k(i)} = \frac{b_k - b_{k(i)}}{\sqrt{MSE_{(i)}c_{kk}}} \equiv \text{ \# of standard errors coefficient is shifted when case included vs excluded}$$

where $\left(\mathbf{X'X}\right)^{-1} = \begin{bmatrix} c_{00} & c_{01} & \cdots & c_{0,p-1} \\ c_{10} & c_{11} & \cdots & c_{1,p-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{p-1,0} & c_{p-1,1} & \cdots & c_{p-1,p-1} \end{bmatrix}$

Problem cases are >1 for small to medium sized datasets, $> \frac{2}{\sqrt{n}}$ for larger ones

Influential Cases in Terms of Vector of Regression Coefficients - Cook's Distance:

$$D_i = \frac{\sum_{j=1}^{n}\left(\hat{Y}_j - \hat{Y}_{j(i)}\right)^2}{pMSE} = \frac{\left(\hat{\mathbf{Y}} - \hat{\mathbf{Y}}_{(\mathbf{i})}\right)'\left(\hat{\mathbf{Y}} - \hat{\mathbf{Y}}_{(\mathbf{i})}\right)}{pMSE} = \frac{e_i^2}{pMSE}\left[\frac{h_{ii}}{\left(1-h_{ii}\right)^2}\right] = \frac{\left(\mathbf{b} - \mathbf{b}_{(i)}\right)'\left(\mathbf{X}'\mathbf{X}\right)\left(\mathbf{b} - \mathbf{b}_{(i)}\right)}{pMSE}$$

Problem cases are $> F\left(0.50; p, n-p\right)$

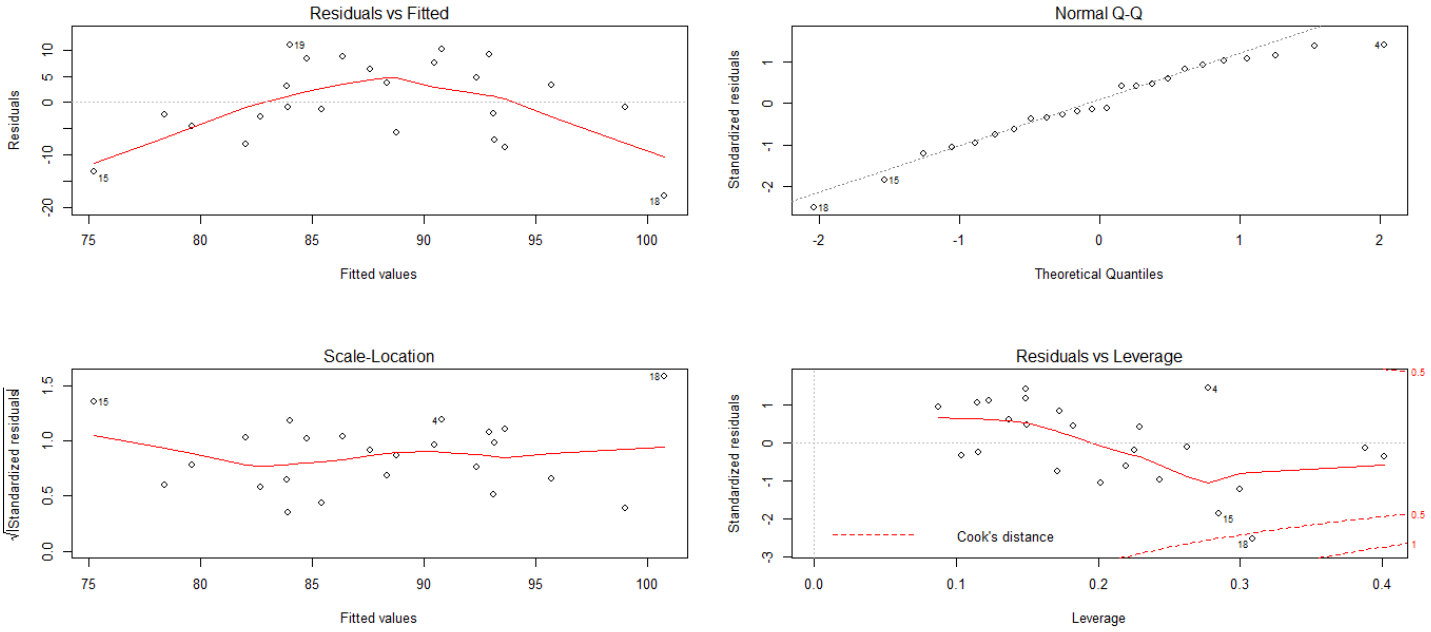
When some cases are highly influential, should check and see if they affect inferences regarding model.

We will obtain DFBETAS from R Output Below. For the Ohio Wheat data $2/\sqrt{n} = 0.408$.

```
wheat <- read.csv("E:\\blue_drive\\sta4210\\ohio_wheat.csv",
       header=TRUE)
attach(wheat); names(wheat)

wheat.mod1 <- lm(whtcnd.y ~ tempon.x1 + rains.x2 + rainon.x3 + sunon.x4)
summary(wheat.mod1)
hatvalues(wheat.mod1)   #### Also given in influence.measures below
rstudent(wheat.mod1)   ### Studentized Deleted Residuals, not printed to save space
influence.measures(wheat.mod1)
par(mfrow=c(2,2))
plot(wheat.mod1)
############### Output ######################
> summary(wheat.mod1)
Call: lm(formula = whtcnd.y ~ tempon.x1 + rains.x2 + rainon.x3 + sunon.x4)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  27.3031    40.2531   0.678    0.506
tempon.x1     1.1556     0.7846   1.473    0.157
rains.x2      2.1818     1.5189   1.436    0.167
rainon.x3     2.3578     1.4669   1.607    0.124
sunon.x4     -0.2373     0.3418  -0.694    0.496

Residual standard error: 8.463 on 19 degrees of freedom
Multiple R-squared:  0.4096,    Adjusted R-squared:  0.2853
F-statistic: 3.295 on 4 and 19 DF,  p-value: 0.03276

> influence.measures(wheat.mod1)
Influence measures of
         lm(formula = whtcnd.y ~ tempon.x1 + rains.x2 + rainon.x3 + sunon.x4) :

    dfb.1_ dfb.tm.1 dfb.rn.2 dfb.rn.3 dfb.sn.4   dffit cov.r  cook.d    hat inf
1   0.0510 -0.03841 -0.12149   0.0750 -0.02906  0.1915 1.455 0.00765 0.1498
2   0.0627 -0.17213  0.16930   0.0120  0.20808  0.3742 1.107 0.02789 0.1149
3  -0.2329  0.36838  0.16618  -0.2116 -0.23410 -0.5354 1.211 0.05693 0.2021
4   0.1763 -0.06560  0.70386  -0.4631 -0.22313  0.9062 1.035 0.15497 0.2776
5   0.1431 -0.09485 -0.12956   0.1935 -0.15938  0.4914 1.063 0.04735 0.1487
6   0.0890 -0.22531  0.03407   0.1954  0.17695 -0.3375 1.361 0.02334 0.1717
7   0.3736 -0.39863  0.25006  -0.2067 -0.01335 -0.5504 1.342 0.06077 0.2433
8  -0.1522  0.08037 -0.05718   0.2622  0.07594 -0.3233 1.516 0.02163 0.2196
9   0.0542 -0.08617 -0.04455   0.0372  0.05220 -0.1190 2.128 0.00299 0.3880   *
10 -0.0341  0.03767  0.06868  -0.0145 -0.01998 -0.1124 1.418 0.00265 0.1037
11 -0.1593  0.02628  0.05624   0.2613  0.19058 -0.2922 2.116 0.01790 0.4020   *
12  0.0576 -0.11011  0.00373   0.1653  0.05425  0.2867 1.138 0.01657 0.0876
13  0.0724  0.01866 -0.03472  -0.0438 -0.17651  0.2309 1.384 0.01105 0.1370
14 -0.0419  0.07146 -0.05416  -0.0114 -0.04371 -0.0988 1.676 0.00206 0.2256
15  0.0713 -0.13131  0.67961   0.3834 -0.26934 -1.2588 0.676 0.27401 0.2850
16 -0.2234  0.16818 -0.17421   0.1319  0.17850  0.4108 1.082 0.03340 0.1231
17 -0.0224  0.03304 -0.05182  -0.0267 -0.00066 -0.0944 1.454 0.00187 0.1159
18 -0.1669  0.40538 -1.04189  -0.9343  0.09278 -2.0128 0.246 0.56859 0.3086   *
19 -0.1756  0.01998  0.21186  -0.0371  0.37751  0.6048 0.889 0.06918 0.1489
20 -0.0614  0.09351 -0.07361   0.0756 -0.06424  0.1991 1.526 0.00829 0.1825
21 -0.2016  0.23867 -0.23036   0.0518  0.01133  0.3773 1.315 0.02896 0.1727
22  0.5282 -0.25315 -0.50129  -0.2720 -0.50260 -0.8091 1.245 0.12739 0.2998
23 -0.0917 -0.00234  0.02935   0.0813  0.19532  0.2244 1.622 0.01053 0.2290
24 -0.0622  0.04688  0.02522   0.0114  0.03187 -0.0725 1.771 0.00111 0.2629
```

R identifies cases 9, 11, and 18 as influential cases in aggregate, these are years: 1902, 1904, and 1911. In particular case 18 has a large impact on b_1, b_2, and b_3. Remedial measures are described in the next chapter.

Residual plots from the Regression analysis:



- The Residuals vs Fitted values plot identifies the two large negative residuals at the low and high extremes of fitted values (cases 15 and 18, respectively).
- The Normal Q-Q plot does not show any extreme departures from normality.
- The Scale-Location plot shows fairly constant variance (with the cases 15 and 18 again being away from the scatter.
- The Residuals versus Leverage plot identifies Case 18 as having a fairly large leverage (but not the highest) and a Cook's D value between 0.5 and 1.

Multicollinearity - Variance Inflation Factors

- Problems when predictor variables are correlated among themselves (apply to partial coefficients)
  - Regression Coefficients of predictors change, depending on what other predictors are included
  - Extra Sums of Squares of predictors change, depending on what other predictors are included
  - Standard Errors of Regression Coefficients increase when predictors are highly correlated
  - Individual Regression Coefficients are not significant, although the overall model is
  - Width of Confidence Intervals for Regression Coefficients increases when predictors are highly correlated
  - Point Estimates of Regression Coefficients can be wrong sign (+/-)

Variance Inflation Factor (VIF)

Original Units for $X_1,...,X_{p-1},Y$: $\boldsymbol{\sigma^2}\{\mathbf{b}\} = \sigma^2(\mathbf{X'X})^{-1}$

Correlation Transformed Values: $X_{ik}^* = \frac{1}{\sqrt{n-1}}\left(\frac{X_{ik}-\overline{X}_k}{s_k}\right)$ $\quad Y_i^* = \frac{1}{\sqrt{n-1}}\left(\frac{Y_i-\overline{Y}}{s_Y}\right)$

$\boldsymbol{\sigma^2}\{\mathbf{b}^*\} = (\sigma^*)^2\mathbf{r}_{xx}^{-1}$ $\quad \sigma^2\{b_k^*\} = (\sigma^*)^2(VIF)_k$

where: $(VIF)_k = \frac{1}{1-R_k^2}$

with $R_k^2 \equiv$ Coefficient of Determination for regression of X_k on the other $p-2$ predictors

$R_k^2 = 0 \Rightarrow (VIF)_k = 1$ $\quad 0 < R_k^2 < 1 \Rightarrow (VIF)_k > 1$ $\quad R_k^2 = 1 \Rightarrow (VIF)_k = \infty$

Multicollinearity is considered problematic wrt least squares estimates if:

$\max\left((VIF)_1,...,(VIF)_{p-1}\right) > 10$ or if $\left(\overline{VIF}\right) = \frac{\sum_{k=1}^{p-1}(VIF)_k}{p-1}$ is much larger than 1

Making use of the DAAG package in R:

```
wheat <- read.csv("E:\\blue_drive\\sta4210\\ohio_wheat.csv",
       header=TRUE)
attach(wheat); names(wheat)

install.packages("DAAG")
library(DAAG)                 # VIF option included in DAAG package
wheat.mod1 <- lm(whtcnd.y ~ tempon.x1 + rains.x2 + rainon.x3 + sunon.x4)
(bf.vif <- vif(wheat.mod1))      # Obtain VIF1,VIF2,VIF3,VIF4

######### Output

> (bf.vif <- vif(wheat.mod1))      # Obtain VIF1,VIF2,VIF3,VIF4
tempon.x1  rains.x2 rainon.x3  sunon.x4
   1.0356    1.0848    1.5983    1.6866
```

There is definitely no evidence of Multicollinearity among the predictors.

Chapter 11 – Model Building: Remedial Measures

Unequal (Independent) Error Variances – Weighted Least Squares (WLS)

- Case 1 – Error Variances known exactly (VERY rare)
- Case 2 – Error Variances known up to a constant
  - Occasionally information known regarding experimental units regarding the relative magnitude (unusual)
  - If "observations" are means of different numbers of units (each with equal variance) at the various X levels, Variance of observation i is σ^2/n_i where n_i is known
- Case 3 – Estimated Variances
  - Data are individual points and squared residuals are treated as estimated variances
  - Variance (or Standard Deviation) is related to one or more predictors, and relation can be modeled (see Breusch-Pagan Test in Chapter 3)
  - Mean and variance of grouped observations are used as data and estimated weights
- Case 4 – Ordinary Least Squares with estimated variances

WLS – Case 1 - Known Variances - I

$$Y_i = \beta_0 + \beta_1 X_{i1} + ... + \beta_{p-1} X_{i,p-1} + \varepsilon_i \qquad \varepsilon_i \sim N\left(0, \sigma_i^2\right) \quad i = 1, ..., n \qquad \sigma\left\{\varepsilon_i, \varepsilon_j\right\} = 0 \quad \forall\, i \neq j$$

$$\Rightarrow \boldsymbol{\sigma}^2\{\boldsymbol{\varepsilon}\} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{bmatrix}$$

Maximum Likelihood Estimation:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left[-\frac{1}{2\sigma_i^2}\left(Y_i - \beta_0 - \beta_1 X_{i1} - ... - \beta_{p-1} X_{i,p-1}\right)^2 \right] \quad \text{setting: } w_i = \frac{1}{\sigma_i^2}$$

$$\Rightarrow \quad L(\boldsymbol{\beta}) = \left[\prod_{i=1}^{n} \sqrt{\frac{w_i}{2\pi}} \right] \exp\left[-\frac{1}{2} \sum_{i=1}^{n} w_i \left(Y_i - \beta_0 - \beta_1 X_{i1} - ... - \beta_{p-1} X_{i,p-1}\right)^2 \right]$$

To maximize $L(\boldsymbol{\beta})$, we need to minimize $Q_w = \sum_{i=1}^{n} w_i \left(Y_i - \beta_0 - \beta_1 X_{i1} - ... - \beta_{p-1} X_{i,p-1}\right)^2$

Note that values with smaller σ_i^2 have larger weights w_i in the weighted least squares criterion.

WLS – Case 1 - Known Variances – II

Easiest to set up in matrix form, where: $\mathbf{W} = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{bmatrix} = \mathbf{W}'$

$\boldsymbol{\sigma}^2\{\mathbf{Y}\} = \boldsymbol{\sigma}^2\{\boldsymbol{\varepsilon}\} = \mathbf{W}^{-1}$

Normal Equations: $(\mathbf{X'WX})\mathbf{b}_w = \mathbf{X'WY}$

$\Rightarrow \quad \mathbf{b}_w = (\mathbf{X'WX})^{-1}\mathbf{X'WY} = \mathbf{AY} \quad \mathbf{A} = (\mathbf{X'WX})^{-1}\mathbf{X'W}$

$\Rightarrow \quad \mathbf{E}\{\mathbf{b_w}\} = \mathbf{AE}\{\mathbf{Y}\} = \mathbf{AX}\boldsymbol{\beta} = (\mathbf{X'WX})^{-1}\mathbf{X'WX}\boldsymbol{\beta} = \boldsymbol{\beta}$

$\Rightarrow \quad \boldsymbol{\sigma}^2\{\mathbf{b_w}\} = \mathbf{A}\boldsymbol{\sigma}^2\{\mathbf{Y}\}\mathbf{A}' = (\mathbf{X'WX})^{-1}\mathbf{X'WW}^{-1}\mathbf{WX}(\mathbf{X'WX})^{-1} = (\mathbf{X'WX})^{-1}$

WLS – Case 2 – Variance Known up to Constant - I

When data are means of unequal numbers of replicates at X-levels, can use any weights generally

$\sigma^2\{\overline{Y}_i\} = \sigma_i^2 = \dfrac{\sigma^2}{r_i} \quad \Rightarrow \quad \boldsymbol{\sigma}^2\{\mathbf{Y}\} = \sigma^2 \begin{bmatrix} r_1^{-1} & 0 & \cdots & 0 \\ 0 & r_2^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_n^{-1} \end{bmatrix}$ $\quad r_i \equiv$ number of replicates at i^{th} level of X

$w_i = \dfrac{1}{\sigma_i^2} = \dfrac{n_i}{\sigma^2} \quad \Rightarrow \quad \mathbf{W} = \dfrac{1}{\sigma^2}\begin{bmatrix} r_1 & 0 & \cdots & 0 \\ 0 & r_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_n \end{bmatrix} = \dfrac{1}{\sigma^2}\mathbf{W}^* \qquad \mathbf{W}^* = \begin{bmatrix} r_1 & 0 & \cdots & 0 \\ 0 & r_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_n \end{bmatrix}$

$\Rightarrow \mathbf{X'WX} = \mathbf{X'}\dfrac{1}{\sigma^2}\mathbf{W}^*\mathbf{X} = \dfrac{1}{\sigma^2}\mathbf{X'W}^*\mathbf{X} \Rightarrow (\mathbf{X'WX})^{-1} = \sigma^2(\mathbf{X'W}^*\mathbf{X})^{-1}$

$\Rightarrow \mathbf{X'WY} = \mathbf{X'}\dfrac{1}{\sigma^2}\mathbf{W}^*\mathbf{Y} = \dfrac{1}{\sigma^2}\mathbf{X'W}^*\mathbf{Y} \quad \Rightarrow \mathbf{b}_w = (\mathbf{X'WX})^{-1}\mathbf{X'WY} = (\mathbf{X'W}^*\mathbf{X})^{-1}\mathbf{X'W}^*\mathbf{Y}$

$\boldsymbol{\sigma}^2\{\mathbf{b}_w\} = (\mathbf{X'WX})^{-1} = \sigma^2(\mathbf{X'W}^*\mathbf{X})^{-1} \qquad \mathbf{s}^2\{\mathbf{b}_w\} = MSE_w(\mathbf{X'W}^*\mathbf{X})^{-1} \qquad MSE_w = \dfrac{\sum_{i=1}^{n} w_i^*\left(Y_i - \hat{Y}_i\right)^2}{(n-p)}$

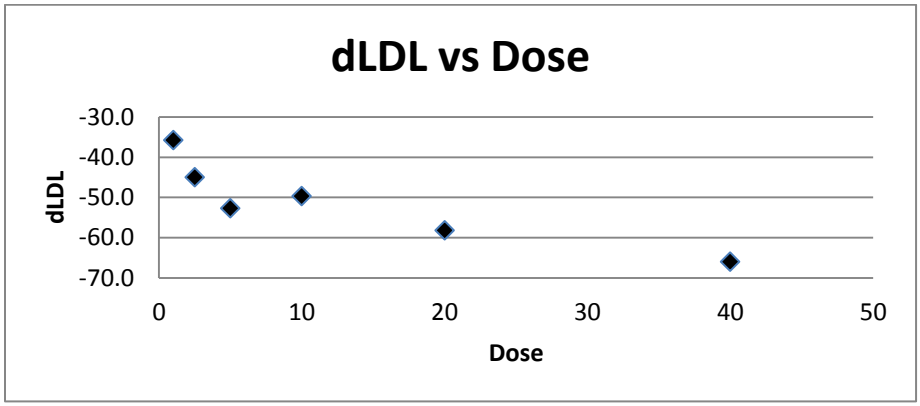
WLS – Case 2 – Variance Known up to Constant – II

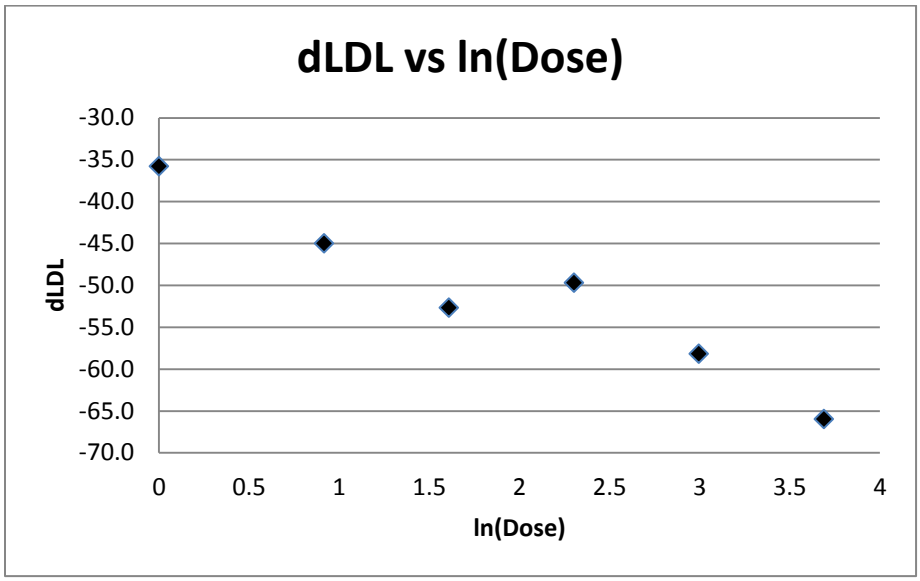
Example: Rosuvastin for Patients with High Cholesterol

Source: "Randomized Dose-Response Study of Rosuvastin in Japanese Patients with Hypercholesterolemia," Saito, et al, *Journal of Atherosclerosis*, 2003, 10:329-336

Japanese study of Rosuvastin for patients with high cholesterol. n = 6 doses, # of patients per dose (r_i) varied: (15, 17, 12, 14, 18, 13)

After preliminary plots, determined that Y = change in LDL, X = ln(Dose)





DoseGrp	ChgLDL	Dose	ln(Dose)	r
1	-35.8	1	0.0000	15
2	-45.0	2.5	0.9163	17
3	-52.7	5	1.6094	12
4	-49.7	10	2.3026	14
5	-58.2	20	2.9957	18
6	-66.0	40	3.6889	13

X		Y	W*					
1	0.0000	-35.8	15	0	0	0	0	0
1	0.9163	-45.0	0	17	0	0	0	0
1	1.6094	-52.7	0	0	12	0	0	0
1	2.3026	-49.7	0	0	0	14	0	0
1	2.9957	-58.2	0	0	0	0	18	0
1	3.6889	-66.0	0	0	0	0	0	13

X'W*X			X'W*Y		Y	Yhat_W	e_w	w*(e_w)^2
89	169.005		-4535.8		-35.8	-36.96	1.2	20.14
169.005	458.0243		-9624.3		-45.0	-43.72	-1.3	27.99
					-52.7	-48.83	-3.9	179.82
NV(X'W*X)			b_w		-49.7	-53.94	4.2	251.82
0.037538	-0.01385		-36.96		-58.2	-59.05	0.9	13.11
-0.01385	0.007294		-7.38		-66.0	-64.17	-1.8	43.75
s2{b_w}			s{b_w}				sum(w*e^2	536.63
5.04	-1.86		2.24				n-p	4
-1.86	0.98		0.99				MSE_W	134.16

R Program – Uses replicate sizes as weights

```
dLDL <- c(-35.8,-45.0,-52.7,-49.7,-58.2,-66.0)
DOSE <- c(1,2.5,5,10,20,40)
r <- c(15,17,12,14,18,13)
lnDOSE <- log(DOSE)

rosuv.wls <- lm(dLDL ~ lnDOSE, weight=r)
summary(rosuv.wls)

####### Output ############

Call:
lm(formula = dLDL ~ lnDOSE, weights = r)

Weighted Residuals:
      1       2       3       4       5       6
  4.488  -5.291 -13.410  15.869   3.620  -6.615

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -36.9588     2.2441 -16.469 7.96e-05 ***
lnDOSE       -7.3753     0.9892  -7.456  0.00173 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.58 on 4 degrees of freedom
Multiple R-squared:  0.9329,    Adjusted R-squared:  0.9161
F-statistic: 55.59 on 1 and 4 DF,  p-value: 0.001729
```

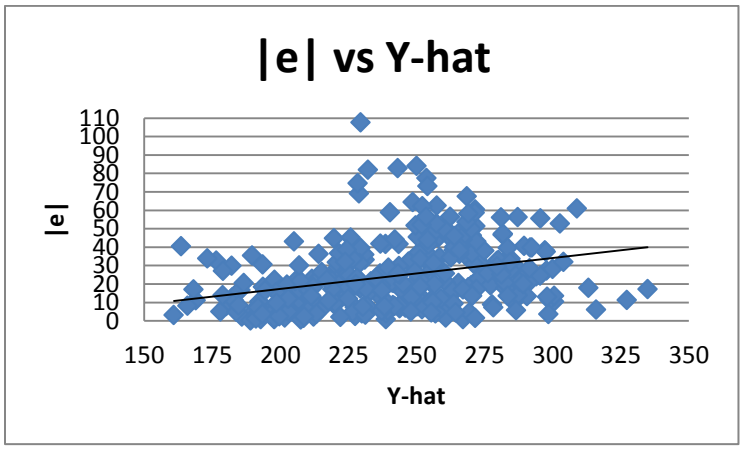
WLS - Case 3 – Estimated Variances (as a Function of the Mean)

- Use squared residuals (estimated variances) or absolute residuals (estimated standard deviations) from OLS to model their levels as functions of predictor variables
  - Plot Residuals, squared residuals and absolute residuals versus fitted values and predictors
  - Using model building techniques used on Y previously to obtain a model for either the variances or standard deviations as functions of the X^s or the mean
  - Fit estimated WLS with the estimated weights (1/variances)
  - Iterate until regression coefficients are stable (iteratively re-weighted least squares)

$$\hat{w}_i = \frac{1}{\hat{v}_i} = \frac{1}{\left(\hat{s}_i\right)^2} \qquad \mathbf{b}_{\hat{w}} = \left(\mathbf{X}'\hat{\mathbf{W}}\mathbf{X}\right)^{-1}\mathbf{X}'\hat{\mathbf{W}}\mathbf{Y} \qquad MSE_{\hat{w}} = \frac{\left(\mathbf{Y}-\mathbf{X}\mathbf{b}_{\hat{w}}\right)'\hat{\mathbf{W}}\left(\mathbf{Y}-\mathbf{X}\mathbf{b}_{\hat{w}}\right)}{n-p}$$

Example: Predicting Weight from Height, Arm Length, and Hand Length in Potential NFL Athletes

Plot of absolute residuals versus fitted values (the plot of squared residuals is not nearly as linear):



Fitted Regression, relating standard deviation to mean (r^2 = .099):

	Coefficients	*tandard Err*	*t Stat*	*P-value*
Intercept	-16.3451	6.8100	-2.4002	0.0169
y-hat	0.1682	0.0278	6.0383	0.0000

OLS estimates: $\hat{s}_i = -16.3541 + 0.1682\hat{Y}_i$ $\qquad \hat{w}_i = \left(\frac{1}{\hat{s}_i}\right)^2$

First Iteration estimates:

X'W1X					X'W1Y
0.688126	49.76622	21.78464	6.456559		154.6008
49.76622	3604.597	1577.543	467.413		11240.14
21.78464	1577.543	691.0041	204.6783		4921.186
6.456559	467.413	204.6783	60.80124		1458.946
INV(X'W1X)					b_W1
1032.767	-10.1305	-5.04973	-14.7926		-634.286
-10.1305	0.432368	-0.6326	-0.11852		7.62315
-5.04973	-0.6326	1.925756	-1.08337		5.198495
-14.7926	-0.11852	-1.08337	6.145407		15.24755

Fitted Regression, relating standard deviation to mean (r^2 = .108):

	Coefficients	*tandard Err*	*t Stat*	*P-value*
Intercept	-20.9842	7.2413	-2.8979	0.0040
Yhat1	0.1884	0.0297	6.3433	0.0000

Second Iteration estimates:

X'W2X					X'W2Y
0.689508	49.81464	21.81689	6.463718		154.3944
49.81464	3604.385	1578.247	467.4489		11213.56
21.81689	1578.247	691.6607	204.7974		4912
6.463718	467.4489	204.7974	60.81508		1455.704
INV(X'W2X)					b_W2
1028.955	-10.0753	-5.13663	-14.6215		-631.051
-10.0753	0.43224	-0.63347	-0.11828		7.594223
-5.13663	-0.63347	1.92795	-1.07742		5.184381
-14.6215	-0.11828	-1.07742	6.107899		15.17688

Fitted Regression, relating standard deviation to mean (r^2 = .108):

	Coefficients	*tandard Err*	*t Stat*	*P-value*
Intercept	-21.1961	7.2696	-2.9157	0.0038
Yhat2	0.1893	0.0298	6.3484	0.0000

Third iteration estimates:

X'W3X					X'W3Y
0.689515	49.81383	21.81648	6.463631		154.3818
49.81383	3604.236	1578.178	467.4308		11212.36
21.81648	1578.178	691.6285	204.7889		4911.46
6.463631	467.4308	204.7889	60.81294		1455.552
INV(X'W3X)					b_W3
1028.887	-10.0759	-5.13435	-14.6203		-630.938
-10.0759	0.43227	-0.63355	-0.11816		7.593839
-5.13435	-0.63355	1.928114	-1.07756		5.182475
-14.6203	-0.11816	-1.07756	6.107288		15.17435

The estimates have stabilized fairly well. Extra iterations are trivial in programming loops in R.

	b_W3	s{b_W3}	t	P-value
Intercept	-630.9382	39.3332	-16.0409	0.0000
Height	7.5938	0.8062	9.4191	0.0000
ArmLng	5.1825	1.7027	3.0437	0.0025
HandLng	15.1743	3.0304	5.0074	0.0000

R Program – Matrix Form, Using lm Function for Regressions of |e| on Y-hat for Weights

```
nflcomb <- read.csv("E:\\blue_drive\\sta4210\\nfl_combine.csv, header=TRUE)
attach(nflcomb); names(nflcomb)

#### Matrix form (using lm for |e|,y-hat regressions) ###########

n <- length(Weight)
X0 <- rep(1,n)
X <- as.matrix(cbind(X0,Height,ArmLng,HandLng))
Y <- as.matrix(Weight)
p <- ncol(X)

#### Fit original regression, and regress |e| on Y-hat

b.ols <- solve(t(X) %*% X) %*% t(X) %*% Y
yhat.ols <- X %*% b.ols
e.ols <- Y - yhat.ols
abs.e.ols <- abs(e.ols)
e.reg.ols <- lm(abs.e.ols ~ yhat.ols)
summary(e.reg.ols)
s.ols <- predict(e.reg.ols)
w.ols <- 1/s.ols^2

b.old <- b.ols
wm.old <- as.matrix(diag(w.ols))
b.diff <- 100

while (b.diff > 0.0001) {
b.new <- solve(t(X) %*% wm.old %*% X) %*% t(X) %*% wm.old %*% Y
yhat.new <- X %*% b.new
abs.e.new <- abs(Y - yhat.new)
wm.new <- as.matrix(diag(1/predict(lm(abs.e.new~yhat.new))^2))
b.diff <- sum((b.new-b.old)^2)
b.old <- b.new
wm.old <- wm.new
}
```

Continued Below

```
b.wls <- b.new
wm.wls <- wm.new
mse.w <- (t(Y-X%*%b.wls) %*% wm.wls %*% (Y-X%*%b.wls))/(n-p)
s2.b.wls <- mse.w[1,1]*solve(t(X) %*% wm.wls %*% X)
s.b.wls <- sqrt(diag(s2.b.wls))
t.b.wls <- b.wls/s.b.wls
print(round(cbind(b.wls,s.b.wls,t.b.wls),3))

############ Output ###############

> e.reg.ols <- lm(abs.e.ols ~ yhat.ols)
> summary(e.reg.ols)

Call:
lm(formula = abs.e.ols ~ yhat.ols)

Residuals:
    Min      1Q  Median      3Q     Max
-30.747 -13.151  -3.238  10.027  85.317

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -16.34510    6.81000  -2.400   0.0169 *
yhat.ols      0.16815    0.02785   6.038 4.16e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 17.34 on 333 degrees of freedom
Multiple R-squared:  0.09869,   Adjusted R-squared:  0.09598
F-statistic: 36.46 on 1 and 333 DF,  p-value: 4.159e-09

> print(round(cbind(b.wls,s.b.wls,t.b.wls),3))
                  s.b.wls
X0      -630.935  39.333 -16.041
Height     7.594   0.806   9.419
ArmLng     5.182   1.703   3.044
HandLng   15.174   3.030   5.007
```

Notes Regarding the program:

- The loop continues iterating the estimation process until the sum of the squared differences between b_{new} and b_{old} is sufficiently small: $(b_{new} - b_{old})' (b_{new} - b_{old}) < 0.001$
- Within each pass of the loop, the weight matrix is obtained with the following command:
    - `wm.new <- as.matrix(diag(1/predict(lm(abs.e.new~yhat.new))^2))`
    - First: Regress the new values of $|e|$ on Y-hat, based on the new b_W
    - Second: Obtain the predicted value of for each observation based on the regression
    - Third: Create a diagonal matrix of weights, where weight is the reciprocal of the squared predicted value: $w = 1/s^2$

Data are Replicate Measurements at Each *X* Level

At each unique level of *X* levels, there are *r* measurements (although they don't have to all be the same). Here, we can compute the sample variance at each of *X* levels. The inverse of the sample variances can be used as the weights, with more precise levels (smaller variances) having higher weights. The model is fit, just as described above, with the estimated weight matrix based on the estimated (inverse) variances. Note that the main difference between this method and the previous case is that we don't estimate variance function (though we could), and use the replications at each distinct X level to estimate the variance.

Example – Spread of Shotgun Pellets by Distance

Source: W.F. Rowe and S.R. Hanson (1985). "Range-of-Fire Estimates from Regression Analysis Applied to the Spreads of Shotgun Pellet Patterns: Results of a Blind Study," *Forensic Science International*, Vol. 28, pp. 239-250.

Experiment: Shotgun fired 6 times at each of 5 distances (10-50ft by 10ft).

Response: *Y*= sqrt(Area of smallest rectangle that would enclose the pellet pattern).

Predictor: *X* = Range of fire (Distance, in feet).

Data/Model: $\sqrt{A_i} = \beta_0 + \beta_1 X_i + \varepsilon_i = \mathbf{X\beta} + \boldsymbol{\varepsilon} \quad \sigma^2\{\boldsymbol{\varepsilon}\} = \begin{bmatrix} \sigma_1^2\mathbf{I}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 \\ \mathbf{0}_6 & \sigma_2^2\mathbf{I}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 \\ \mathbf{0}_6 & \mathbf{0}_6 & \sigma_3^2\mathbf{I}_6 & \mathbf{0}_6 & \mathbf{0}_6 \\ \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \sigma_4^2\mathbf{I}_6 & \mathbf{0}_6 \\ \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \sigma_5^2\mathbf{I}_6 \end{bmatrix}$

Range	Y1	Y2	Y3	Y4	Y5	Y6	Mean	SD
10	2.60	3.35	3.33	3.06	3.38	3.85	3.262	0.413
20	6.84	6.32	6.96	5.85	5.95	6.29	6.368	0.453
30	6.51	6.72	8.24	7.38	9.84	9.42	8.018	1.393
40	10.28	11.47	14.10	12.54	16.13	11.03	12.592	2.183
50	11.80	13.74	15.18	20.13	16.94	14.09	15.313	2.904

Note that the standard deviation increases with the range. The weight matrix, W is the inverse of the estimated Variance-Covariance matrix:

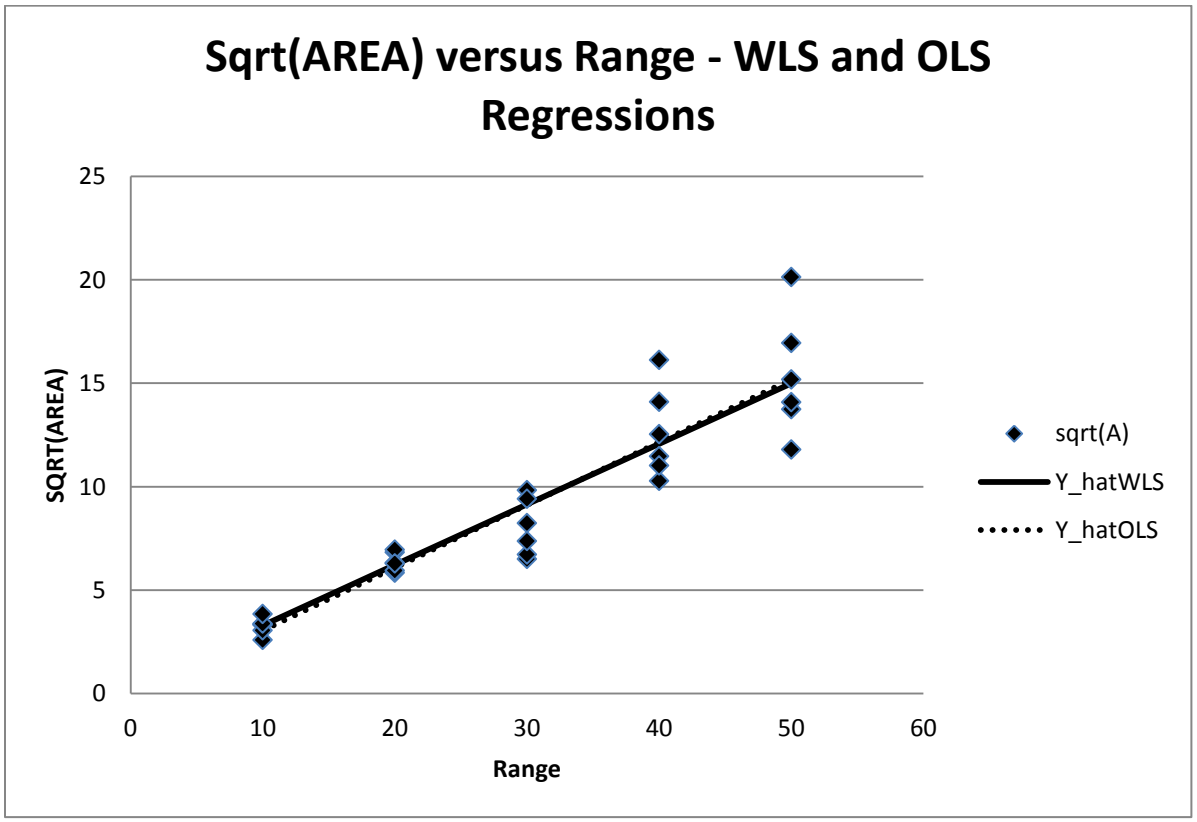
$$\mathbf{W} = \begin{bmatrix} \left(\frac{1}{0.413^2}\right)\mathbf{I}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 \\ \mathbf{0}_6 & \left(\frac{1}{0.453^2}\right)\mathbf{I}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 \\ \mathbf{0}_6 & \mathbf{0}_6 & \left(\frac{1}{1.393^2}\right)\mathbf{I}_6 & \mathbf{0}_6 & \mathbf{0}_6 \\ \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \left(\frac{1}{2.183^2}\right)\mathbf{I}_6 & \mathbf{0}_6 \\ \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \left(\frac{1}{2.904^2}\right)\mathbf{I}_6 \end{bmatrix} = \begin{bmatrix} 5.8627\mathbf{I}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 \\ \mathbf{0}_6 & 4.8731\mathbf{I}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 \\ \mathbf{0}_6 & \mathbf{0}_6 & 0.5153\mathbf{I}_6 & \mathbf{0}_6 & \mathbf{0}_6 \\ \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 & 0.2098\mathbf{I}_6 & \mathbf{0}_6 \\ \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 & 0.1186\mathbf{I}_6 \end{bmatrix}$$

X'WX			X'WY	
69.58888	1116.71902		352.9546	
1116.719	21810.9021		6801.031	
INV(X'WX)			Beta_W	SE{B_W}
0.080562	-0.0041248		0.381986	0.293888
-0.00412	0.00025704		0.29226	0.0166
SSE_WLS	MSE_WLS			
30.01878	1.07209938			

Based on OLS, we get:

b_0 = 0.012667

b_1 = 0.303267



R Program (Analyzes only Cartridge Type = 2, Study had 2 Brands of Cartridge)

```
sg_spread <-
read.table("http://www.stat.ufl.edu/~winner/data/shotgun_spread.dat",header=F,
                col.names=c("cartridge","range","sqrtA","SD_range"))
attach(sg_spread)
regweight <- 1/(SD_range^2)

######## Note: there were 2 types of Cartridge, Analyze on Cartridge Type
### Ordinary Least Squares
sg.mod1 <- lm(sqrtA[cartridge==2] ~ range[cartridge==2])
summary(sg.mod1)
anova(sg.mod1)

#### Weighted Least Squares
sg.mod2 <- lm(sqrtA[cartridge==2] ~ range[cartridge==2],
  weight=regweight[cartridge==2])
summary(sg.mod2)
anova(sg.mod2)
```

R Output

```
> summary(sg.mod1)
Call: lm(formula = sqrtA[cartridge == 2] ~ range[cartridge == 2])

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           0.01267    0.75413   0.017    0.987
range[cartridge == 2]  0.30327    0.02274  13.337 1.18e-13 ***

Residual standard error: 1.761 on 28 degrees of freedom
Multiple R-squared:  0.864,     Adjusted R-squared:  0.8591
F-statistic: 177.9 on 1 and 28 DF,  p-value: 1.184e-13

> anova(sg.mod1)
Analysis of Variance Table

Response: sqrtA[cartridge == 2]
                     Df Sum Sq Mean Sq F value    Pr(>F)
range[cartridge == 2]  1 551.82  551.82  177.89 1.184e-13 ***
Residuals             28  86.86    3.10
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>

> summary(sg.mod2)

Call:lm(formula = sqrtA[cartridge == 2] ~ range[cartridge == 2], weights = regweight[cartridge ==
2])

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)            0.3821     0.2940   1.299    0.204
range[cartridge == 2]   0.2923     0.0166  17.604   <2e-16 ***

Residual standard error: 1.035 on 28 degrees of freedom
Multiple R-squared:  0.9171,    Adjusted R-squared:  0.9142
F-statistic: 309.9 on 1 and 28 DF,  p-value: < 2.2e-16

> anova(sg.mod2)
Analysis of Variance Table

Response: sqrtA[cartridge == 2]
                     Df Sum Sq Mean Sq F value    Pr(>F)
range[cartridge == 2]  1 332.06  332.06  309.89 < 2.2e-16 ***
Residuals             28  30.00    1.07
```

Case 4 – OLS with Estimated Variances

$$\mathbf{E}\{\mathbf{Y}\} = \mathbf{X\beta} \qquad \boldsymbol{\sigma}^2\{\mathbf{Y}\} = \boldsymbol{\sigma}^2\{\boldsymbol{\varepsilon}\} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{bmatrix}$$

$$\mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'Y} = \mathbf{AY} \qquad \mathbf{A} = (\mathbf{X'X})^{-1}\mathbf{X'}$$

$$\Rightarrow \quad \boldsymbol{\sigma}^2\{\mathbf{b}\} = \mathbf{A}\boldsymbol{\sigma}^2\{\mathbf{Y}\}\mathbf{A'} = (\mathbf{X'X})^{-1}\mathbf{X'}\boldsymbol{\sigma}^2\{\boldsymbol{\varepsilon}\}\mathbf{X}(\mathbf{X'X})^{-1}$$

Note: $E\{\varepsilon_i^2\} = \sigma_i^2$ Use e_i^2 as an estimator of σ_i^2: $\quad \mathbf{S_0} = \begin{bmatrix} e_1^2 & 0 & \cdots & 0 \\ 0 & e_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e_n^2 \end{bmatrix}$

$$\mathbf{s}^2\{\mathbf{b}\} = (\mathbf{X'X})^{-1}\mathbf{X'S_0X}(\mathbf{X'X})^{-1}$$

This is referred to White's estimator, and is robust to specification of form of error variance.

Example: Factors Affecting Recycling Yield in Scotland Local Authorities

Source: J. Baird, R. Curry, and T. Reid (2013). "Development and Application of a Multiple Linear Regression Model to Consider the Impact of Weekly Waste Container Capacity on the Yield from Kerbside Recycling Programs in Scotland," *Waste Management & Research*, Vol. 31, pp. 306-314

Response Variable: Y = Average weekly Recycling yield (kg/household/week)

Predictor Variables: X_1 = Recycling Capacity per household per week (kg), X_2 = Residual (Waste) Capacity per household per week (kg), X_3 = number of materials that are collected in recycling containers.

Data: Fitted Values, Residuals, and Squared Residuals:

LocAuth	int.x0	reccap.x1	rescap.x2	nummat.x3	yield	y-hat	e	e^2
Aberdeen City	1	62.5	240	5	2.11	2.050296	0.059704	0.003565
Angus	1	55	120	4	1.87	2.569433	-0.69943	0.489206
Argyll and Bute	1	120	120	5	2.8	3.235181	-0.43518	0.189382
Clackmannanshire	1	147.5	120	8	4.17	3.99955	0.17045	0.029053
Dumfries and Galloway	1	40	240	1	0.88	1.135409	-0.25541	0.065234
Dundee City	1	87.5	240	6	2.35	2.421626	-0.07163	0.00513
East Ayrshire	1	147.5	120	4	3.68	3.250274	0.429726	0.184665
East Dunbartonshire	1	110	240	5	2.65	2.399917	0.250083	0.062542
East Lothian	1	50	240	5	2.81	1.95829	0.85171	0.72541
East Renfrewshire	1	75	240	6	2.36	2.32962	0.03038	0.000923
Edinburgh, City of	1	92.5	240	6	2.04	2.458428	-0.41843	0.175082
Eileen Siar	1	120	120	5	2.97	3.235181	-0.26518	0.070321
Falkirk	1	147.5	120	9	3.89	4.186869	-0.29687	0.088131
Fife	1	90	90	3	3.02	2.830144	0.189856	0.036045
Glasgow City	1	60	240	3	1.79	1.657256	0.132744	0.017621
Highland	1	27.5	240	2	1.63	1.230723	0.399277	0.159422
Inverclyde	1	120	120	4	3.44	3.047862	0.392138	0.153773
Midlothian	1	88	120	5	3.85	2.999646	0.850354	0.723101
Moray	1	80	120	6	2.66	3.128082	-0.46808	0.219101
North Ayrshire	1	133.75	120	7	4.16	3.711025	0.448975	0.201579
North Lanarkshire	1	155	120	8	4.37	4.054753	0.315247	0.09938
Orkney	1	27.5	240	3	1.35	1.418042	-0.06804	0.00463
Perth and Kinross	1	120	120	4	2.75	3.047862	-0.29786	0.088721
Renfrewshire	1	145	120	6	3.1	3.606511	-0.50651	0.256553
Scottish Borders	1	70	180	7	2.88	2.860967	0.019033	0.000362
Shetland Islands	1	74	240	4	1.24	1.947622	-0.70762	0.500728
South Ayrshire	1	87.5	120	6	3.58	3.183285	0.396715	0.157383
South Lanarkshire	1	155	120	7	3.84	3.867434	-0.02743	0.000753
Stirling	1	55	120	8	3.27	3.318709	-0.04871	0.002373
West Dunbartonshire	1	82.5	240	6	2.22	2.384824	-0.16482	0.027167
West Lothian	1	120	120	5	3.03	3.235181	-0.20518	0.042099

Standard Regression output (*n* = 31 localities):

ANOVA						
	df	*SS*	*MS*	*F*	*gnificance F*	
Regression	3	20.3313	6.7771	38.2853	0.0000	
Residual	27	4.7794	0.1770			
Total	30	25.1107				
	Coefficients	*andard Err*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	2.17699	0.49679	4.3821	0.0002	1.15766	3.19633
reccap.x1	0.00736	0.00286	2.5732	0.0159	0.00149	0.01323
rescap.x2	-0.00635	0.00165	-3.8477	0.0007	-0.00973	-0.00296
nummat.x3	0.18732	0.04962	3.7753	0.0008	0.08551	0.28912

Note the following:

- R^2 = 20.3313/25.1107 = 0.8097, over 80% of variation in yield is explained by the model
- b_1 and b_3 are positive: As recycling capacity and number of materials increase, yield increases
- b_2 is negative: As residual (waste) capacity increases, recycling yield decreases

Robust Estimator for $s^2\{b\}$, s{b} and t-tests:

X'X					X'Y			
31	2945.75	5190	163		86.76			
2945.75	324421.5625	449670	16644.75		9064.51			
5190	449670	976500	26130		13305			
163	16644.75	26130	959		491.15			
INV(X'X)					b_OLS	Robust s{b}	t	
1.394238	-0.00501381	-0.00401918	-0.04044		2.17699	0.49880	4.364	
-0.00501	4.62204E-05	1.48638E-05	-0.00036		0.00736	0.00259	2.837	
-0.00402	1.48638E-05	1.53723E-05	6.3E-06		-0.00635	0.00163	-3.905	
-0.04044	-0.00035502	6.30121E-06	0.013907		0.18732	0.03375	5.549	
X'S_0X					Robust s2{b}			
4.779434	422.3177259	782.1668389	23.61595		0.248797	-0.00094619	-0.00073721	-0.004
422.3177	43143.90285	63489.43795	2211.961		-0.00095	6.73157E-06	2.6043E-06	-3.2E-05
782.1668	63489.43795	144093.2616	3750.159		-0.00074	2.6043E-06	2.64181E-06	4.36E-06
23.61595	2211.96091	3750.158623	125.042		-0.004	-3.1998E-05	4.36398E-06	0.001139

Note that the standard errors for b_1 and b_2 do not change appreciably, but for b_3, it is quite a bit smaller for the robust estimator (0.03375 vs 0.04962).

R Program and Output

```
recycle <- read.csv("E:\\blue_drive\\sta4210\\scottish_recycle.csv",
       header=TRUE)
attach(recycle); names(recycle)

recycle.ols <- lm(yield ~ reccap.x1 + rescap.x2 + nummat.x3)
summary(recycle.ols)

####### Matrix approach for White's robust standard errors

n <- length(yield)
X <- as.matrix(cbind(int.x0,reccap.x1,rescap.x2,nummat.x3))
Y <- as.matrix(yield)

b.ols <- solve(t(X) %*% X) %*% t(X) %*% Y
e <- Y - (X %*% b.ols)
e2 <- e %*% t(e)
i.n <- diag(n)
S.0 <- e2 * i.n
s2.b.rob <- solve(t(X) %*% X) %*% (t(X) %*% S.0 %*% X) %*% solve(t(X) %*% X)
s.b.rob <- sqrt(diag(s2.b.rob))
t.b.rob <- b.ols/s.b.rob

print(round(cbind(b.ols,s.b.rob,t.b.rob),5))

####### Output ######

> summary(recycle.ols)
Call: lm(formula = yield ~ reccap.x1 + rescap.x2 + nummat.x3)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.176991   0.496792   4.382 0.000160 ***
reccap.x1    0.007360   0.002860   2.573 0.015887 *
rescap.x2   -0.006347   0.001650  -3.848 0.000661 ***
nummat.x3    0.187319   0.049617   3.775 0.000799 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4207 on 27 degrees of freedom
Multiple R-squared:  0.8097,    Adjusted R-squared:  0.7885
F-statistic: 38.29 on 3 and 27 DF,  p-value: 7.253e-10

> print(round(cbind(b.ols,s.b.rob,t.b.rob),5))
                    s.b.rob
int.x0     2.17699 0.49880  4.36450
reccap.x1  0.00736 0.00259  2.83691
rescap.x2 -0.00635 0.00163 -3.90507
nummat.x3  0.18732 0.03375  5.54944
```

Note that once the residual vector is obtained, we compute **e2 = e %*% t(e)** . This creates an *n*x*n* matrix with squared residuals on the main diagonal, and cross-products elsewhere. When we element-wise multiply **e2** by a nxn identity matrix, we get $\mathbf{S_0}$: **S.0 ← e2 *i.n**

Multicollinearity – Remedial Measures - I

- Goal: Prediction – Multicollinearity not an issue if new cases have similar multicollinearity among predictors
  - Firm uses shipment size (number of pallets) and weight (tons) to predict unloading times. Future shipments have similar correlation between predictors, predictions and PI^s are valid
- Linear Combinations of Predictors can be generated that are uncorrelated (Principal Components)
  - Good: Multicollinearity gone
  - Bad: New variables may not have "physical" interpretation
- Use of Cross-Section and Time Series in Economics
  - Model: Demand = f(Income , Price)
  - Step 1: Cross-section (1 time): Regress Demand on Income (b_I)
  - Step 2: Time Series: Regress Residuals from 1 on Price (b_P)

Multicollinearity – Ridge Regression –I

- Mean Square Error of an Estimator = Variance + $Bias^2$
- Goal: Add Bias to estimator to reduce MSE(Estimator)
- Makes use of Standardized Regression Model with Correlation Transformation

$$MSE\left\{b^R\right\} = E\left\{\left(b^R-\beta\right)^2\right\} = \sigma^2\left\{b^R\right\} + \left(E\left\{b^R\right\}-\beta\right)^2 = \text{Variance} + (\text{Bias})^2$$

Correlation Transformation / Standardized Regression Model:

$$Y_i^* = \frac{Y_i - \overline{Y}}{\left(\sqrt{n-1}\right)s_Y} \qquad X_{ik}^* = \frac{X_{ik} - \overline{X}_k}{\left(\sqrt{n-1}\right)s_k} \qquad Y_i^* = \beta_1^* X_{i1}^* + \ldots + \beta_{p-1}^* X_{i,p-1}^* + \varepsilon_i^* \quad \Rightarrow \quad \mathbf{r}_{XX}\mathbf{b} = \mathbf{r}_{YX}$$

Ridge Regression Estimator (with $c \geq 0$): $\left(\mathbf{r}_{XX} + c\mathbf{I}\right)\mathbf{b}^R = \mathbf{r}_{YX} \quad \Rightarrow \quad \mathbf{b}^R = \left(\mathbf{r}_{XX} + c\mathbf{I}\right)^{-1}\mathbf{r}_{YX} = \begin{bmatrix} b_1^R \\ \vdots \\ b_{p-1}^R \end{bmatrix}$

Goal: choose small *c* such that the estimators stabilize (flatten) and VIF^s get small. Common way is to plot the "Ridge Trace", the regression coefficients versus *c* to determine where the estimates stabilize (flatten out). A second plot is to plot the Variance Inflation Factors (VIF) versus *c*, and determine where they all go below 10. Various other methods are also used, based on PRESS statistic and Generalized Cross Validation (see Case Studies web page example (**China Carbon Emissions and Population Factors** for more detail).

Computational Formulas:

Variance Inflation Factors:

Diagonal Elements of the matrix: $\left(\mathbf{r}_{XX}+c\mathbf{I}\right)^{-1}\mathbf{r}_{XX}\left(\mathbf{r}_{XX}+c\mathbf{I}\right)^{-1}$

Error Sum of Squares (Correlation Transformed Data):

$$\hat{Y}_i^* = b_1^R X_{i1}^* + \ldots + b_{p-1}^R X_{i,p-1}^* \qquad SSE_R = \sum_{i=1}^{n}\left(Y_i^* - \hat{Y}_i^*\right)^2$$

$$R_R^2 = 1 - SSE_R \qquad \text{since } SSTO_R = 1$$

Example: China Carbon Emissions and Population Factors (1978-2008)

Source: Q. Zhu and X. Peng (2012). "The Impacts of Population Change on Carbon Emissions in China During 1978-2008," *Environmental Impact Assessment Review*, Vol. 36, pp. 1-8.

- Data Years: 1978-2008 (n = 31 Years)
- Dependent Variable – **C**arbon Emissions (million-tons)
- Independent Variables
  - **P**opulation (10,000s)
  - **U**rbanization Rate (%)
  - Percentage of Population of **W**orking Age (%)
  - Average **H**ousehold Size (persons/hhold)
  - Per Capita **E**xpenditures (Adjusted to Year=2000)

$$\ln\left(C_t\right) = \beta_0 + \beta_P \ln\left(P_t\right) + \beta_U \ln\left(U_t\right) + \beta_W \ln\left(W_t\right) + \beta_H \ln\left(H_t\right) + \beta_E \ln\left(E_t\right) + \varepsilon_t$$

Short-hand Notation: $Y_t = \beta_0 + \beta_1 X_{t1} + \beta_2 X_{t2} + \beta_3 X_{t3} + \beta_4 X_{t4} + \beta_5 X_{t5} + \varepsilon_t$

Note: R has some built-in packages for Ridge Regression, but I have not been able to make their numeric values to match what I get working "brute-force" or other statistical packages (like SAS). It is probably due to scaling the variables.

Data: Logarithms of original variables and correlation transformed values (also includes fitted values and residuals).

year	Y	X1	X2	X3	X4	X5	Y*	X1*	X2*	X3*	X4*	X5*	Y*-hat	e*
1978	3.7079	4.5671	2.8859	4.0860	1.5390	6.6067	-0.25530364	-0.33527	-0.31921	-0.35395	0.283565	-0.30929	-0.25559	0.0003
1979	3.7293	4.5803	2.9423	4.0943	1.5369	6.6733	-0.24676359	-0.31175	-0.28215	-0.3276	0.280433	-0.29053	-0.27144	0.0247
1980	3.7062	4.5922	2.9648	4.1026	1.5282	6.7593	-0.25599086	-0.29053	-0.26742	-0.30147	0.267839	-0.26634	-0.24384	-0.0122
1981	3.6961	4.6059	3.0037	4.1109	1.5129	6.8395	-0.26003988	-0.26617	-0.24183	-0.27555	0.245534	-0.24376	-0.22619	-0.0338
1982	3.7640	4.6215	3.0507	4.1190	1.5063	6.9048	-0.23289243	-0.23828	-0.21096	-0.24985	0.23587	-0.22538	-0.23547	0.0026
1983	3.8171	4.6348	3.0736	4.1331	1.4951	6.9838	-0.21167078	-0.21461	-0.1959	-0.20562	0.219618	-0.20313	-0.20772	-0.0039
1984	3.9006	4.6478	3.1359	4.1469	1.4839	7.0959	-0.17827626	-0.19143	-0.15496	-0.162	0.203183	-0.17158	-0.18825	0.0100
1985	3.9814	4.6620	3.1659	4.1608	1.4656	7.2226	-0.14596142	-0.1662	-0.13527	-0.11848	0.176496	-0.13592	-0.1325	-0.0135
1986	4.0316	4.6776	3.1995	4.1742	1.4446	7.2689	-0.12587795	-0.13849	-0.1132	-0.07605	0.145877	-0.12287	-0.12317	-0.0027
1987	4.0963	4.6941	3.2316	4.1875	1.4231	7.3265	-0.09997955	-0.1091	-0.09211	-0.03418	0.1146	-0.10667	-0.10782	0.0078
1988	4.1659	4.7098	3.2508	4.1919	1.3987	7.4012	-0.07216683	-0.08114	-0.07952	-0.02034	0.079044	-0.08563	-0.07521	0.0030
1989	4.1816	4.7247	3.2661	4.1964	1.3788	7.3994	-0.0658873	-0.05456	-0.06941	-0.00609	0.04996	-0.08615	-0.08604	0.0202
1990	4.1875	4.7391	3.2737	4.2008	1.3686	7.4354	-0.06351212	-0.029	-0.06442	0.007619	0.035198	-0.076	-0.08083	0.0173
1991	4.2363	4.7520	3.2936	4.1942	1.3584	7.5186	-0.04401771	-0.00594	-0.05137	-0.01321	0.020285	-0.05259	-0.06266	0.0186
1992	4.2786	4.7636	3.3127	4.1927	1.3481	7.6430	-0.02708921	0.014687	-0.03881	-0.01796	0.005217	-0.01757	-0.01545	-0.0116
1993	4.3441	4.7751	3.3318	4.2002	1.3376	7.7240	-0.00091241	0.035081	-0.02625	0.005731	-0.01001	0.005227	0.013989	-0.0149
1994	4.4044	4.7862	3.3503	4.1987	1.3297	7.7694	0.023216	0.054948	-0.01415	0.001007	-0.02153	0.017999	0.015919	0.0073
1995	4.4827	4.7968	3.3687	4.2077	1.3191	7.8450	0.054514694	0.073714	-0.00205	0.029247	-0.03704	0.039292	0.045549	0.0090
1996	4.5283	4.8072	3.4171	4.2077	1.3137	7.9349	0.072760914	0.092284	0.029744	0.029247	-0.04486	0.064583	0.050824	0.0219
1997	4.5162	4.8173	3.4629	4.2121	1.2920	7.9790	0.067938194	0.11023	0.059865	0.043273	-0.07655	0.077004	0.054519	0.0134
1998	4.4614	4.8264	3.5071	4.2136	1.2892	8.0362	0.04601728	0.126429	0.088863	0.047934	-0.08056	0.09312	0.0449	0.0011
1999	4.5053	4.8346	3.5490	4.2151	1.2754	8.1155	0.063587225	0.141066	0.116446	0.052589	-0.10078	0.115434	0.062956	0.0006
2000	4.5314	4.8421	3.5896	4.2506	1.2355	8.1975	0.07401138	0.154461	0.143099	0.164526	-0.15893	0.138521	0.136298	-0.0623
2001	4.5553	4.8491	3.6286	4.2542	1.2296	8.2571	0.083582663	0.166919	0.168713	0.175727	-0.16743	0.155294	0.139825	-0.0562
2002	4.6147	4.8555	3.6659	4.2528	1.2208	8.3248	0.107327379	0.17832	0.193197	0.171252	-0.18027	0.174349	0.150117	-0.0428
2003	4.7768	4.8616	3.7020	4.2542	1.2179	8.3928	0.172130411	0.189098	0.216964	0.175727	-0.18458	0.193474	0.157121	0.0150
2004	4.9350	4.8675	3.7319	4.2616	1.1969	8.4707	0.23540028	0.199538	0.236605	0.1989	-0.21509	0.21542	0.198474	0.0369
2005	5.0332	4.8734	3.7610	4.2772	1.1756	8.5452	0.274672344	0.210052	0.255676	0.248238	-0.24625	0.236382	0.244348	0.0303
2006	5.1148	4.8786	3.7819	4.2811	1.1537	8.6369	0.307295448	0.219422	0.269437	0.260453	-0.27809	0.262203	0.299223	0.0081
2007	5.1939	4.8838	3.8053	4.2840	1.1537	8.7386	0.338946778	0.228608	0.28482	0.269583	-0.27809	0.290815	0.33276	0.0062
2008	5.2589	4.8888	3.8217	4.2877	1.1506	8.8220	0.364940948	0.237612	0.29555	0.281282	-0.28269	0.314306	0.365357	-0.0004

Correlation Transformation

$$Y_t^* = \frac{Y_t - \overline{Y}}{s_Y\sqrt{n-1}} \qquad X_{tj}^* = \frac{X_{tj} - \overline{X}_j}{s_j\sqrt{n-1}} \qquad \overline{X}_j = \frac{\sum_{t=1}^{n} X_{tj}}{n} \qquad s_j^2 = \frac{\sum_{t=1}^{n}\left(X_{tj} - \overline{X}_j\right)}{n-1}$$

$$\mathbf{X}^* = \begin{bmatrix} X_{11}^* & X_{12}^* & \cdots & X_{15}^* \\ X_{21}^* & X_{22}^* & \cdots & X_{25}^* \\ \vdots & \vdots & \ddots & \vdots \\ X_{31,1}^* & X_{31,2}^* & \cdots & X_{31,5}^* \end{bmatrix} \qquad \mathbf{Y}^* = \begin{bmatrix} Y_1^* \\ Y_2^* \\ \vdots \\ Y_{31}^* \end{bmatrix}$$

$$\mathbf{X}^{*\prime}\mathbf{X}^* = \mathbf{R}_{XX} = \begin{bmatrix} 1 & r_{12} & \cdots & r_{15} \\ r_{21} & 1 & \cdots & r_{25} \\ \vdots & \vdots & \ddots & \vdots \\ r_{51} & r_{52} & \cdots & 1 \end{bmatrix} \qquad \mathbf{X}^{*\prime}\mathbf{Y}^* = \mathbf{R}_{XY} = \begin{bmatrix} r_{Y1} \\ r_{Y2} \\ \vdots \\ r_{Y5} \end{bmatrix}$$

R_XX=X*'X*						R_XY=X*'Y*	
1	0.97534	0.971243	-0.98733	0.984727		0.952534	
0.97534	1	0.98011	-0.99068	0.995225		0.9748024	
0.971243	0.98011	1	-0.98021	0.97729		0.9601475	
-0.98733	-0.99068	-0.98021	1	-0.9942		-0.978609	
0.984727	0.995225	0.97729	-0.9942	1		0.9825869	
INV(R_XX)						b*_ols	se(b_ols)
50.6167	36.87374	-9.84813	32.98043	-44.1276		-0.9311	0.1764
36.87374	147.7218	-27.8081	21.49857	-134.777		-1.0452	0.3013
-9.84813	-27.8081	31.39742	13.30218	19.91369		0.2074	0.1389
32.98043	21.49857	13.30218	122.3776	54.79551		-0.7746	0.2742
-44.1276	-134.777	19.91369	54.79551	213.6031		1.9668	0.3623
SSE	MSE	s					
0.015977	0.000615	0.024789					

Note: The VIF^s are the diagonal elements of $\mathbf{R}_{XX}^{-1}$:

VIF_1=50.6, VIF_2=147.7, VIF_3=31.4, VIF_4=122.4, VIF_5=213.6 all much larger than 10.

Formulas for regression coefficients and their estimated variance-covariance matrices:

Ordinary Least Squares Estimator (Special case of Ridge Estimator with $c = 0$):

$$\hat{\boldsymbol{\gamma}}_{\mathbf{OLS}} = \left(\mathbf{X}^{*\prime}\mathbf{X}^{*}\right)^{-1}\mathbf{X}^{*\prime}\mathbf{Y}^{*} \qquad s^2\left\{\hat{\boldsymbol{\gamma}}_{\mathbf{OLS}}\right\} = MSE_{OLS}\left(\mathbf{X}^{*\prime}\mathbf{X}^{*}\right)^{-1}$$

$$MSE_{OLS} = \frac{\left(\mathbf{Y}^{*} - \mathbf{X}^{*}\hat{\boldsymbol{\gamma}}_{\mathbf{OLS}}\right)'\left(\mathbf{Y}^{*} - \mathbf{X}^{*}\hat{\boldsymbol{\gamma}}_{\mathbf{OLS}}\right)}{n-p}$$

Ridge Estimator:

$$\hat{\boldsymbol{\gamma}}_{\mathbf{R}} = \left(\mathbf{X}^{*\prime}\mathbf{X}^{*} + c\mathbf{I}\right)^{-1}\mathbf{X}^{*\prime}\mathbf{Y}^{*} \qquad s^2\left\{\hat{\boldsymbol{\gamma}}_{\mathbf{R}}\right\} = MSE_c\left(\mathbf{X}^{*\prime}\mathbf{X}^{*} + c\mathbf{I}\right)^{-1}\left(\mathbf{X}^{*\prime}\mathbf{X}^{*}\right)\left(\mathbf{X}^{*\prime}\mathbf{X}^{*} + c\mathbf{I}\right)^{-1} \qquad c \geq 0$$

$$MSE_c \equiv \frac{\left(\mathbf{Y}^{*} - \mathbf{X}^{*}\hat{\boldsymbol{\gamma}}_{\mathbf{R}}(c)\right)'\left(\mathbf{Y}^{*} - \mathbf{X}^{*}\hat{\boldsymbol{\gamma}}_{\mathbf{R}}(c)\right)}{n-p}$$

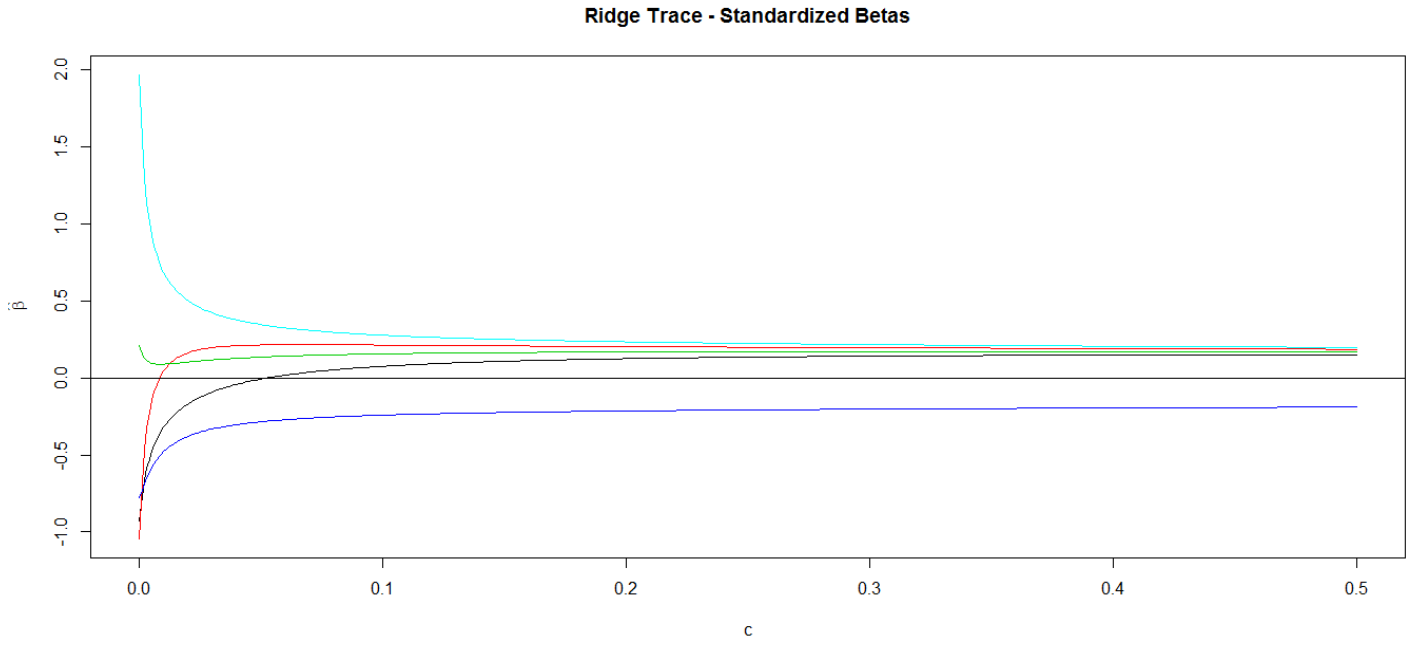
R Program

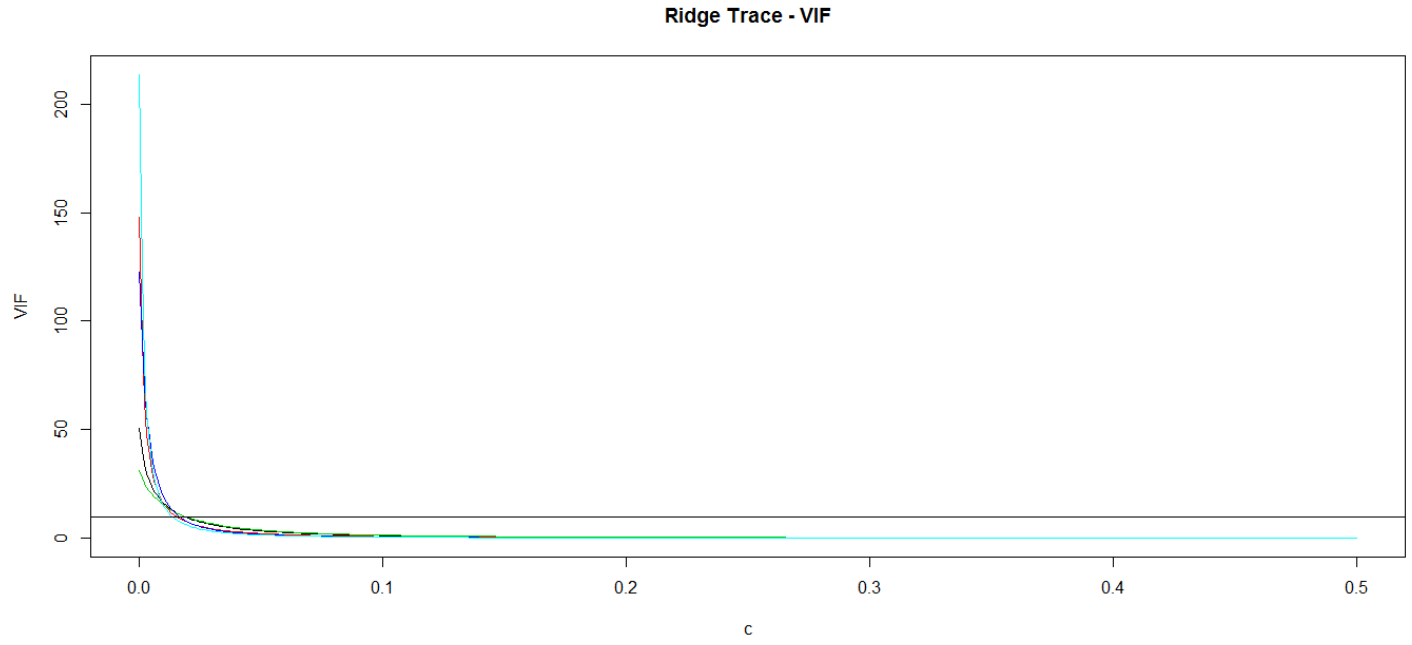
```
china.carbon <- read.table("http://www.stat.ufl.edu/~winner/data/china_carbon.dat",header=F,
    col.names=c("year","emit","pop","urban","workage","hhsize","pcexp"))

attach(china.carbon)

#### Obtain Mean and SD for each variable (log transformed data)
m.emit <- mean(log(emit));  s.emit <- sd(log(emit))
m.pop <- mean(log(pop));  s.pop <- sd(log(pop))
m.urban <- mean(log(urban));   s.urban <- sd(log(urban))
m.workage <- mean(log(workage));  s.workage <- sd(log(workage))
m.hhsize <- mean(log(hhsize));  s.hhsize <- sd(log(hhsize))
m.pcexp <- mean(log(pcexp));  s.pcexp <- sd(log(pcexp))

#### Correlation Transformation
z.emit <- (log(emit)-m.emit)/(sqrt(30)*s.emit)
z.pop <- (log(pop)-m.pop)/(sqrt(30)*s.pop)
z.urban <- (log(urban)-m.urban)/(sqrt(30)*s.urban)
z.workage <- (log(workage)-m.workage)/(sqrt(30)*s.workage)
z.hhsize <- (log(hhsize)-m.hhsize)/(sqrt(30)*s.hhsize)
z.pcexp <- (log(pcexp)-m.pcexp)/(sqrt(30)*s.pcexp)

#### X matrix (no intercept) and Y vector for Y* and X*
XC <- as.matrix(cbind(z.pop,z.urban,z.workage,z.hhsize,z.pcexp))
YC <- as.matrix(z.emit)

#### Standardized OLS Regression
cc.ols.std <- lm(YC ~ XC - 1)
summary(cc.ols.std)


IR <- diag(ncol(XC))         ### Identity matrix (5x5)
p <- ncol(XC)                ### Number of predictors = 5 (no intercept model)
n <- length(emit)            ### Sample size = 31
min_c <- 0;  max_c <- 0.50; step_c <- 0.001    ### Range of c values
num_c <- length(seq(min_c,max_c,step_c))       ### Number of c values

#### Initialize matrices and vectors to save regression coefficients, VIFs, SSEs
beta <- matrix(rep(0,num_c*p),ncol=p)
VIF <- matrix(rep(0,num_c*p),ncol=p)
SSE <- rep(0,num_c)
iter <- 0

### Loop through all c values and obtain b_Ridge, VIF, SSE
for (c in seq(min_c,max_c,step_c)) {
iter <- iter+1

R.INVXX <- solve((t(XC)%*%XC) + (c*IR))     #### INV(X*'X+cI)
beta_R <- R.INVXX %*%  t(XC) %*% YC         #### b_Ridge

e <- YC - (XC %*% beta_R)                   #### Residual Vector

### Ridge VIF =INV(XC'XC + kI) (XC'XC) INV(XC'XC + kI)  - Marquardt 1970
VIF_R <- R.INVXX %*% t(XC) %*% XC %*% R.INVXX     #### VIF_Ridge matrix
SSE[iter] <- t(e) %*% e                           #### SSE = e'e for this row

for (i in 1:p) {
beta[iter,i] <- beta_R[i]       #### Save Regression coeffs to this row of beta
VIF[iter,i] <- VIF_R[i,i]       #### Save VIF values to this row of VIF
}
}
```

Continued Below

```
RMSE <- sqrt(SSE/(n-p))       #### Compute matrix of s = sqrt(MSE) values

#### Generate Ridge Trace for Standardized Betas
cplot <- matrix(rep(seq(min_c,max_c,step_c),each=p),byrow=T,ncol=p)
matplot(cplot,beta,type="l",lty=1,
  xlab="c",ylab=expression(hat(beta)),main="Ridge Trace - Standardized Betas")
abline(h=0)

#### Generate Ridge Trace for VIFs
cplot <- matrix(rep(seq(min_c,max_c,step_c),each=p),byrow=T,ncol=p)
matplot(cplot,VIF,type="l",lty=1,
  xlab="c",ylab="VIF",main="Ridge Trace - VIF")
abline(h=10)


####### Estimates and standard errors when c=.20
R.INVXX20 <- solve((t(XC)%*%XC) + (0.20*IR))     #### INV(X*'X+0.20I)
beta_R20 <- R.INVXX20 %*%  t(XC) %*% YC          #### b_Ridge(c=0.20)

e20 <- YC - (XC %*% beta_R20)                    #### Residual Vector (c=0.20)
MSE20 <- (t(e20) %*% e20)/(n-p)                  #### MSE (c=.20)
s2.b20 <- MSE20[1,1] * R.INVXX20 %*% t(XC)%*%XC %*% R.INVXX20
s.b20 <- sqrt(diag(s2.b20))
t.b20 <- beta_R20/s.b20

print(round(cbind(beta_R20,s.b20,t.b20),5))

######## Output #############

> summary(cc.ols.std)

Call:
lm(formula = YC ~ XC - 1)

Residuals:
      Min        1Q    Median        3Q       Max
-0.062286 -0.007796  0.003041  0.014214  0.036926

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
XCz.pop      -0.9311     0.1764  -5.279 1.61e-05 ***
XCz.urban    -1.0452     0.3013  -3.469  0.00184 **
XCz.workage   0.2074     0.1389   1.493  0.14741
XCz.hhsize   -0.7746     0.2742  -2.825  0.00897 **
XCz.pcexp     1.9668     0.3623   5.429 1.09e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02479 on 26 degrees of freedom
Multiple R-squared:  0.984,     Adjusted R-squared:  0.981
F-statistic: 320.3 on 5 and 26 DF,  p-value: < 2.2e-16


> print(round(cbind(beta_R20,s.b20,t.b20),5))
                    s.b20
z.pop      0.12446 0.02692   4.62342
z.urban    0.20283 0.02086   9.72492
z.workage  0.16779 0.02899   5.78852
z.hhsize  -0.21494 0.01733 -12.40559
z.pcexp    0.23375 0.01724  13.55786
>
```

From the plots below, the betas stabilize by about c = .20, VIFs stabilize earlier. Authors chose c = .20.





The standard errors of the estimated standardized regression coefficients have decreased by orders of magnitude of about 10, based on using Ridge Regression.

Robust Regression for Influential Cases

- Influential cases, after having been ruled out as recording errors or indicative of new predictors, can be reduced in terms of their impacts on the regression model
- Two commonly applied Robust Regression Models:
  - **Least Absolute Residuals (LAR) Regression** – Choose the coefficients that minimize sum of absolute deviations (as opposed to squared deviations in OLS). No closed form solutions, must use specialized programs (can be run in R, using the **quantreg** package).
  - **IRLS Robust Regression** – Uses Iteratively Re-weighted Least Squares where weights are based on how much each case is an outlier (lower weights for larger outliers)

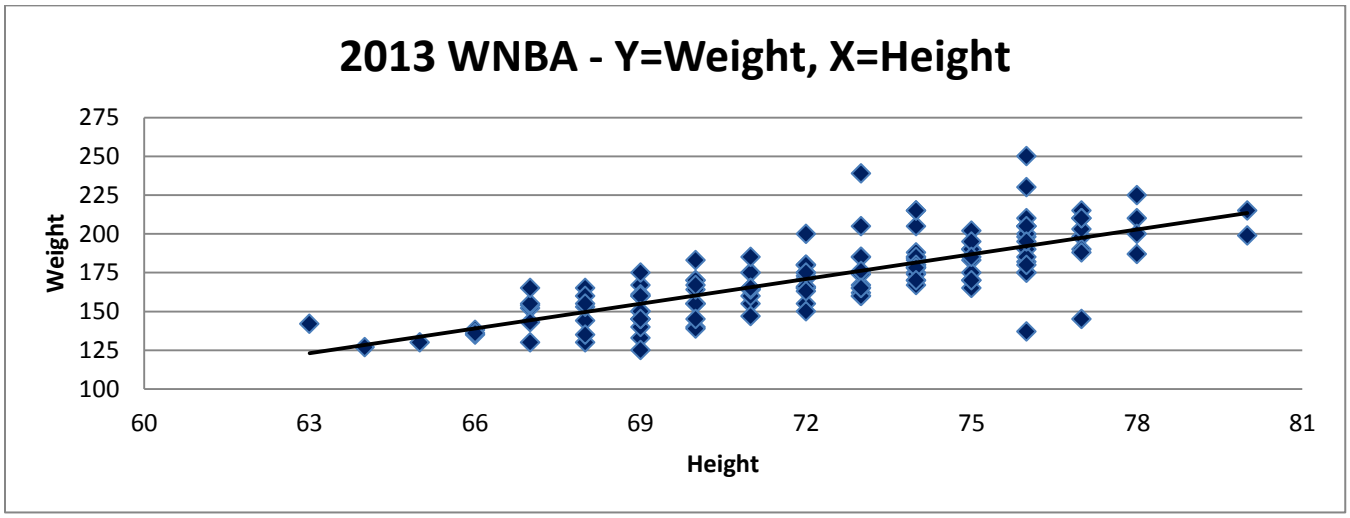
Example: Heights and Weights of 2013 WNBA Professional Basketball Players

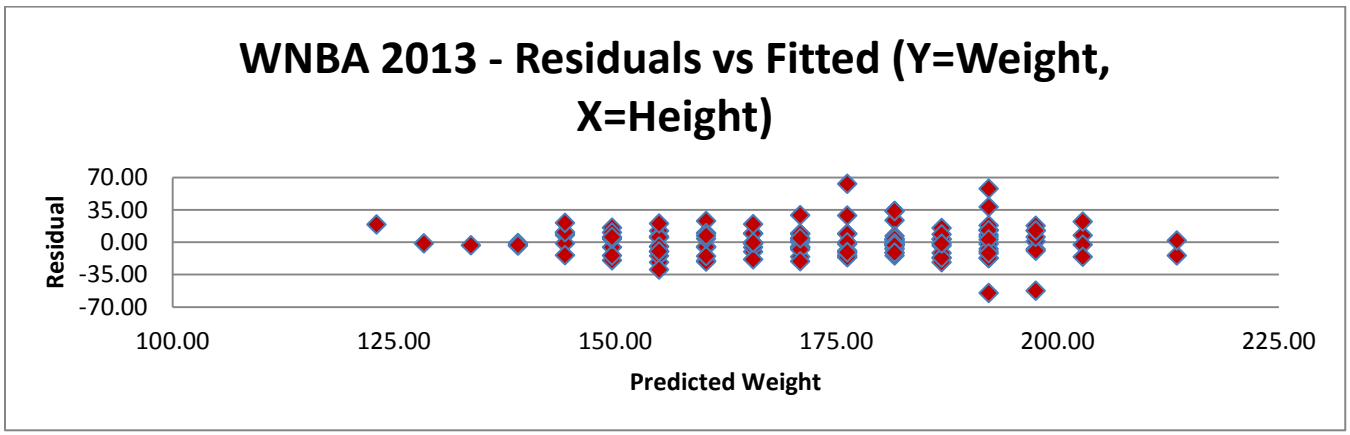
Source: www.wnba.com

Response Variable: Y = Weight (lbs) X = Height (inches) Presumably, you can control weight, not height

	Coefficients	*Standard Erro*	*t Stat*	*P-value*
Intercept	-212.05	28.78	-7.37	0.0000
Height	5.32	0.40	13.39	0.0000

$\hat{Y} = -212.05 + 5.32X \qquad r^2 = .5670$





Data: Height, Weight, Predicted Weight Residual, Hat Value, Studentized Deleted Residual

ID	Ht	Wt	Y-hat_i	e_i	h_i	t_i	ID	Ht	Wt	Y-hat_i	e_i	h_i	t_i	ID	Ht	Wt	Y-hat_i	e_i	h_i	t_i
1	77	198	197.55	0.45	0.0203	0.03	48	76	200	192.23	7.77	0.0152	0.49	95	69	175	154.99	20.01	0.0143	1.26
2	67	152	144.36	7.64	0.0251	0.48	49	71	155	165.63	-10.63	0.0084	-0.67	96	76	230	192.23	37.77	0.0152	2.42
3	69	145	154.99	-9.99	0.0143	-0.63	50	65	130	133.72	-3.72	0.0409	-0.24	97	73	165	176.27	-11.27	0.0074	-0.71
4	76	205	192.23	12.77	0.0152	0.80	51	74	180	181.59	-1.59	0.0088	-0.10	98	70	164	160.31	3.69	0.0107	0.23
5	77	190	197.55	-7.55	0.0203	-0.48	52	72	165	170.95	-5.95	0.0073	-0.37	99	72	150	170.95	-20.95	0.0073	-1.32
6	70	155	160.31	-5.31	0.0107	-0.33	53	73	185	176.27	8.73	0.0074	0.55	100	72	180	170.95	9.05	0.0073	0.57
7	76	175	192.23	-17.23	0.0152	-1.09	54	69	160	154.99	5.01	0.0143	0.31	101	76	137	192.23	-55.23	0.0152	-3.63
8	69	133	154.99	-21.99	0.0143	-1.39	55	70	155	160.31	-5.31	0.0107	-0.33	102	77	210	197.55	12.45	0.0203	0.78
9	73	160	176.27	-16.27	0.0074	-1.02	56	71	175	165.63	9.37	0.0084	0.59	103	80	199	213.51	-14.51	0.0428	-0.93
10	72	200	170.95	29.05	0.0073	1.84	57	74	178	181.59	-3.59	0.0088	-0.22	104	71	185	165.63	19.37	0.0084	1.22
11	72	155	170.95	-15.95	0.0073	-1.00	58	78	225	202.87	22.13	0.0265	1.41	105	74	185	181.59	3.41	0.0088	0.21
12	74	170	181.59	-11.59	0.0088	-0.73	59	74	185	181.59	3.41	0.0088	0.21	106	75	183	186.91	-3.91	0.0114	-0.25
13	71	164	165.63	-1.63	0.0084	-0.10	60	77	145	197.55	-52.55	0.0203	-3.45	107	68	155	149.67	5.33	0.0191	0.33
14	73	162	176.27	-14.27	0.0074	-0.89	61	70	145	160.31	-15.31	0.0107	-0.96	108	69	125	154.99	-29.99	0.0143	-1.91
15	69	167	154.99	12.01	0.0143	0.75	62	67	155	144.36	10.64	0.0251	0.67	109	74	174	181.59	-7.59	0.0088	-0.48
16	77	188	197.55	-9.55	0.0203	-0.60	63	76	205	192.23	12.77	0.0152	0.80	110	73	239	176.27	62.73	0.0074	4.16
17	70	140	160.31	-20.31	0.0107	-1.28	64	77	203	197.55	5.45	0.0203	0.34	111	66	136	139.04	-3.04	0.0324	-0.19
18	75	202	186.91	15.09	0.0114	0.95	65	66	138	139.04	-1.04	0.0324	-0.07	112	76	210	192.23	17.77	0.0152	1.12
19	68	130	149.67	-19.67	0.0191	-1.24	66	68	160	149.67	10.33	0.0191	0.65	113	73	165	176.27	-11.27	0.0074	-0.71
20	74	183	181.59	1.41	0.0088	0.09	67	72	173	170.95	2.05	0.0073	0.13	114	76	205	192.23	12.77	0.0152	0.80
21	78	210	202.87	7.13	0.0265	0.45	68	76	182	192.23	-10.23	0.0152	-0.64	115	71	164	165.63	-1.63	0.0084	-0.10
22	78	200	202.87	-2.87	0.0265	-0.18	69	70	183	160.31	22.69	0.0107	1.43	116	63	142	123.08	18.92	0.0616	1.22
23	66	135	139.04	-4.04	0.0324	-0.26	70	71	160	165.63	-5.63	0.0084	-0.35	117	74	178	181.59	-3.59	0.0088	-0.22
24	74	179	181.59	-2.59	0.0088	-0.16	71	72	180	170.95	9.05	0.0073	0.57	118	69	150	154.99	-4.99	0.0143	-0.31
25	76	190	192.23	-2.23	0.0152	-0.14	72	69	145	154.99	-9.99	0.0143	-0.63	119	76	195	192.23	2.77	0.0152	0.17
26	75	190	186.91	3.09	0.0114	0.19	73	76	175	192.23	-17.23	0.0152	-1.09	120	78	187	202.87	-15.87	0.0265	-1.01
27	67	154	144.36	9.64	0.0251	0.61	74	73	174	176.27	-2.27	0.0074	-0.14	121	74	180	181.59	-1.59	0.0088	-0.10
28	69	150	154.99	-4.99	0.0143	-0.31	75	68	135	149.67	-14.67	0.0191	-0.93	122	75	195	186.91	8.09	0.0114	0.51
29	67	143	144.36	-1.36	0.0251	-0.09	76	74	180	181.59	-1.59	0.0088	-0.10	123	71	165	165.63	-0.63	0.0084	-0.04
30	70	139	160.31	-21.31	0.0107	-1.34	77	74	215	181.59	33.41	0.0088	2.12	124	75	165	186.91	-21.91	0.0114	-1.38
31	76	198	192.23	5.77	0.0152	0.36	78	67	130	144.36	-14.36	0.0251	-0.91	125	70	167	160.31	6.69	0.0107	0.42
32	70	170	160.31	9.69	0.0107	0.61	79	73	160	176.27	-16.27	0.0074	-1.02	126	74	178	181.59	-3.59	0.0088	-0.22
33	71	175	165.63	9.37	0.0084	0.59	80	74	188	181.59	6.41	0.0088	0.40	127	76	180	192.23	-12.23	0.0152	-0.77
34	75	190	186.91	3.09	0.0114	0.19	81	76	185	192.23	-7.23	0.0152	-0.45	128	72	175	170.95	4.05	0.0073	0.25
35	70	155	160.31	-5.31	0.0107	-0.33	82	68	153	149.67	3.33	0.0191	0.21	129	67	165	144.36	20.64	0.0251	1.31
36	73	205	176.27	28.73	0.0074	1.82	83	74	215	181.59	33.41	0.0088	2.12	130	71	147	165.63	-18.63	0.0084	-1.17
37	69	140	154.99	-14.99	0.0143	-0.94	84	77	210	197.55	12.45	0.0203	0.78	131	76	250	192.23	57.77	0.0152	3.81
38	72	180	170.95	9.05	0.0073	0.57	85	73	176	176.27	-0.27	0.0074	-0.02	132	69	145	154.99	-9.99	0.0143	-0.63
39	74	205	181.59	23.41	0.0088	1.48	86	69	161	154.99	6.01	0.0143	0.38	133	80	215	213.51	1.49	0.0428	0.09
40	68	165	149.67	15.33	0.0191	0.97	87	74	167	181.59	-14.59	0.0088	-0.92	134	64	127	128.40	-1.40	0.0506	-0.09
41	68	144	149.67	-5.67	0.0191	-0.36	88	69	160	154.99	5.01	0.0143	0.31	135	71	165	165.63	-0.63	0.0084	-0.04
42	72	165	170.95	-5.95	0.0073	-0.37	89	70	170	160.31	9.69	0.0107	0.61	136	75	175	186.91	-11.91	0.0114	-0.75
43	73	167	176.27	-9.27	0.0074	-0.58	90	72	175	170.95	4.05	0.0073	0.25	137	75	170	186.91	-16.91	0.0114	-1.06
44	72	171	170.95	0.05	0.0073	0.00	91	74	184	181.59	2.41	0.0088	0.15	138	74	170	181.59	-11.59	0.0088	-0.73
45	77	215	197.55	17.45	0.0203	1.10	92	72	166	170.95	-4.95	0.0073	-0.31	139	75	185	186.91	-1.91	0.0114	-0.12
46	75	170	186.91	-16.91	0.0114	-1.06	93	72	163	170.95	-7.95	0.0073	-0.50							
47	73	185	176.27	8.73	0.0074	0.55	94	74	175	181.59	-6.59	0.0088	-0.41							

With respect to the Studentized Deleted Residuals, four players have values over 3 in absolute value:

- $t_{110} = 4.16$, $t_{131} = 3.81$, $t_{101} = -3.63$, $t_{60} = -3.45$

To determine whether they effected the regression excessively, we will obtain regression coefficient estimates based on least absolute deviations, using R.

Least absolute deviation estimator minimizes: $Q_{LAD} = \sum_{i=1}^{n} \left| Y_i - \left(\beta_0 + \beta_1 X_{i1} + ... + \beta_{p-1} X_{i,p-1} \right) \right|$

R Program and Output

```
wnba <- read.csv("E:\\blue_drive\\sta4210\\wnba_ht_wt.csv", header=TRUE)
attach(wnba); names(wnba)

wnba.ols <- lm(Weight ~ Height)
summary(wnba.ols)
install.packages("quantreg")
library(quantreg)
wnba.lad <- rq(Weight ~ Height,0.5)
summary(wnba.lad)

##### Output ############

> summary(wnba.ols)

Call: lm(formula = Weight ~ Height)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -212.0523    28.7833  -7.367 1.48e-11 ***
Height         5.3195     0.3971  13.395  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 16 on 137 degrees of freedom
Multiple R-squared:  0.567,     Adjusted R-squared:  0.5639
F-statistic: 179.4 on 1 and 137 DF,  p-value: < 2.2e-16

> summary(wnba.lad)

Call: rq(formula = Weight ~ Height, tau = 0.5)

tau: [1] 0.5

Coefficients:
            coefficients lower bd   upper bd
(Intercept) -212.20000   -240.32621 -164.02269
Height         5.30000      4.55328    5.69293
```

Note that for this example, there is very little difference in the estimates based on ordinary least squares and least absolute deviations.

OLS: $\hat{Y}_{OLS} = -212.0523 + 5.3195X$ LAD: $\hat{Y}_{LAD} = -212.2000 + 5.3000X$

IRLS – Robust Regression

1. Choose weight function (we will use Huber here)
2. Obtain starting weights for each case.
3. Use starting weights in Weighted Least Squares and obtain residuals after fitting equation
4. Use Residuals from step 3 to generate new weights.
5. Continue iterations to convergence (small changes in estimated regression coefficients or residuals or fitted values)

Applying Procedure for Huber weight function (1.345 is tuning parameter to achieve high efficiency to normal model)

1. Huber weight function: $w = \begin{cases} 1 & |u| \le 1.345 \\ \dfrac{1.345}{|u|} & |u| > 1.345 \end{cases}$

2. Use initial residuals from OLS fit after scaling by Median Absolute Deviation (Robust estimate of σ):

$$MAD = \frac{1}{0.6745}\text{median}\left\{\left|e_i - \text{median}\left\{e_i\right\}\right|\right\} \qquad u_i = \frac{e_i}{MAD}$$

3. Iterate to Convergence

Note: Bisquare weight Function: $w = \begin{cases} \left[1 - \left(\dfrac{u}{4.685}\right)^2\right]^2 & |u| \le 4.685 \\ 0 & |u| > 4.685 \end{cases}$

Example: Ohio Wheat Condition by Meteorological Conditions

We re-fit the Ohio wheat regression relating Wheat condition to 4 meteorological factors. Here we include 4 models: ordinary least squares, least absolute deviations (quantreg package), default robust (MASS package), and a brute force IRLS method with Huber weights.

```
wheat <- read.csv("E:\\blue_drive\\sta4210\\ohio_wheat.csv",
       header=TRUE)
attach(wheat); names(wheat)

wheat.mod1 <- lm(whtcnd.y ~ tempon.x1 + rains.x2 + rainon.x3 + sunon.x4)
summary(wheat.mod1)

library(quantreg)
wheat.lad <- rq(whtcnd.y ~ tempon.x1 + rains.x2 + rainon.x3 + sunon.x4)
summary(wheat.lad)

library(MASS)
huber.rreg1 <- rlm(whtcnd.y ~ tempon.x1 + rains.x2 + rainon.x3 + sunon.x4)
summary(huber.rreg1)
```

Continued Below

```
################ Begin brute-force IRWLS - Huber Method
n <- length(whtcnd.y)
w <- rep(1,n)
irwls.1 <- lm(whtcnd.y ~ tempon.x1 + rains.x2 + rainon.x3 + sunon.x4)
res1 <- residuals(irwls.1)
b.old <- coef(irwls.1)
MAD <- (1/0.6745)*median(abs(res1-median(res1)))
u <- res1/MAD
for(i in 1:n)  w[i] <- min(1,1.345/abs(u[i]))
delta_b <- 100.0
num.iter=0
while (delta_b > 0.000001) {
num.iter <- num.iter + 1
irwls.2 <- lm(whtcnd.y ~ tempon.x1 + rains.x2 + rainon.x3 + sunon.x4,weights=w)
res2 <- residuals(irwls.2)
b.new <- coef(irwls.2)
MAD <- (1/0.6745)*median(abs(res2-median(res2)))
u <- res2/MAD
for(i in 1:n)  w[i] <- min(1,1.345/abs(u[i]))
delta_b <- sum((b.new-b.old)^2)
b.old <- b.new
}
num.iter
summary(irwls.2)

###### Output ####
> wheat.mod1 <- lm(whtcnd.y ~ tempon.x1 + rains.x2 + rainon.x3 + sunon.x4)
> summary(wheat.mod1)
Call: lm(formula = whtcnd.y ~ tempon.x1 + rains.x2 + rainon.x3 + sunon.x4)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  27.3031    40.2531   0.678    0.506
tempon.x1     1.1556     0.7846   1.473    0.157
rains.x2      2.1818     1.5189   1.436    0.167
rainon.x3     2.3578     1.4669   1.607    0.124
sunon.x4     -0.2373     0.3418  -0.694    0.496
Residual standard error: 8.463 on 19 degrees of freedom
> wheat.lad <- rq(whtcnd.y ~ tempon.x1 + rains.x2 + rainon.x3 + sunon.x4)
> summary(wheat.lad)
Call: rq(formula = whtcnd.y ~ tempon.x1 + rains.x2 + rainon.x3 + sunon.x4)
tau: [1] 0.5
Coefficients:
            coefficients lower bd  upper bd
(Intercept)  -0.74925    -17.88000 111.95127
tempon.x1     1.53782     -0.99551   1.94836
rains.x2      0.71739     -0.12597   7.13836
rainon.x3     3.57183     -3.89182   5.68620
sunon.x4     -0.04809     -0.91444   0.17960
>
> library(MASS)
> huber.rreg1 <- rlm(whtcnd.y ~ tempon.x1 + rains.x2 + rainon.x3 + sunon.x4)
> summary(huber.rreg1)
Call: rlm(formula = whtcnd.y ~ tempon.x1 + rains.x2 + rainon.x3 + sunon.x4)
Coefficients:
            Value   Std. Error t value
(Intercept) 29.1599 38.2836     0.7617
tempon.x1    1.0677  0.7462     1.4308
rains.x2     2.6192  1.4446     1.8131
rainon.x3    2.7366  1.3951     1.9615
sunon.x4    -0.2461  0.3251    -0.7569

Residual standard error: 8.846 on 19 degrees of freedom
> summary(irwls.2)
Call:lm(formula = whtcnd.y ~ tempon.x1 + rains.x2 + rainon.x3 + sunon.x4, weights = w)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  29.5713    37.4541   0.790   0.4395
tempon.x1     1.0554     0.7320   1.442   0.1657
rains.x2      2.6584     1.4475   1.837   0.0820 .
rainon.x3     2.7865     1.3899   2.005   0.0594 .
sunon.x4     -0.2486     0.3182  -0.781   0.4443
Residual standard error: 7.868 on 19 degrees of freedom
```

In the section on Influence Measures, case 18 had large negative DFBETAS for X_2 (Rainfall in September) and X_3 (Rainfall in Oct/Nov). The negative values meant that the coefficients were higher when that case was removed, than when it was included. Note that for OLS, b_2 and b_3 were 2.1818 and 2.3578, respectively. For IRWLS they increase to 2.6584 and 2.7865, respectively. This reflects the decrease in the influence of case 18 on the regression coefficients due to decreasing its weight.

Nonparametric Regression

- **Locally Weighted Regressions (Lowess)**
  - Works best with few predictors, and when data have been transformed to normality with constant error variances, and predictors have been scaled by their standard deviation (sqrt(*MSE*) or *MAD* when outliers are present)
  - Applies WLS with weights as distances from the individual points in a neighborhood to the target (q = proportion of data used in the neighborhood, typically 0.40 to 0.60)
- **Regression Trees**
  - *X*-space is broken down into multiple sub-spaces (1-step at a time), and each region is modeled by the mean response
  - Each step is chosen to minimize the within region sum of squares

Lowess Method

- Assumes model has been selected and built so assumptions of errors being approximately normal with constant variance holds.
- Predictors of different units should be scaled by SD or MAD in distance measure

Assuming p-1=2 predictor variables and $q \equiv$ proportion of sample used at each point:

Distance Measure (Each point from target point): $d_i = \sqrt{\left(\frac{X_{i1} - X_{h1}}{s_1}\right)^2 + \left(\frac{X_{i2} - X_{h2}}{s_2}\right)^2}$

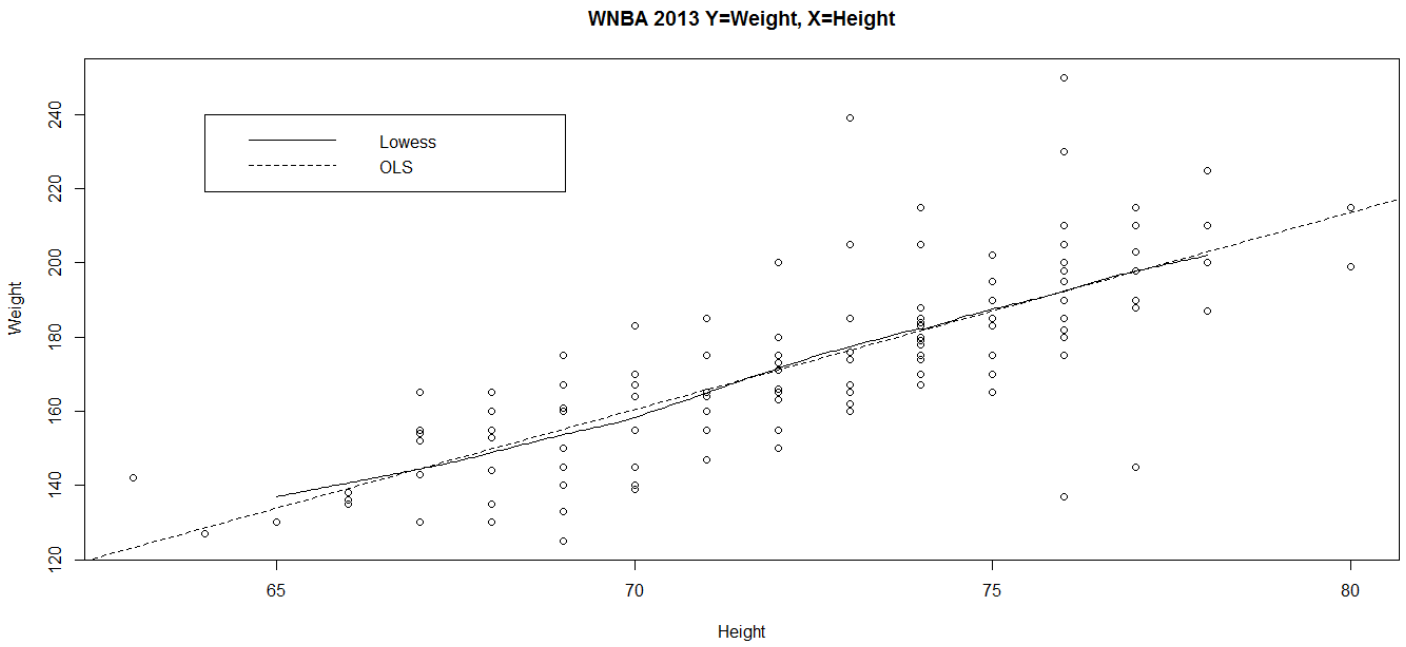
Weight Function: $w_i = \begin{cases} \left[1 - \left(d_i / d_q\right)^3\right]^3 & d_i < d_q \\ 0 & d_i \geq d_q \end{cases}$

$d_q \equiv$ size of neighborhood so q of sample is included

Local Fitting: Fit first-order or second-order model by WLS and get fitted value at each target point

Example with p-1=1 Predictor: WNBA Heights and Weights

```
wnba <- read.csv("E:\\blue_drive\\sta4210\\wnba_ht_wt.csv",
       header=TRUE)
attach(wnba); names(wnba)

wnba.ols <- lm(Weight ~ Height)
summary(wnba.ols)


#### Generate Lowess Curve
Y <- Weight
X1 <- Height
n <- length(Y)
qn <- floor(n/2)
sd1 <- sd(X1);
X <- as.matrix(cbind(rep(1,n),X1))
Y <- as.matrix(Y)
Yhat <- matrix(rep(0,length(seq(65,78,0.05))),ncol=1)
rownum <- 0
for (X1h in seq(65,78,0.05)) {
rownum <- rownum + 1
X_h <- as.matrix(cbind(1,X1h))
d <- sqrt(((X1-X1h)/sd1)^2)
dq <- d[rank(d,ties.method="random") == qn]
w <- rep(0,n)
for (i in 1:n) {
if (d[i] < dq) w[i] <- (1-(d[i]/dq)^3)^3
}
w <- diag(w)
Yhat[rownum] <- X_h %*%solve(t(X) %*% w %*% X) %*%
                        (t(X) %*% w %*% Y)
}
X1h <- seq(65,78,0.05)
plot(Height,Weight,main="WNBA 2013 Y=Weight, X=Height")
lines(X1h,Yhat,type="l")
abline(wnba.ols,lty=2)
legend(64,240,c("Lowess","OLS"),lty=c(1,2))
```



Example with p-1=2 Predictors: Texas January Mean Temperature by LAT, ELEV

Source: www.noaa.gov

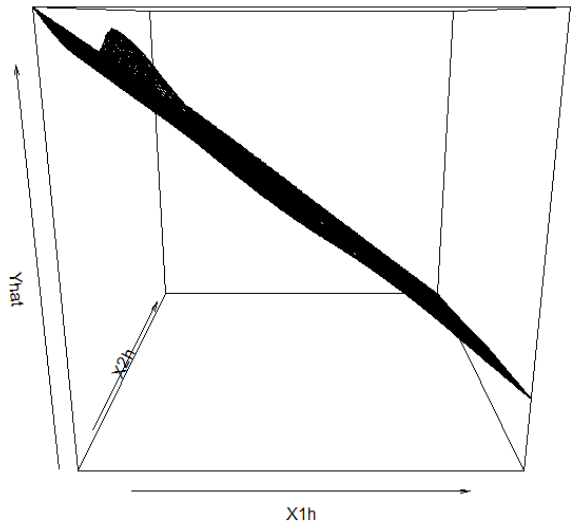
Response Variable: Y = Mean January Temp (F) at n = 369 Texas weather stations

Predictors: X_1 = Latitude (degrees North Latitude), X_2 = Elevation (Feet above sea level)

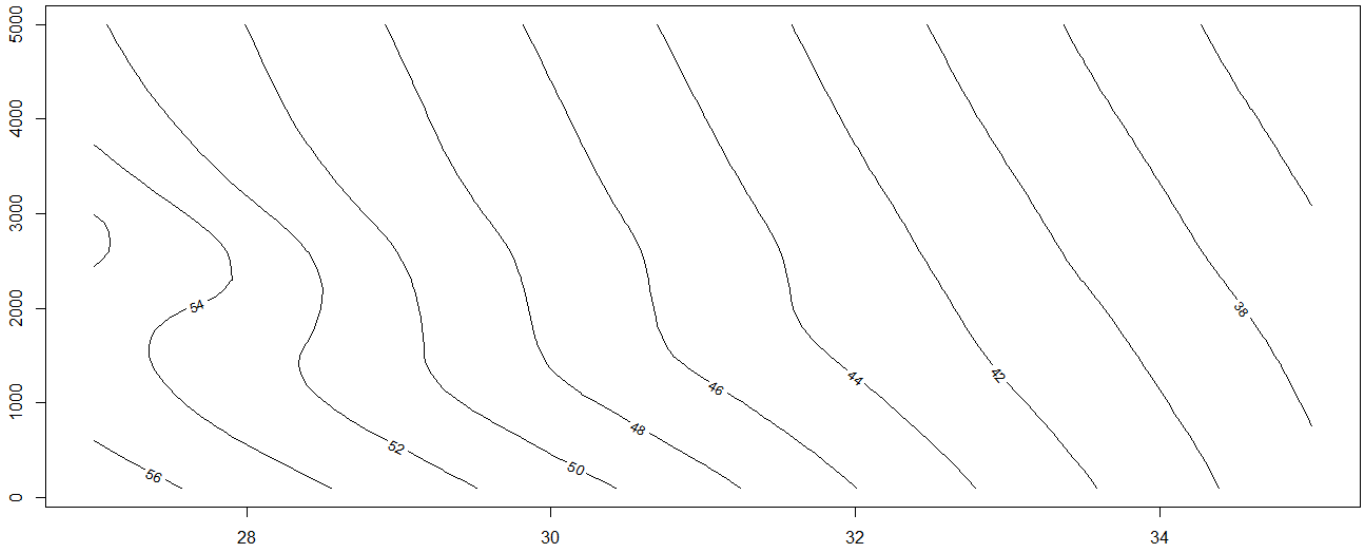
ANOVA						
	df	*SS*	*MS*	*F*	*gnificance F*	
Regressio	2	13155.37	6577.68	3906.83	0.0000	
Residual	366	616.21	1.68			
Total	368	13771.58				
	Coefficients	*andard Err*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	118.2876	1.035998	114.18	0.0000	116.2504	120.3249
ELEV	-0.00106	5.83E-05	-18.10	0.0000	-0.00117	-0.00094
LAT	-2.26583	0.034246	-66.16	0.0000	-2.33318	-2.19849

```
txnorm <- read.csv("E:\\blue_drive\\sta4210\\TXnorm1.csv",header=TRUE)
attach(txnorm); names(txnorm)
X1 <- LAT; X2 <- ELEV; Y <- Mean1;

#### Generate Lowess Curve
n <- length(Y)
qn <- floor(n/2)
sd1 <- sd(X1);   sd2 <- sd(X2)
X <- as.matrix(cbind(rep(1,n),X1,X2))
Y <- as.matrix(Y)
x1.lo <- 27; x1.hi <- 35; x1.step <- 0.10
x2.lo <- 100; x2.hi <- 5000; x2.step <- 100

Yhat <- matrix(rep(0,
length(seq(x1.lo,x1.hi,x1.step))*length(seq(x2.lo,x2.hi,x2.step))),
                ncol=length(seq(x2.lo,x2.hi,x2.step)))
rownum <- 0
for (X1h in seq(x1.lo,x1.hi,x1.step)) {
rownum <- rownum + 1
colnum <- 0
for (X2h in seq(x2.lo,x2.hi,x2.step)) {
colnum <- colnum + 1
X_h <- as.matrix(cbind(1,X1h,X2h))
d <- sqrt(((X1-X1h)/sd1)^2 + ((X2-X2h)/sd2)^2)
dq <- d[rank(d,ties.method="random") == qn]
w <- rep(0,n)
for (i in 1:n) {
if (d[i] < dq) w[i] <- (1-(d[i]/dq)^3)^3
}
w <- diag(w)
Yhat[rownum,colnum] <- X_h %*%solve(t(X) %*% w %*% X) %*%
                        (t(X) %*% w %*% Y)
}}

X1h <- seq(x1.lo,x1.hi,x1.step);   X2h <- seq(x2.lo,x2.hi,x2.step)
persp(X1h,X2h,Yhat,xlim=c(x1.lo,x1.hi),ylim=c(x2.lo,x2.hi),main="Lowess 3d
plot")
contour(X1h,X2h,Yhat,xlim=c(x1.lo,x1.hi),ylim=c(x2.lo,x2.hi),
 main="Lowess contour plot",labcex=0.8)
```

Lowess 3d plot



Lowess contour plot



Regression Trees

- Splits region covered by the predictor variables into increasing number of sub-regions, where predicted value is mean response of all cases falling in the sub-region.
- Goal: Choose "cut-points" that minimize Error Sum of Squares for the current number of sub-regions – computationally intensive, no closed form solution
- Problem: SSE will continually drop until there are as many sub-regions as distinct combinations of predictors (SSE → 0)
- Validation samples can be used to determine which number of nodes (sub-regions – 1) that minimizes Mean Square Prediction Error (MSPR)

Example: Values of Minerals Extracted by State and Land Area

Source: http://minerals.er.usgs.gov/minerals/pubs/commodity/statistical_summary/index.html#myb
(retrieved 6/23/2014).

Non-Fuel mineral production ($10M) and land area ($1000m^2$) for the 50 United States in 2011.

Y = ln(Value), X=ln(Area)

Procedure for obtaining Regression Tree:

- Sort the states by X = ln(Area) in ascending order
- Compute the corrected sum of squares for all values of Y in the states from the lowest, up to the current, label this as RSS_1
- Compute the corrected sum of squares for all values of Y in states above the current state, in terms of Y values, label this as RSS_2
- Compute the sum of squares for this partition: $RSS_p = RSS_1 + RSS_2$
- Choose the partition that minimizes RSS_p. Software will choose the "cut-point" as half way between the best current state and the next highest one.
- Once the first partition is made, then we try all sub-partitions within the first 2 partitions, and repeat the process.

In the following output, for the column labelled RSSp(1), in row 2, for instance, we have the statement:

=devsq($C$2:c2) + devsq(c3:$C$51)

This has the effect of creating a partition with only Rhode Island in one part, all other states in the other. This command can be copied all the way down, where c2 and c3 keep changing for each row, while $C$2 and $C$51 stay the same. We do not copy this to the last row (Alaska). **Keep in mind, the groups are being "defined" by their *X* levels (which were used to sort dataset), but the sums of squares are based on the *Y* levels.**

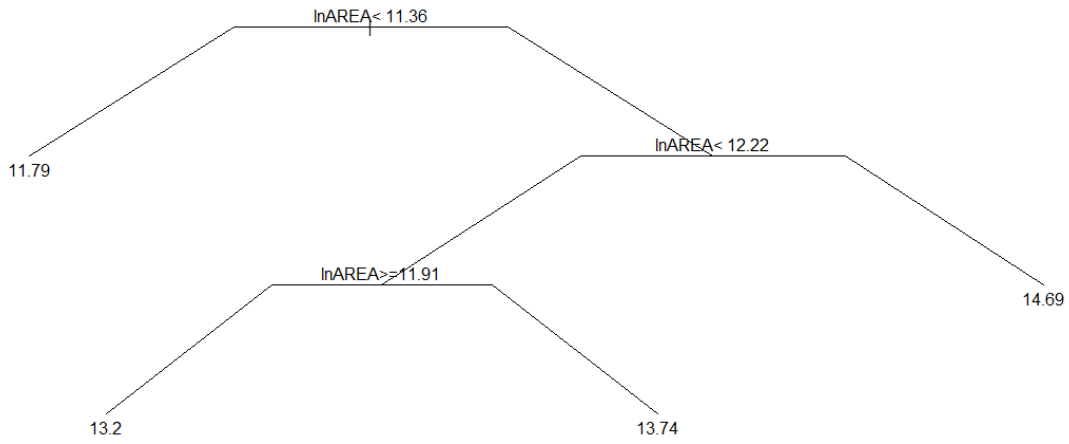
State	lnAREA	lnValue	Cell	RSSp(1)	RSSp(2)
Rhode Island	7.9047	10.6502	C2	82.8713	45.5076
Delaware	8.5291	9.3237	C3	65.7620	39.0947
Connecticut	9.4335	11.9576	C4	65.5661	41.6400
Hawaii	9.7172	11.5229	C5	61.5482	41.7367
New Jersey	9.8627	12.5245	C6	62.3662	43.8637
Massachusetts	9.9184	12.3239	C7	61.4442	44.9121
New Hampshire	10.0519	11.5099	C8	56.3334	44.3457
Vermont	10.0858	11.6784	C9	51.7930	43.8586
Maryland	10.1386	12.5879	C10	51.3392	45.0293
West Virginia	11.0413	12.6885	C11	51.0187	46.1045
South Carolina	11.2645	13.0878	C12	52.1337	46.9214
Maine	**11.2885**	**11.6784**	**C13**	**46.9361**	#N/A
Indiana	11.4393	13.5437	C14	49.5547	46.7303
Kentucky	11.5425	13.5811	C15	51.9056	46.5476
Virginia	11.5425	13.9895	C16	55.4264	46.6686
Ohio	11.5712	13.7768	C17	57.7683	46.6136
Tennessee	11.5806	13.6854	C18	59.5190	46.4973
Louisiana	11.6351	13.0498	C19	58.9719	45.8691
Pennsylvania	11.6613	14.2855	C20	62.2972	46.2161
Mississippi	11.7035	12.1808	C21	58.7557	44.6035
New York	11.7118	14.1082	C22	61.2762	44.9198
North Carolina	11.7440	13.6447	C23	62.1683	44.6902
Alabama	11.7830	13.7747	C24	63.3341	44.5872
Arkansas	11.8130	13.5785	C25	63.8054	44.2485
Florida	11.8494	15.0481	C26	68.3129	45.3863
Wisconsin	11.8565	13.4343	C27	68.1961	44.9432
Illinois	11.8776	13.8832	C28	69.2093	44.8938
Iowa	11.8845	13.3893	C29	68.8329	44.3356
Michigan	11.8982	14.6951	C30	71.7911	45.0547
Georgia	11.9184	14.1871	C31	73.3009	45.2401
Washington	12.0552	13.5171	C32	73.1020	44.8209
Missouri	12.0895	14.6040	C33	75.4560	45.3551
Oklahoma	12.0895	13.3179	C34	74.6917	44.7388
North Dakota	12.0951	11.7361	C35	69.6187	41.9461
South Dakota	12.1910	12.6508	C36	66.4235	39.6392
Nebraska	**12.2011**	**12.3800**	**C37**	**61.8331**	**36.1836**
Minnesota	12.2356	15.3174	C38	66.6847	38.6376
Kansas	12.2643	13.9288	C39	66.8878	38.1631
Utah	12.2690	15.2741	C40	71.2419	40.2954
Idaho	12.2737	14.0931	C41	71.8085	40.0899
Oregon	12.4252	12.6281	C42	67.3019	36.5015
Wyoming	12.4332	14.5763	C43	69.1683	37.1025
Colorado	12.5025	14.4730	C44	70.5217	37.3491
Nevada	12.5567	16.1181	C45	77.4732	41.4345
Arizona	12.5913	15.9426	C46	83.0643	44.3959
New Mexico	12.6571	14.0387	C47	83.0775	43.9414
Montana	12.8400	14.1802	C48	83.2282	43.4918
California	12.9092	14.9818	C49	85.7862	44.6207
Texas	13.4269	14.9241	C50	88.0399	45.5498
Alaska	14.2076	15.1531	C51	#N/A	#N/A

For the first partition, RSSp(1) is minimized when all states up to, and including Maine are in one group, and all other states are in the other group. The *X*-level for the cut-point would be half way between Maine's and Indiana's levels:
$(11.29 + 11.44)/2 \approx 11.36$

For the second partition, we first try all sub-partitions within Rhode Island to Maine, keeping the other partition (Indiana to Alaska) intact. Then we try keeping Rhode Island to Maine intact, and try all sub-partitions from Indiana to Alaska. RSSp(2) is minimized when we have three groups: Rhode Island to Maine, Indiana to Nebraska, and Minnesota to Alaska. The second cut-point is half way between Nebraska's and Minnesota's levels:
$(12.20 + 12.24)/2 = 12.22$

R Program and Output (Makes use of the rpart package)

```
stmin <- read.csv("E:\\blue_drive\\sta4210\\state_mineral.csv",header=TRUE)
attach(stmin); names(stmin)

library(rpart)
(mintree1 <- rpart(lnValue ~ lnAREA,stmin,method="anova",cp=.0001))
plot(mintree1,margin=.10)
text(mintree1)

plot(mintree1,compress=T,uniform=T,branch=0.4,margin=.10)
text(mintree1)

printcp(mintree1)

########### Output #############

Error in printcp(mintree1) : object 'mintree1' not found
> (mintree1 <- rpart(lnValue ~ lnAREA,stmin,method="anova",cp=.0001))
n= 50

node), split, n, deviance, yval
      * denotes terminal node

 1) root 50 90.951040 13.46410
   2) lnAREA< 11.3639 12 11.537110 11.79448 *
   3) lnAREA>=11.3639 38 35.398650 13.99134
     6) lnAREA< 12.21835 24 14.017400 13.58507
      12) lnAREA>=11.9083 7  6.177052 13.19900 *
      13) lnAREA< 11.9083 17  6.367372 13.74404 *
     7) lnAREA>=12.21835 14 10.628880 14.68781 *
```



Note that R has further made one more split, the last cut-point being 11.91. The numbers at the bottom of each node is the mean of *Y* for that group.

Example: LPGA Prize Winnings and Performance Statistics (2008)

Source: www.lpga.com

Response Variable: Y = ln(Prize Winnings/Round)

Predictors: X_1 = Average Driving Distance, X_2 = Percent Fairways Reached, X_3 = Percent Greens Reached in Regulation, X_4 = Average Putts per Round, X_5 = Percent of Time getting par or better when in sand trap

We fit a Regression Tree, with the default levels for the **rpart** package. It uses cross-validation to obtain a relative error for hold-out samples.

```
lpga <- read.csv("E:\\blue_drive\\sta4210\\lpga2008.csv",header=TRUE)
attach(lpga); names(lpga)

install.packages("rpart")
library(rpart)

(lpgatree1 <- rpart(lnprz ~dist+fairway+green+putts+sandsv,
lpga,method="anova"))
plot(lpgatree1,compress=T,uniform=T,branch=0.4,margin=.10)
text(lpgatree1)
printcp(lpgatree1)

lpgatree2 <- prune.rpart(lpgatree1,.010561)
plot(lpgatree2,compress=T,uniform=T,branch=0.4,margin=.10)
text(lpgatree2)
1-sum(residuals(lpgatree2)^2)/sum((lnprz-mean(lnprz))^2)

lpgatree3 <- prune.rpart(lpgatree1,.027677)
plot(lpgatree3,compress=T,uniform=T,branch=0.4,margin=.10)
text(lpgatree3)
1-sum(residuals(lpgatree3)^2)/sum((lnprz-mean(lnprz))^2)

lpgaols1 <- lm(lnprz ~dist+fairway+green+putts+sandsv,lpga)
summary(lpgaols1)

######  Output #########
node), split, n, deviance, yval
      * denotes terminal node

 1) root 157 168.8622000 7.983957
   2) putts>=29.465 63  34.9813300 7.151130
     4) green< 63.55 37  17.7577400 6.786422
       8) fairway< 63.4 10   7.5963210 6.202440 *
       9) fairway>=63.4 27   5.4879870 7.002711
        18) dist< 238 12   1.9975940 6.680450 *
        19) dist>=238 15   1.2471850 7.260520 *
     5) green>=63.55 26   5.2985240 7.670138
      10) putts>=30.49 11   1.0050850 7.339045 *
      11) putts< 30.49 15   2.2033020 7.912940 *
   3) putts< 29.465 94  60.8977500 8.542129
     6) green< 64.1 57  23.0836200 8.116114
      12) green< 59.6 12   0.5995004 7.419792 *
      13) green>=59.6 45  15.1141700 8.301800
        26) putts>=28.195 30   5.7225080 8.014120
          52) sandsv< 44.5 22   2.1488480 7.865255 *
          53) sandsv>=44.5 8   1.7453830 8.423500 *
        27) putts< 28.195 15   1.9432840 8.877160 *
     7) green>=64.1 37  11.5326300 9.198422
      14) green< 67.5 23   4.0944720 8.897313
        28) putts>=28.82 7   1.4778570 8.476357 *
        29) putts< 28.82 16   0.8335013 9.081481 *
      15) green>=67.5 14   1.9269410 9.693100 *
```

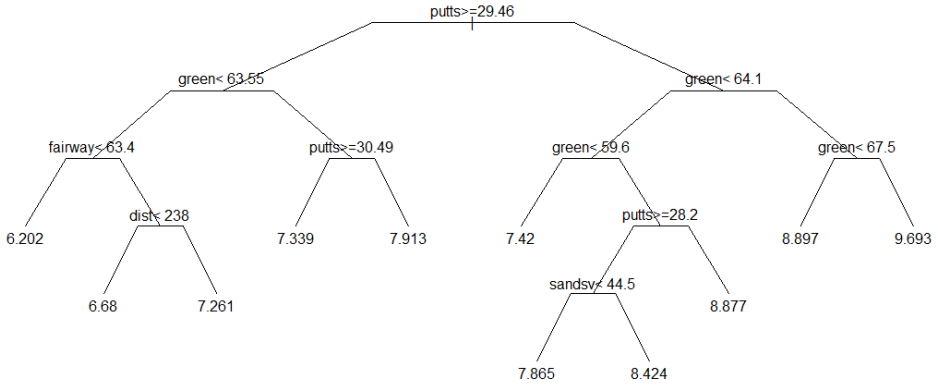
Output Continued

```
Regression tree:
rpart(formula = lnprz ~ dist + fairway + green + putts + sandsv,
    data = lpga, method = "anova")

Variables actually used in tree construction:
[1] dist    fairway green   putts   sandsv

Root node error: 168.86/157 = 1.0756

n= 157

         CP nsplit rel error  xerror     xstd
1  0.432205      0   1.00000 1.01189 0.123128
2  0.155639      1   0.56779 0.67259 0.087811
3  0.070620      2   0.41216 0.53029 0.081770
4  0.043877      3   0.34154 0.51774 0.068584
5  0.032637      5   0.25378 0.48989 0.068215
6  0.027676      6   0.22114 0.40569 0.063988
7  0.013284      7   0.19347 0.40801 0.069041
8  0.012378      8   0.18018 0.37876 0.064803
9  0.010827      9   0.16781 0.37476 0.064619
10 0.010560     10   0.15698 0.35898 0.057372
11 0.010000     11   0.14642 0.35898 0.057372

> 1-sum(residuals(lpgatree2)^2)/sum((lnprz-mean(lnprz))^2)
[1] 0.8430204

> 1-sum(residuals(lpgatree3)^2)/sum((lnprz-mean(lnprz))^2)
[1] 0.7788553

> summary(lpgaols1)

Call: lm(formula = lnprz ~ dist + fairway + green + putts + sandsv, data = lpga)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 15.811996   1.667791   9.481  < 2e-16 ***
dist        -0.004090   0.005376  -0.761  0.44797
fairway     -0.006942   0.007976  -0.870  0.38551
green        0.183908   0.012069  15.238  < 2e-16 ***
putts       -0.629492   0.031931 -19.714  < 2e-16 ***
sandsv       0.011987   0.003964   3.024  0.00293 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3937 on 151 degrees of freedom
Multiple R-squared:  0.8614,    Adjusted R-squared:  0.8568
F-statistic: 187.6 on 5 and 151 DF,  p-value: < 2.2e-16
```

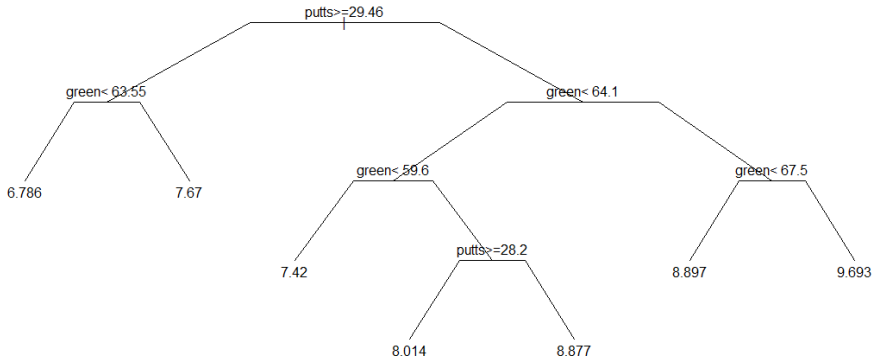
Comments: Note that different runs of the program will give different cross-validation errors, as the samples will change from run to run.

- Based on the relative cross-validation error (divided by the base tree (no splits)), which is denoted **xerror**, the best model is the model with the minimum value, which is the model with 10 splits (cp = .010560). When we “prune” the tree to keep only these splits, that is model lpgatree2, it has R^2 = .843. With 10 splits, this model can be thought of as having 11 parameters.
- If we choose a simpler tree, that is, the one with the lowest xerror that is within one standard error of the minimum (0.35898 + 0.057372 = 0.416352), is the model with 6 splits (cp = .027676). When we “prune” the tree to keep only these splits, that is model lpgatree3, it has R^2 = .779. With 6 splits, this model can be thought of as having 7 parameters.
- The linear regression model, with 6 parameters, has R^2 = .861.

Lpgatree2



Lpgatree3



Comments:

- The first split is at **putts** per round equal to **29.46**.
  - For putts > 29.46, there are 63 golfers with mean lnprize = 7.15
  - For putts < 29.46, there are 94 golfers with mean lnprize = 8.54
- The second split is **within golfers with putts > 29.46**, and it is **greens** in regulation equal to **63.5**
  - For green < 63.5, there are 37 golfers with mean lnprize = 6.79
  - For green > 63.5, there are 26 golfers with mean lnprize = 7.67
- The third split is **within golfers with putts < 29.46**, and it is **green** in regulation equal to **64.1**
  - For green < 64.1, there are 57 golfers with mean lnprize = 8.12
  - For green > 64.1, there are 37 golfers with mean lnprize = 9.20
- Eventually each golfer ends up in one of the final nodes

Bootstrapping to Estimate Precision

- Computationally intensive methods used to estimate precision of estimators in non-standard situations
- Based on re-sampling (with replacement) from observed samples, re-computing estimates repeatedly (nonparametric). Parametric methods also exist (not covered here)
- Once original sample is observed, and quantity of interest estimated, many samples of size *n* (selected with replacement) are obtained, each sample providing a new estimate.
- The standard deviation of the new sample quantities represents an estimate of the standard error.
- Confidence Intervals can also be obtained by selecting the extreme (say 2.5^{th} and 97.5^{th}) percentiles of the sample quantities.

Two Basic Approaches

- **Fixed *X* Sampling** (Model is good fit, constant variance, predictor variables are fixed, i.e. controlled experiment)
  - Fit the regression, obtain all fitted values and residuals
  - Keeping the corresponding *X*-level(s) and fitted values, re-sample the *n* residuals (with replacement)
  - Add the bootstrapped residuals to the fitted values, and re-fit the regression (repeat process many times)
- **Random *X* Sampling** (Not sure of adequacy of model: fit, non-constant variance, random predictor variables)
  - After fitting regression, and estimating quantities of interest, sample *n* cases (with replacement) and re-estimate quantities of interest with "new" datasets (repeat many times)

Bootstrap Confidence Intervals – Reflection Method

Suppose we are interested in Estimating β_1 in simple regression (generalizes to other parameters):

1) Fit regression on original dataset, obtain $b_1 = \dfrac{\sum_{i=1}^{n}\left(X_i-\overline{X}\right)\left(Y_i-\overline{Y}\right)}{\sum_{i=1}^{n}\left(X_i-\overline{X}\right)^2}$

2) Obtain *m* bootstrap samples and their estimated slopes: $b_{j1}^{*} \quad j=1,\ldots,m$

3) Order the b_{j1}^{*} from smallest to largest, and obtain $b_1^{*}\left(\alpha/2\right)$ and $b_1^{*}\left(1-\left(\alpha/2\right)\right)$
(the lower and upper percentiles of the distribution)

4) Compute $d_1 = b_1 - b_1^{*}\left(\alpha/2\right) \qquad d_2 = b_1^{*}\left(1-\left(\alpha/2\right)\right) - b_1$

5) Approximate $\left(1\text{-}\alpha\right)100\%$ CI for β_1: $\quad b_1 - d_2 \le \beta_1 \le b_1 + d_1$

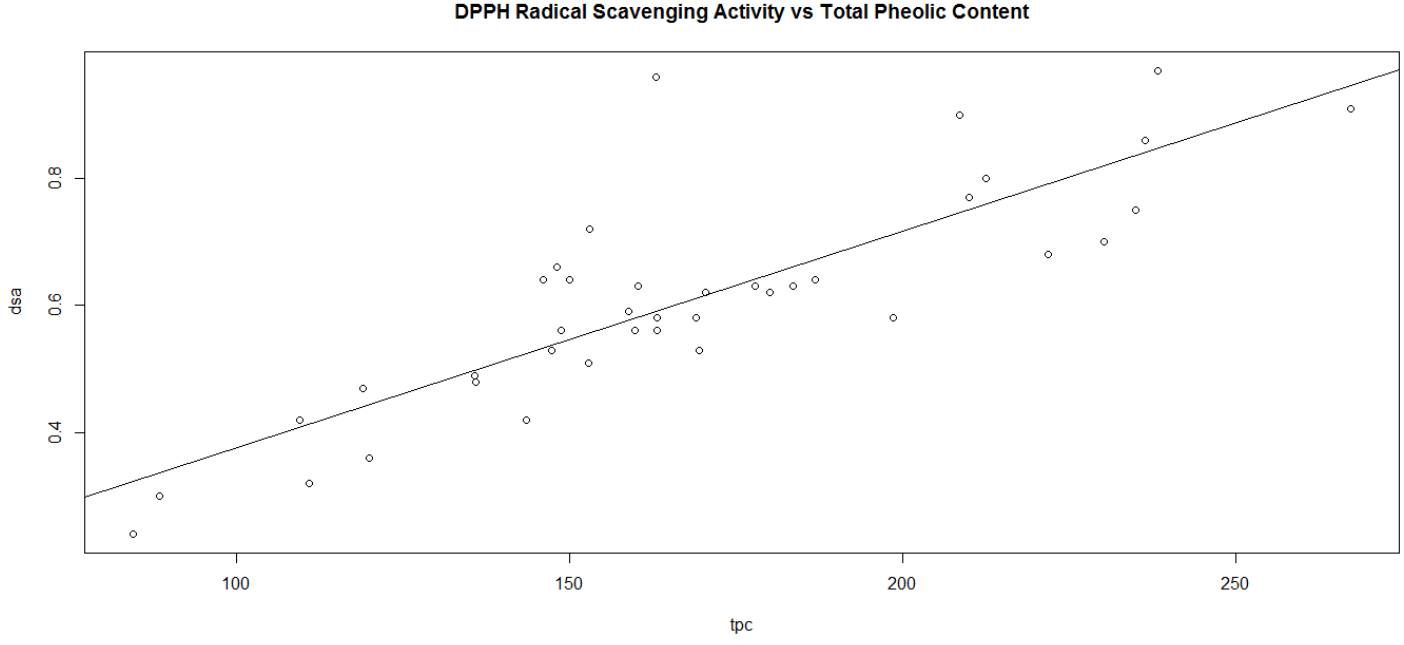
Example: Antioxidant Levels and Activity in a Sample of n=40 Lager Beers

Source: H. Zhao, H. Li, G. Sun, B.Yang, M.Zhao (2013). "Assessment of endogenous antioxidative compounds and antioxidant activities of lager beers," *Journal of the Science of Food and Agriculture*, Vol. 93, pp. 910-917.

Response Variable: Y = DPPH Radical Scavenging Activity (mmol TE/L), labelled **dsa** in dataset
Predictor Variable: X = Total Phenolic Content (mg GAE/L), labelled **tpc** in dataset

Regression Model:

```
lager <- read.csv("E:\\blue_drive\\data_articles\\lager_antioxidant_reg.csv",header=TRUE)
attach(lager); names(lager)

lager.mod1 <- lm(dsa ~ tpc)
summary(lager.mod1)
confint(lager.mod1)
yhat <- predict(lager.mod1)
e <- residuals(lager.mod1)
b1 <- coef(lager.mod1)[2]
plot(tpc,dsa,main="DPPH Radical Scavenging Activity vs Total Pheolic Content")
abline(lager.mod1)

########## Output #################
> summary(lager.mod1)
Call: lm(formula = dsa ~ tpc)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.0343018  0.0639781   0.536    0.595
tpc         0.0034132  0.0003694   9.240 2.93e-11 ***

Residual standard error: 0.09629 on 38 degrees of freedom
Multiple R-squared:  0.692,     Adjusted R-squared:  0.6839
F-statistic: 85.38 on 1 and 38 DF,  p-value: 2.926e-11
> confint(lager.mod1)
                  2.5 %      97.5 %
(Intercept) -0.09521518 0.163818791
tpc          0.00266544 0.004160979
```



Bootstrap Method 1: Fixed *X* levels. Keep all 40 fitted (predicted values), and add sampled (with replacement) residuals. Save all regression slopes (b_1). This makes use of the matrix approach (quicker than using lm). Note, this program is a continuation of the previous one.

```
n <- length(dsa)
X <- as.matrix(cbind(rep(1,n),tpc))

set.seed(13579)
num.boot <- 10000
b1.boot <- rep(0,num.boot)

for (i in 1:num.boot) {
e.boot <- as.matrix(sample(e,size=n,replace=TRUE))
Y.boot <- yhat + e.boot

b.boot <- solve(t(X) %*% X) %*% t(X) %*% Y.boot
b1.boot[i] <- b.boot[2,1]
}

hist(b1.boot, breaks=24)

(b1.boot_025 <- quantile(b1.boot,.025))
(b1.boot_975 <- quantile(b1.boot,.975))

(b1.boot.sd <- sd(b1.boot))

(d1 <- b1-b1.boot_025)
(d2 <- b1.boot_975-b1)

(beta1.95CI <- c(b1-d2,b1+d1))

################ Output  #######
> (b1.boot_025 <- quantile(b1.boot,.025))
       2.5%
0.002700025
> (b1.boot_975 <- quantile(b1.boot,.975))
      97.5%
0.004141108
>
> (b1.boot.sd <- sd(b1.boot))
[1] 0.0003603104
>
> (d1 <- b1-b1.boot_025)
          tpc
0.0007131838
> (d2 <- b1.boot_975-b1)
       97.5%
0.0007278994
>
> (beta1.95CI <- c(b1-d2,b1+d1))
        tpc         tpc
0.002685310 0.004126393
```

Comments:

- The **sample** command selects 40 residuals (with replacement) from the original regression, and adds them to the fitted values from the first regression for a new set of *Y* values. The *X* values remain the same.
- The original 95% Confidence Interval (from the regression analysis) for β_1 is (0.002665,0.004161)
- The middle 95% of the bootstrap estimates of β_1 fall in the range (0.002700 , 0.004141)
- The 95% Confidence Interval (from the reflection method) for β_1 is (0.002685 , 0.004126)
- The results by all methods are very similar

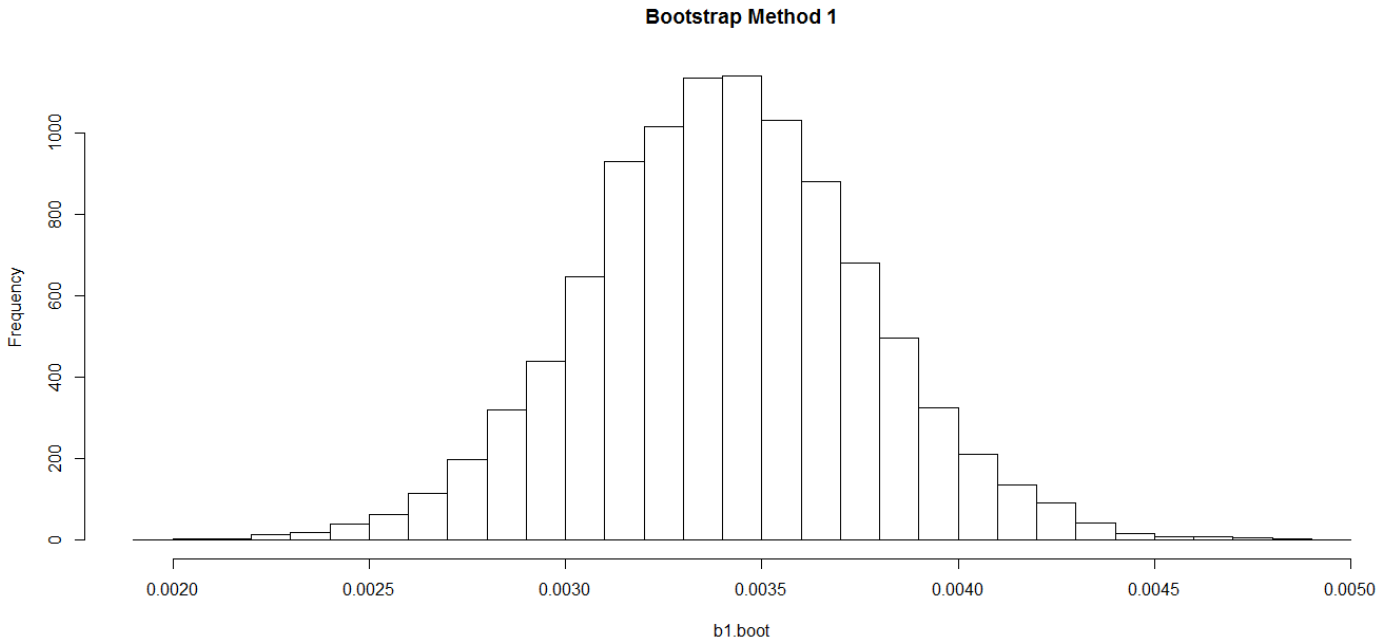
Bootstrap Method 2: Random X levels. Sample 40 pairs (X,Y) from the original sample, with replacement. Fit the model for each new sample. Save all regression slopes (b_1). This makes use of the matrix approach (quicker than using lm). Note, this program is a continuation of the previous one.

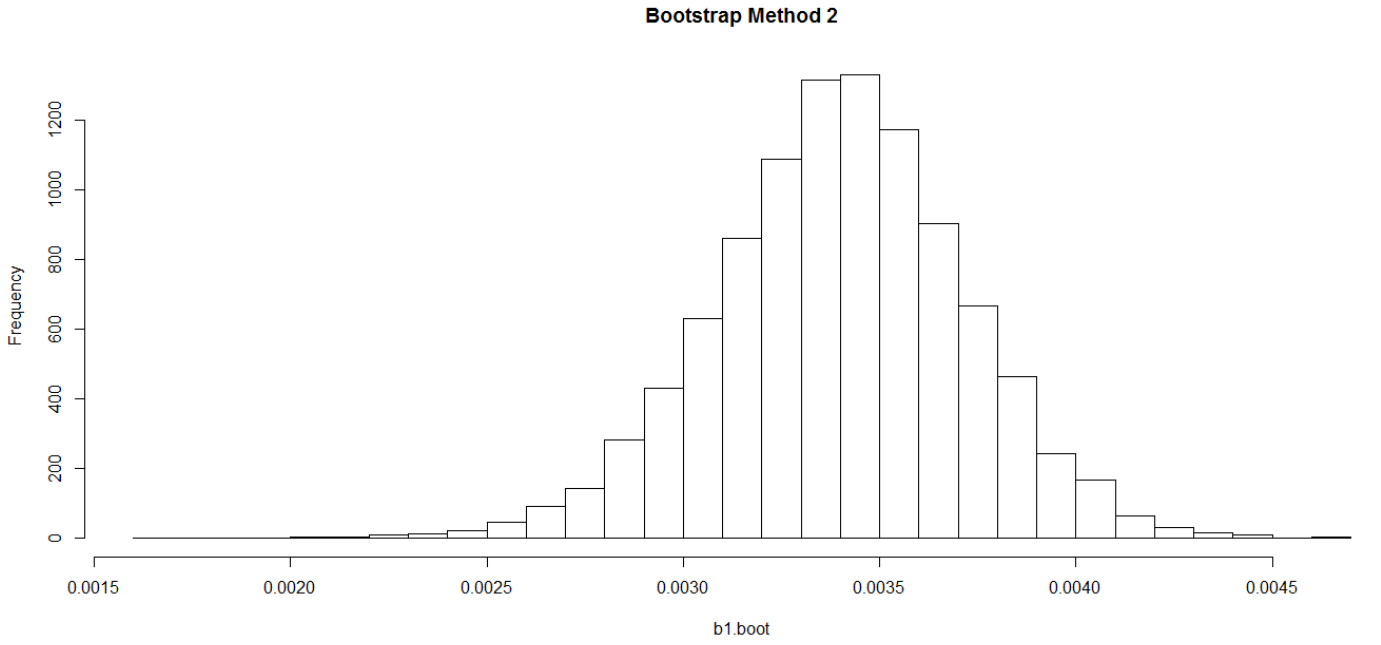
```
###### Bootstrap by selecting n (X,Y) pairs with replacement

num.boot <- 10000
set.seed(34567)
b1.boot <- rep(0,num.boot)

for (i in 1:num.boot)  {
boot.sample <- sample(1:n,size=n,replace=TRUE)
dsa.b <- dsa[boot.sample]
tpc.b <- tpc[boot.sample]
X.boot <- as.matrix(cbind(rep(1,n),tpc.b))
Y.boot <- dsa.b
b.boot <- solve(t(X.boot) %*% X.boot) %*% t(X.boot) %*% Y.boot
b1.boot[i] <- b.boot[2,1]
}

hist(b1.boot, breaks=24)

(b1.boot_025 <- quantile(b1.boot,.025))
(b1.boot_975 <- quantile(b1.boot,.975))

(b1.boot.sd <- sd(b1.boot))
(d1 <- b1-b1.boot_025)
(d2 <- b1.boot_975-b1)
(beta1.95CI <- c(b1-d2,b1+d1))

###### Output #######
> (b1.boot_025 <- quantile(b1.boot,.025))
      2.5%
0.002754063
> (b1.boot_975 <- quantile(b1.boot,.975))
     97.5%
0.004018062
>
> (b1.boot.sd <- sd(b1.boot))
[1] 0.0003181617
>
> (d1 <- b1-b1.boot_025)
         tpc
0.0006591461
> (d2 <- b1.boot_975-b1)
      97.5%
0.0006048525
>
> (beta1.95CI <- c(b1-d2,b1+d1))
        tpc         tpc
0.002808357 0.004072355
```

Comments:

- The **sample** command selects 40 integers between 1 and 40 (with replacement) and then the sample X and Y values are those pairs. Some observations (beers, in this case) will be observed multiple times in a given sample, others will not be observed in the given sample.
- The original 95% Confidence Interval (from the regression analysis) for β_1 is (0.002665,0.004161)
- The middle 95% of the bootstrap estimates of β_1 fall in the range (0.002754 , 0.004018)
- The 95% Confidence Interval (from the reflection method) for β_1 is (0.002808 , 0.004072)

Histograms of the Bootstrap Samples:





Chapter 12 - Autocorrelation in Time Series

Time series data refer to the process of observing data on common units over time. For instance, we may consider U.S. oil consumption over a series of years, as well as population and other economic variables.

Issues in Autocorrelated Data

- When error terms are correlated (not independent), problems occur when using ordinary least squares (OLS) estimates
  - Regression Coefficients are Unbiased, but not Minimum Variance
  - MSE underestimates s^2
  - Standard errors of regression coefficients based on OLS underestimate the true standard error
  - $\Rightarrow$ Inflated t and F statistics and artificially narrow confidence intervals

Autocorrelated Errors (1st Order) $\Rightarrow \quad \varepsilon_t = \rho\varepsilon_{t-1} + u_t$

where $u_t \equiv$ uncorrelated disturbances (typically assumed to be normal)

First-Order Model

First-Order Autoregressive Model ($AR(1)$):

Simple Regression: $Y_t = \beta_0 + \beta_1 X_t + \varepsilon_t \quad t = 1,...,n \qquad \varepsilon_t = \rho\varepsilon_{t-1} + u_t$

$\rho \equiv$ autoregression parameter with $|\rho| < 1$

$u_t \sim N\left(0, \sigma^2\right)$ and independent

Generalizes to Multiple Regression:

$Y_t = \beta_0 + \beta_1 X_{t1} + ... + \beta_{p-1} X_{t,p-1} + \varepsilon_t \quad t = 1,...,n \qquad \varepsilon_t = \rho\varepsilon_{t-1} + u_t$

Properties of Errors (assumption regarding ε_1 for model consistency):

$$\varepsilon_1 \sim N\left(0, \frac{\sigma^2}{1-\rho^2}\right)$$

$$\varepsilon_2 = \rho\varepsilon_1 + u_2 \quad \Rightarrow \quad E\{\varepsilon_2\} = \rho E\{\varepsilon_1\} + E\{u_2\} = 0 \qquad \sigma^2\{\varepsilon_2\} = \rho^2\sigma^2\{\varepsilon_1\} + \sigma^2\{u_2\} = \rho^2\left(\frac{\sigma^2}{1-\rho^2}\right) + \sigma^2 = \frac{\sigma^2}{1-\rho^2}$$

Covariance: $\sigma\{\varepsilon_2, \varepsilon_1\} = \sigma\{\rho\varepsilon_1 + u_2, \varepsilon_1\} = \rho\sigma^2\{\varepsilon_2\} + \sigma\{u_2, \varepsilon_1\} = \rho\sigma^2\{\varepsilon_2\} + 0 = \frac{\rho\sigma^2}{1-\rho^2}$

Corrrelation: $\rho\{\varepsilon_2, \varepsilon_1\} = \frac{\sigma\{\varepsilon_2, \varepsilon_1\}}{\sigma\{\varepsilon_2\}\sigma\{\varepsilon_1\}} = \frac{\frac{\rho\sigma^2}{1-\rho^2}}{\sqrt{\frac{\sigma^2}{1-\rho^2}}\sqrt{\frac{\sigma^2}{1-\rho^2}}} = \rho$

In General:

$$\varepsilon_t = \rho\varepsilon_{t-1} + u_t = \rho\left(\rho\varepsilon_{t-2} + u_{t-1}\right) + u_t = \rho^2\varepsilon_{t-2} + \rho u_{t-1} + u_t = \ldots = \sum_{s=0}^{\infty}\rho^s u_{t-s}$$

$$E\left\{\varepsilon_t\right\} = 0 \qquad \sigma^2\left\{\varepsilon_t\right\} = \sigma^2\left\{\sum_{s=0}^{\infty}\rho^s u_{t-s}\right\} = \sum_{s=0}^{\infty}\rho^{2s}\sigma^2\left\{u_{t-s}\right\} = \sigma^2\sum_{s=0}^{\infty}\rho^{2s} = \frac{\sigma^2}{1-\rho^2}$$

Covariance: $\sigma\left\{\varepsilon_t, \varepsilon_{t-s}\right\} = \dfrac{\rho^s\sigma^2}{1-\rho^2} \quad s \geq 0$

Correlation: $\rho\left\{\varepsilon_t, \varepsilon_{t-s}\right\} = \rho^s \quad s \geq 0$

$$\boldsymbol{\sigma}^2\left\{\boldsymbol{\varepsilon}\right\} = \frac{\sigma^2}{1-\rho^2}\begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{n-1} \\ \rho & 1 & \rho & \cdots & \rho^{n-2} \\ \rho^2 & \rho & 1 & \cdots & \rho^{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{n-1} & \rho^{n-2} & \rho^{n-3} & \cdots & 1 \end{bmatrix}$$

$AR(2)$: $\varepsilon_t = \rho_1\varepsilon_{t-1} + \rho_2\varepsilon_{t-2} + u_t$ Even Higher or models can be fit as well.

Test For Independence - Durbin-Watson Test

$Y_t = \beta_0 + \beta_1 X_{t1} + \ldots + \beta_{p-1}X_{t,p-1} + \varepsilon_t \quad \varepsilon_t = \rho\varepsilon_{t-1} + u_t \quad u_t \sim NID\left(0, \sigma^2\right) \quad |\rho| < 1$

$H_0 : \rho = 0 \quad \Rightarrow$ Errors are uncorrelated over time

$H_A : \rho > 0 \quad \Rightarrow$ Positively correlated

1) Obtain Residuals from Regression

2) Compute Durbin-Watson Statistic (given below)

3) Obtain Critical Values from Table B.7, pp. 1330-1331 (R will provide a p-value)

If $DW < d_L\left(p-1, n\right) \Rightarrow$ Reject H_0 If $DW > d_U\left(p-1, n\right) \Rightarrow$ Conclude H_0 Otherwise Inconclusive

Note: R will produce a bootstrap based *P*-value, to avoid the "Inconclusive" Outcome

Test Statistic: $DW = \dfrac{\sum_{t=2}^{n}\left(e_t - e_{t-1}\right)^2}{\sum_{t=1}^{n}e_t^2}$

$E\left\{\varepsilon_t\right\} = 0 \quad E\left\{\varepsilon_t\varepsilon_{t-1}\right\} = \dfrac{\rho\sigma^2}{1-\rho^2}$

$$\Rightarrow \quad \sum_{t=2}^{n}\left(e_t - e_{t-1}\right)^2 = \sum_{t=2}^{n}e_t^2 + \sum_{t=2}^{n}e_{t-1}^2 - 2\sum_{t=2}^{n}e_t e_{t-1} \approx 2\sum_{t=1}^{n}e_t^2 - 2n\frac{\rho\sigma^2}{1-\rho^2}$$

$\Rightarrow$ Under H_0, expect DW ≈ 2

Autocorrelation - Remedial Measures

- Determine whether a missing predictor variable can explain the autocorrelation in the errors
- Include a linear (trend) term if the residuals show a consistent increasing or decreasing pattern
- Include seasonal dummy variables if data are quarterly or monthly and residuals show cyclic behavior
- Use transformed Variables that remove the (estimated) autocorrelation parameter (Cochrane-Orcutt and Hildreth-Lu Procedures)
- Use First Differences
- Estimated Generalized Least Squares (Uses the estimated Variance-Covariance matrix, similar to Weighted Least Squares)

Transformed Variables

Suppose ρ is known: $Y_t = \beta_0 + \beta_1 X_t + \varepsilon_t \qquad \varepsilon_t = \rho\varepsilon_{t-1} + u_t$

Let $Y_t' = Y_t - \rho Y_{t-1} = \left(\beta_0 + \beta_1 X_t + \varepsilon_t\right) - \rho\left(\beta_0 + \beta_1 X_{t-1} + \varepsilon_{t-1}\right) =$

$\beta_0\left(1-\rho\right) + \beta_1\left(X_t - \rho X_{t-1}\right) + \left(\varepsilon_t - \rho\varepsilon_{t-1}\right) = \beta_0\left(1-\rho\right) + \beta_1\left(X_t - \rho X_{t-1}\right) + u_t$

$\Rightarrow \quad Y_t' = \beta_0' + \beta_1' X_t' + u_t$ (Standard Simple linear regression with independent errors)

where:

$Y_t' = Y_t - \rho Y_{t-1} \qquad X_t' = X_t - \rho X_{t-1} \qquad \beta_0' = \beta_0\left(1-\rho\right) \qquad \beta_1' = \beta_1$

In Practice, we need to estimate ρ with a sample based value *r*

$Y_t' = Y_t - rY_{t-1} \qquad X_t' = X_t - rX_{t-1}$

Fit: $\hat{Y}' = b_0' + b_1' X'$ and if errors are uncorrelated, back transform to:

$\hat{Y} = b_0 + b_1 X$ where: $b_0 = \dfrac{b_0'}{1-r} \qquad s\{b_0\} = \dfrac{s\{b_0'\}}{1-r} \qquad b_1 = b_1' \qquad s\{b_1\} = s\{b_1'\}$

Cochrane-Orcutt Method

- Start by estimating r in Model: $e_t = re_{t-1} + u_t$ by regression through the origin for residuals (see below)
- Fit transformed regression model (previous slide)
- Check to see if new residuals are uncorrelated (Durbin-Watson test), based on the transformed model
- If uncorrelated, stop and keep current model
- If correlated, repeat process with new estimate r based on current regression residuals from the original (back transformed) model

$$r = \frac{\sum_{t=2}^{n} e_{t-1}e_t}{\sum_{t=2}^{n} e_{t-1}^2}$$

Hildreth-Lu and First Difference Methods

- **Hildreth-Lu Method**
  - Find value of r (between 0 and 1) that minimizes the SSE for the transformed model by grid search
  - Apply the transformed analysis based on the estimated r

- **First Differences Method**
  - Uses $r = 1$ in transformed model ($Y_t' = Y_t - Y_{t-1}$ $X_t' = X_t - X_{t-1}$)
  - Set $b_0' = 0$ and fits regression through origin of Y' on X'
  - When back-transforming:

$$b_0 = \overline{Y} - b_1'\overline{X} \qquad b_1 = b_1'$$

Example: Central Valley California Water Pumped and Salmon Trapped in Pumps

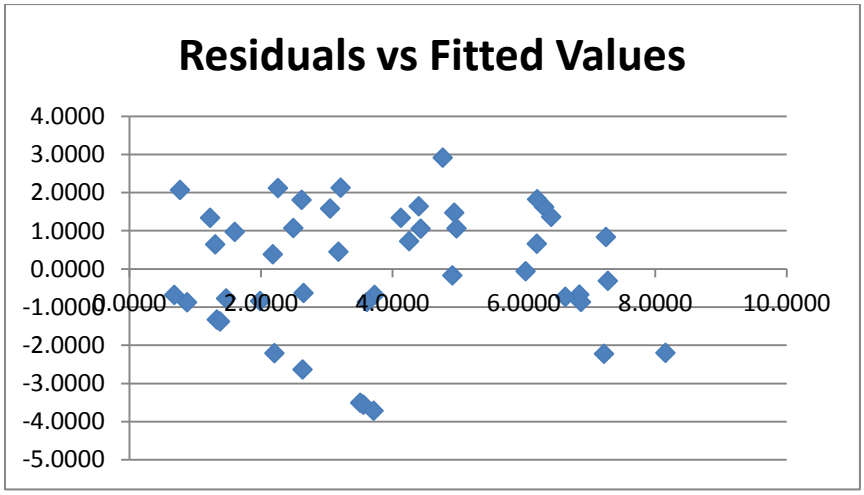
Source: M. Vincent (2013). "Fishy Statistics: In re Consolidated Salmonid Cases and the Problem of Autocorrelation," Jurimetrics Journal of Law, Science and Technology, Vol. 53, #2, pp. 143-162.

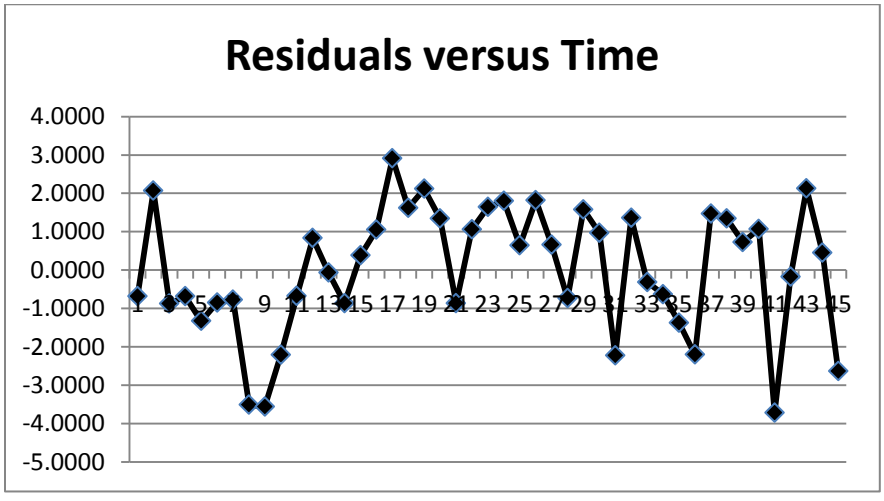
Study involved n = 45 periods from 1998-2006 (5 periods per 9 years)

Response Variable: Y = ln(# Salmon Trapped in pumps)

Predictor Variable: X = Amount of Water Exported to California's Central valley

ANOVA						
	df	*SS*	*MS*	*F*	*gnificance F*	
Regression	1	197.2211	197.2211	70.3118	0.0000	
Residual	43	120.6129	2.8050			
Total	44	317.8341				
	Coefficients	*andard Err*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	0.6551	0.4606	1.4222	0.1622	-0.2738	1.5841
water.x	0.0156	0.0019	8.3852	0.0000	0.0119	0.0194





Durbin-Watson Test

year	period.yr	water.x	fishtrap	lnft.y	period	y-hat	e	(e.t-e.t-1)^2	(e.t)^2	e.t*e.t-1	(e.t-1)^2
1998	1	196.57	21	3.0445	1	3.7301	-0.6856	0.0000	0.4700	0.0000	0.0000
1998	2	7.29	17	2.8332	2	0.7692	2.0640	7.5603	4.2602	-1.4150	0.4700
1998	3	14.31	1	0.0000	3	0.8790	-0.8790	8.6613	0.7726	-1.8143	4.2602
1998	4	1.87	1	0.0000	4	0.6844	-0.6844	0.0379	0.4684	0.6016	0.7726
1998	5	43.22	1	0.0000	5	1.3312	-1.3312	0.4184	1.7722	0.9111	0.4684
1999	1	85.36	3.13	1.1417	6	1.9904	-0.8487	0.2328	0.7203	1.1299	1.7722
1999	2	52.2	2	0.6931	7	1.4717	-0.7786	0.0049	0.6062	0.6608	0.7203
1999	3	182.8	1	0.0000	8	3.5147	-3.5147	7.4861	12.3529	2.7365	0.6062
1999	4	185.67	1	0.0000	9	3.5596	-3.5596	0.0020	12.6705	12.5107	12.3529
1999	5	99.26	1	0.0000	10	2.2079	-2.2079	1.8271	4.8747	7.8590	12.6705
2000	1	395.93	477.89	6.1694	11	6.8486	-0.6792	2.3367	0.4614	1.4997	4.8747
2000	2	421.69	3232.23	8.0809	12	7.2516	0.8293	2.2757	0.6877	-0.5633	0.4614
2000	3	343.57	389.22	5.9641	13	6.0296	-0.0655	0.8006	0.0043	-0.0543	0.6877
2000	4	189.12	15.57	2.7454	14	3.6135	-0.8681	0.6442	0.7537	0.0568	0.0043
2000	5	97.7	13	2.5649	15	2.1835	0.3814	1.5614	0.1455	-0.3311	0.7537
2001	1	241.58	241.19	5.4856	16	4.4342	1.0514	0.4489	1.1055	0.4011	0.1455
2001	2	262.93	2162.97	7.6792	17	4.7681	2.9111	3.4582	8.4743	3.0608	1.1055
2001	3	361.56	2773.85	7.9280	18	6.3110	1.6170	1.6746	2.6147	4.7072	8.4743
2001	4	102.57	79.25	4.3726	19	2.2596	2.1130	0.2460	4.4646	3.4167	2.6147
2001	5	36.54	13	2.5649	20	1.2267	1.3382	0.6003	1.7907	2.8275	4.4646
2002	1	397.57	406.69	6.0080	21	6.8743	-0.8663	4.8597	0.7505	-1.1593	1.7907
2002	2	276.37	418.86	6.0375	22	4.9784	1.0591	3.7072	1.1217	-0.9175	0.7505
2002	3	239.57	421.04	6.0427	23	4.4027	1.6400	0.3374	2.6895	1.7369	1.1217
2002	4	125.77	83.63	4.4264	24	2.6226	1.8038	0.0269	3.2539	2.9583	2.6895
2002	5	41.65	7	1.9459	25	1.3067	0.6392	1.3563	0.4086	1.1531	3.2539
2003	1	354.86	3060.47	8.0263	26	6.2062	1.8201	1.3945	3.3128	1.1635	0.4086
2003	2	354.58	954.44	6.8611	27	6.2018	0.6593	1.3475	0.4347	1.2000	3.3128
2003	3	382.23	363.88	5.8968	28	6.6343	-0.7375	1.9511	0.5440	-0.4863	0.4347
2003	4	153.42	102.51	4.6300	29	3.0551	1.5749	5.3475	2.4804	-1.1616	0.5440
2003	5	60.46	13	2.5649	30	1.6009	0.9640	0.3732	0.9293	1.5182	2.4804
2004	1	419.93	147.55	4.9942	31	7.2241	-2.2299	10.2007	4.9724	-2.1496	0.9293
2004	2	368.59	2394.42	7.7809	32	6.4210	1.3599	12.8867	1.8494	-3.0325	4.9724
2004	3	423.51	1058.19	6.9643	33	7.2801	-0.3158	2.8080	0.0997	-0.4294	1.8494
2004	4	127.49	7.5	2.0149	34	2.6495	-0.6346	0.1016	0.4027	0.2004	0.0997
2004	5	46.31	1	0.0000	35	1.3796	-1.3796	0.5550	1.9032	0.8754	0.4027
2005	1	479.65	385.81	5.9553	36	8.1583	-2.2030	0.6780	4.8531	3.0392	1.9032
2005	2	274.24	610.57	6.4144	37	4.9451	1.4693	13.4859	2.1590	-3.2369	4.8531
2005	3	222.33	237.55	5.4704	38	4.1330	1.3374	0.0174	1.7885	1.9650	2.1590
2005	4	230.14	145.83	4.9825	39	4.2552	0.7273	0.3722	0.5290	0.9727	1.7885
2005	5	117.72	35.33	3.5648	40	2.4966	1.0682	0.1162	1.1410	0.7769	0.5290
2006	1	195.78	1	0.0000	41	3.7177	-3.7177	22.9047	13.8214	-3.9712	1.1410
2006	2	272.2	114.4	4.7397	42	4.9131	-0.1734	12.5618	0.0301	0.6448	13.8214
2006	3	163.67	208.45	5.3397	43	3.2154	2.1243	5.2795	4.5126	-0.3685	0.0301
2006	4	161.41	37.55	3.6257	44	3.1801	0.4456	2.8179	0.1986	0.9466	4.5126
2006	5	126.73	1	0.0000	45	2.6376	-2.6376	9.5061	6.9568	-1.1754	0.1986
							Sum	155.2706	120.6129	39.2643	113.6561

$$DW = \frac{\sum_{t=2}^{n}\left(e_t - e_{t-1}\right)^2}{\sum_{t=1}^{n} e_t^2} = \frac{155.2706}{120.6129} = 1.2873$$

$d_L\left(1,45\right) = 1.48 \qquad d_U\left(1,45\right) = 1.57$ Evidence of autocorrelated errors

Cochrane-Orcutt Method

$$r_{CO} = \frac{\sum_{t=2}^{n} e_{t-1}e_t}{\sum_{t=2}^{n} e_{t-1}^2} = \frac{39.2643}{113.6561} = 0.3455 \qquad Y_t^{'} = Y_t - 0.3455Y_{t-1} \qquad X_t^{'} = X_t - 0.3455X_{t-1}$$

	Coefficients	*tandard Err*	*t Stat*	*P-value*
Intercept	0.6895	0.3434	2.0082	0.0511
X'	0.0137	0.0018	7.4550	0.0000

$$\hat{Y}' = b_0^{'} + b_1^{'}X' = 0.6895 + 0.0137$$

$$DW = 1.6662 \quad d_L\left(1, 45-1=44\right) \approx 1.48 \quad d_U\left(1, 45-1=44\right) \approx 1.57 \quad \Rightarrow \quad \text{Uncorrelated Errors}$$

$$b_0 = \frac{b_0^{'}}{1-r} = \frac{0.6895}{1-0.3455} = 1.0535 \qquad s\left\{b_0\right\} = \frac{s\left\{b_0^{'}\right\}}{1-r} = \frac{0.3434}{1-0.3455} = 0.5257$$

$$b_1 = b_1^{'} = 0.0137 \qquad s\left\{b_1\right\} = s\left\{b_1^{'}\right\} = 0.0018$$

$$\hat{Y} = 1.0535 + 0.0137X$$

All Methods - R Program

```
salmon <- read.csv("E:\\blue_drive\\sta4210\\salmon_tsreg.csv",header=TRUE)
attach(salmon); names(salmon)

### Regression Model, DW Statistic, and Cochrane-Orthcutt R
n <- length(lnft.y)
salmon.mod1 <- lm(lnft.y ~ water.x)
summary(salmon.mod1)
anova(salmon.mod1)
SSE.mod1 <- deviance(salmon.mod1)
e.mod1 <- residuals(salmon.mod1)

plot(e.mod1,type="o",xlab="Time",ylab="Residual")

DW1.mod1 <- 0
CO1.mod1 <- 0
CO2.mod1 <- 0

for (t in 2:n) {
DW1.mod1 <- DW1.mod1 + (e.mod1[t] - e.mod1[t-1])^2
CO1.mod1 <- CO1.mod1 + e.mod1[t-1]*e.mod1[t]
CO2.mod1 <- CO2.mod1 + (e.mod1[t-1])^2
}

(DW.mod1 <- DW1.mod1/SSE.mod1)
(COr.mod1 <- CO1.mod1/CO2.mod1)

#### Output #####
> (DW.mod1 <- DW1.mod1/SSE.mod1)
1.287346
> (COr.mod1 <- CO1.mod1/CO2.mod1)
0.3454653
```

Program Continued

```
################### Cochrane-Orcutt Method

Yt <- lnft.y[2:n] - (COr.mod1 * lnft.y[1:(n-1)])
Xt <- water.x[2:n] - (COr.mod1 * water.x[1:(n-1)])
# print(cbind(Yt,Xt))

CO.mod1t <- lm(Yt ~ Xt)
(CO.mod1t.sum <- summary(CO.mod1t))
anova(CO.mod1t)
e.CO1t <- residuals(CO.mod1t)
SSE.CO1t <- deviance(CO.mod1t)

DW1.CO1t <- 0
for (t in 2:(n-1)) DW1.CO1t <- DW1.CO1t + (e.CO1t[t] - e.CO1t[t-1])^2
(DW.CO1t <- DW1.CO1t/SSE.CO1t)

(b0.CO1o <- coef(CO.mod1t.sum)[1,1]/(1-COr.mod1))
(b1.CO1o <- coef(CO.mod1t.sum)[2,1])
(s.b0.CO1o <- coef(CO.mod1t.sum)[1,2]/(1-COr.mod1))
(s.b1.CO1o <- coef(CO.mod1t.sum)[2,2])

####### Output ######

Call:lm(formula = Yt ~ Xt)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.689554   0.343373   2.008   0.0511 .
Xt          0.013680   0.001835   7.455 3.27e-09 ***

> (DW.CO1t <- DW1.CO1t/SSE.CO1t)
1.666194
> (b0.CO1o <- coef(CO.mod1t.sum)[1,1]/(1-COr.mod1))
1.053503
> (b1.CO1o <- coef(CO.mod1t.sum)[2,1])
[1] 0.01367989
> (s.b0.CO1o <- coef(CO.mod1t.sum)[1,2]/(1-COr.mod1))
0.5246058
> (s.b1.CO1o <- coef(CO.mod1t.sum)[2,2])
[1] 0.001834967

##################### Hildreth-Lu Method #########

SSE.r <- matrix(rep(0,2*length(seq(0.01,0.99,0.01))),ncol=2)
r.count <- 0
HL.r <- 0
min.SSE <- 10000000
for (r in seq(0.01,0.99,0.01))   {
r.count <- r.count + 1
Yt <- lnft.y[2:n] - (HL.r * lnft.y[1:(n-1)])
Xt <- water.x[2:n] - (HL.r * water.x[1:(n-1)])
SSE.r[r.count,1] <- r
SSE.r[r.count,2] <- deviance(lm(Yt ~ Xt))
if (deviance(lm(Yt ~ Xt)) < min.SSE) {
min.SSE <- deviance(lm(Yt ~ Xt))
HL.r <- r
}}
HL.r
# SSE.r
Yt <- lnft.y[2:n] - (HL.r * lnft.y[1:(n-1)])
Xt <- water.x[2:n] - (HL.r * water.x[1:(n-1)])
HL.mod1t <- lm(Yt ~ Xt)
(HL.mod1t.sum <- summary(HL.mod1t))
e.HL1t <- residuals(HL.mod1t)
SSE.HL1t <- deviance(HL.mod1t)
DW1.HL1t <- 0
for (t in 2:(n-1)) DW1.HL1t <- DW1.HL1t + (e.HL1t[t] - e.HL1t[t-1])^2
(DW.HL1t <- DW1.HL1t/SSE.HL1t)
(b0.HL1o <- coef(HL.mod1t.sum)[1,1]/(1-HL.r))
(b1.HL1o <- coef(HL.mod1t.sum)[2,1])
(s.b0.HL1o <- coef(HL.mod1t.sum)[1,2]/(1-HL.r))
(s.b1.HL1o <- coef(HL.mod1t.sum)[2,2])
```

Continued Below

```
#### Output #######

> HL.r
[1] 0.44

Call: lm(formula = Yt ~ Xt)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.641084   0.315729   2.030   0.0487 *
Xt          0.013179   0.001812   7.272 5.95e-09 ***

> (DW.HL1t <- DW1.HL1t/SSE.HL1t)
1.783135
> (b0.HL1o <- coef(HL.mod1t.sum)[1,1]/(1-HL.r))
[1] 1.144794
> (b1.HL1o <- coef(HL.mod1t.sum)[2,1])
[1] 0.01317853
> (s.b0.HL1o <- coef(HL.mod1t.sum)[1,2]/(1-HL.r))
[1] 0.5638018
> (s.b1.HL1o <- coef(HL.mod1t.sum)[2,2])
[1] 0.001812196

############## First Differences Method

Yt <- lnft.y[2:n]-lnft.y[1:(n-1)]
Xt <- water.x[2:n]-water.x[1:(n-1)]

FD.mod1t <- lm(Yt ~ -1 + Xt)
(FD.mod1t.sum <- summary(FD.mod1t))
anova(FD.mod1t)

(b0.FD1o <- mean(lnft.y) - (coef(FD.mod1t.sum)[1,1]*mean(water.x)))
(b1.FD1o <- coef(FD.mod1t.sum)[1,1])
(s.b1.FD1o <- coef(FD.mod1t.sum)[1,2])

###### Output #######

Call: lm(formula = Yt ~ -1 + Xt)
Coefficients:
   Estimate Std. Error t value Pr(>|t|)
Xt 0.011810   0.001698   6.956 1.49e-08 ***

> (b0.FD1o <- mean(lnft.y) - (coef(FD.mod1t.sum)[1,1]*mean(water.x)))
[1] 1.450595
> (b1.FD1o <- coef(FD.mod1t.sum)[1,1])
[1] 0.0118096
> (s.b1.FD1o <- coef(FD.mod1t.sum)[1,2])
[1] 0.001697646
```

Forecasting with Autocorrelated Errors

Makes use of any of the 3 estimation techniques (C-O, H-L, First Differences):

$Y_t = \beta_0 + \beta_1 X_t + \varepsilon_t \qquad \varepsilon_t = \rho\varepsilon_{t-1} + u_t$

$\Rightarrow \quad Y_t = \beta_0 + \beta_1 X_t + \rho\varepsilon_{t-1} + u_t \quad \Rightarrow \quad \Rightarrow \quad Y_{n+1} = \beta_0 + \beta_1 X_{n+1} + \rho\varepsilon_n + u_{n+1}$

3 Elements:

1. Expected Value: $\beta_0 + \beta_1 X_{n+1}$ Estimated as $\hat{Y}_{n+1} = b_0 + b_1 X_{n+1}$
2. Multiple of period *n* Error Term: $\rho\varepsilon_n$ Estimated as re_n
3. Current disturbance $u_{n+1} \sim N\left(0, \sigma^2\right)$

Forecast for period $n+1$ (note the notation is "Forecast", not F-distribution :

$$F_{n+1} = \hat{Y}_{n+1} + re_n$$

Standard Error of the Prediction (based on transformed model):

$$s^2\{\text{pred}\} = MSE'\left[1 + \frac{1}{n-1} + \frac{\left(X'_{n+1} - \overline{X}'\right)^2}{\sum_{i=2}^{n}\left(X'_i - \overline{X}'\right)^2}\right]$$

Approximate 95% PI: $F_{n+1} \pm t\left(1-(\alpha/2); n-3\right)s\{\text{pred}\}$ (First Differences has n-1 observations)

Example: Central Valley California Water Pumped and Salmon Trapped in Pumps

Suppose that we know in period 46, there will be X = 250 units of water exported to the Central Valley.

From the Cochrane –Orcutt method, we have the following results:

$$\hat{Y} = 1.0535 + 0.0137X \quad r_{CO} = 0.3455 \quad X_{45} = 126.73 \quad Y_{45} = 0$$

$$\Rightarrow \quad \hat{Y}_{45} = 1.0535 + 0.0137(126.73) = 2.7897 \quad e_{45} = 0 - 2.7897 = -2.7897$$

$$\Rightarrow \quad F_{46} = \hat{Y}_{46} + re_{45} = 1.0535 + 0.0137(250) + 0.3455(-2.7897) = 4.4785 - 0.9638 = 3.5147$$

$$\overline{X}' = 135.4292 \quad X'_{46} = 250 - 0.3455(126.73) = 206.2148 \quad \sum_{i=2}^{n}\left(X'_i - \overline{X}'\right)^2 = 733642.6 \quad MSE' = 2.4702$$

$$s^2\{\text{pred}\} = MSE'\left[1 + \frac{1}{n-1} + \frac{\left(X'_{n+1} - \overline{X}'\right)}{\sum_{i=2}^{n}\left(X'_i - \overline{X}'\right)^2}\right] = 2.4702\left[1 + \frac{1}{45-1} + \frac{\left(206.2148 - 135.4292\right)^2}{733642.6}\right] = 2.5432$$

Approximate 95% PI: $F_{n+1} \pm t\left(1-(\alpha/2); n-3\right)s\{\text{pred}\} \equiv 3.5147 \pm 2.0181\sqrt{2.5432} \equiv 3.5147 \pm 3.2184 \equiv (0.2963, 6.7331)$

Recalling that Y is the log of the numbers of fish caught, we can take exponentials of the end points to get the number of fish predicted: exp(0.2963) = 1.345, exp(6.7331) = 839.75. This is a very wide prediction interval! Estimating the mean of all months with X = 250, would be much more precise.

Generalized Least Squares

For models with correlated errors, the observations are not independent, so that the variance-covariance matrix of **Y** is not diagonal. For specific structures (such as AR(p)), we can set-up the matrix in terms of a few parameters. Then, we can make a linear transformation of **Y** (and **X**), so that the transformed **Y** has a variance-covariance matrix of the form $\sigma^2\mathbf{I}$. For the AR(1) model, we have the following variance-covariance structure and transformation:

$$\sigma^2\{\boldsymbol{\varepsilon}\} = \sigma^2\left\{\begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \vdots\\ \varepsilon_n\end{bmatrix}\right\} = \frac{\sigma^2}{1-\rho^2}\begin{bmatrix}1 & \rho & \cdots & \rho^{n-1}\\ \rho & 1 & \cdots & \rho^{n-2}\\ \vdots & \vdots & \ddots & \vdots\\ \rho^{n-1} & \rho^{n-2} & \cdots & 1\end{bmatrix}$$

$$\mathbf{Y}^* = \mathbf{T}^{-1}\mathbf{Y} = \begin{bmatrix}\sqrt{1-\rho^2} & 0 & 0 & \cdots & 0 & 0\\ -\rho & 1 & 0 & \cdots & 0 & 0\\ 0 & -\rho & 1 & \cdots & 0 & 0\\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & \cdots & 1 & 0\\ 0 & 0 & 0 & \cdots & -\rho & 1\end{bmatrix}\begin{bmatrix}Y_1\\ Y_2\\ \vdots\\ Y_n\end{bmatrix} = \mathbf{T}^{-1}\mathbf{X}\boldsymbol{\beta} + \mathbf{T}^{-1}\boldsymbol{\varepsilon} = \mathbf{X}^*\boldsymbol{\beta} + \boldsymbol{\varepsilon}^* \qquad \sigma^2\{\mathbf{Y}^*\} = \sigma^2\mathbf{I}$$

We end up with estimates for β, σ^2, ρ. This is referred to as **Estimated Generalized Least Squares (EGLS)**, aka **Feasible Generalized Least Squares (FGLS)**.

R Program (Makes use of the nlme package)

```
salmon <- read.csv("E:\\blue_drive\\sta4210\\salmon_tsreg.csv",header=TRUE)
attach(salmon); names(salmon)

salmon.mod1 <- lm(lnft.y ~ water.x)
summary(salmon.mod1)
plot(residuals(salmon.mod1),type="l")

install.packages("car")
library(car)
durbinWatsonTest(salmon.mod1)

install.packages("nlme")
library(nlme)
salmon.mod2 <- gls(lnft.y ~ water.x, correlation=corARMA(p=1), method='ML')
summary(salmon.mod2)

###### Output ##############

> summary(salmon.mod1)
Call: lm(formula = lnft.y ~ water.x)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.655146   0.460642   1.422    0.162
water.x     0.015643   0.001866   8.385 1.36e-10 ***

> durbinWatsonTest(salmon.mod1)
 lag Autocorrelation D-W Statistic p-value
   1       0.3255393      1.287346    0.01
 Alternative hypothesis: rho != 0
```

Continued Below

```
> summary(salmon.mod2)
Generalized least squares fit by maximum likelihood
  Model: lnft.y ~ water.x
  Data: NULL
      AIC      BIC    logLik
  173.349 180.5757 -82.67451

Correlation Structure: AR(1)
 Formula: ~1
 Parameter estimate(s):
      Phi
0.4217852

Coefficients:
               Value Std.Error  t-value p-value
(Intercept) 1.0903352 0.5405791 2.016976    0.05
water.x     0.0132779 0.0017986 7.382224    0.00

Residual standard error: 1.67204
Degrees of freedom: 45 total; 43 residual
```

In many cases when the errors are highly correlated the standard error of b_1 will be larger based on EGLS than OLS, reflecting the fact that when errors are correlated, our "effective" sample size is smaller than *n*. In this case, they are about the same (slightly smaller for EGLS). Note that while the errors are correlated, the autocorrelation parameter is estimated to be 0.422 (not terribly high). The estimator, and its variance-covariance matrix are given below.

$$\hat{\boldsymbol{\beta}}_{EGLS} = \left(\mathbf{X}' \left(\hat{\boldsymbol{\Sigma}} \right)^{-1} \mathbf{X} \right)^{-1} \mathbf{X}' \left(\hat{\boldsymbol{\Sigma}} \right)^{-1} \mathbf{Y} \qquad s^2 \left\{ \hat{\boldsymbol{\beta}}_{EGLS} \right\} = \left(\mathbf{X}' \left(\hat{\boldsymbol{\Sigma}} \right)^{-1} \mathbf{X} \right)^{-1}$$

Chapter 13 – Nonlinear Regression

In many economic and biological settings, the relationship between a response variable and its predictor variable(s) is nonlinear, often based on a theoretical model with differential equations. While, we have used polynomial regression to model nonlinear relationships, often times we have a specific nonlinear function, and we want to estimate the model's parameter(s). We use the term nonlinear to be with respect to the model parameters. Iterative methods of nonlinear least squares are used to fit the models.

Nonlinear Relations with respect to X – Linear wrt β^s

1) Polynomial Models: $E\{Y_i\} = \beta_0 + \beta_1 X_i + \beta_2 X_i^2$

$$\frac{\partial E\{Y_i\}}{\partial X_i} = \frac{\partial}{\partial X_i}\left[\beta_0 + \beta_1 X_i + \beta_2 X_i^2\right] = 0 + \beta_1 + 2\beta_2 X_i = h(X_i)$$

$$\frac{\partial E\{Y_i\}}{\partial \beta_0} = 1 \qquad \frac{\partial E\{Y_i\}}{\partial \beta_1} = X_i \qquad \frac{\partial E\{Y_i\}}{\partial \beta_2} = X_i^2$$ None are functions of $\boldsymbol{\beta} = \begin{bmatrix}\beta_0\\ \beta_1\\ \beta_2\end{bmatrix}$

2) Transformed Variable Models: $E\left\{\sqrt{Y_i}\right\} = \beta_0 + \beta_1 \ln(X_{i1}) + \beta_2\left(\frac{1}{X_{i2}}\right)$

$$\frac{\partial E\left\{\sqrt{Y_i}\right\}}{\partial X_{i1}} = \beta_1\left(\frac{1}{X_{i1}}\right) = h_1(X_{i1}) \qquad \frac{\partial E\left\{\sqrt{Y_i}\right\}}{\partial X_{i2}} = -\beta_2\left(\frac{1}{X_{i2}^2}\right) = h_2(X_{i2})$$

$$\frac{\partial E\left\{\sqrt{Y_i}\right\}}{\partial \beta_0} = 1 \qquad \frac{\partial E\left\{\sqrt{Y_i}\right\}}{\partial \beta_1} = \ln(X_{i1}) \qquad \frac{\partial E\left\{\sqrt{Y}_i\right\}}{\partial \beta_2} = \left(\frac{1}{X_{i2}}\right)$$ None are functions of $\boldsymbol{\beta} = \begin{bmatrix}\beta_0\\ \beta_1\\ \beta_2\end{bmatrix}$

In each case: $E\{Y_i\} = f(\mathbf{X_i}, \boldsymbol{\beta}) = \mathbf{X_i'}\boldsymbol{\beta}$

Case 1: $\mathbf{X_i'} = \begin{bmatrix}1 & X_i & X_i^2\end{bmatrix}$ Case 2: $\mathbf{X_i'} = \begin{bmatrix}1 & \ln(X_{i1}) & \frac{1}{X_{i2}}\end{bmatrix}$

These models can be fit by ordinary least squares with the transformed variables acting as the X^s.

Nonlinear Regression Models

Nonlinear Regression models often use $\boldsymbol{\gamma}$ as vector of coefficients to distinguish from linear models:

Exponential Regression Models (Often used for modeling growth, where rate of growth changes):

$E\{Y_i\} = \gamma_0 \exp(\gamma_1 X_i) \quad \Rightarrow \quad \frac{\partial E\{Y_i\}}{\partial \gamma_0} = \exp(\gamma_1 X_i) \qquad \frac{\partial E\{Y_i\}}{\partial \gamma_1} = \gamma_0 X_i \exp(\gamma_1 X_i)$ functions of $\boldsymbol{\gamma}$

$f(\mathbf{X_i}, \boldsymbol{\gamma}) = \gamma_0 \exp(\gamma_1 X_i) \neq \mathbf{X_i'}\boldsymbol{\gamma}$

More general exponential model (with errors independent and $N(0, \sigma^2)$):

$Y_i = \gamma_0 + \gamma_1 \exp(\gamma_2 X_i) + \varepsilon_i$ Typically, $\gamma_0 > 0, \gamma_1 > 0, \gamma_2 < 0$

$\Rightarrow$ Intercept: $E(Y_i \mid X_i = 0) = \gamma_0 + \gamma_1(1) = \gamma_0 + \gamma_1$

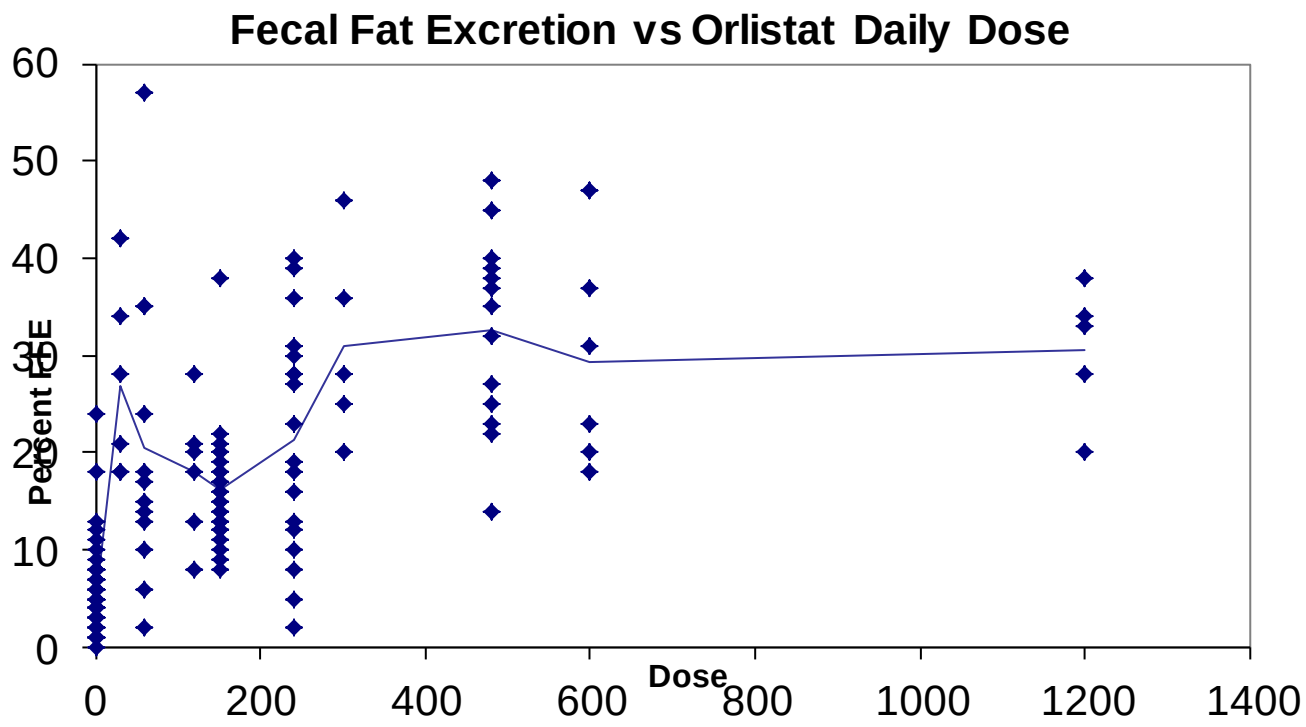
$\Rightarrow$ Asymptote: $E(Y_i \mid X_i \to \infty) = \gamma_0 + \gamma_1(0) = \gamma_0$

$\Rightarrow$ "Half-way" Point: $E\left(Y_i \mid X_i = \frac{0.693}{|\gamma_2|}\right) = \gamma_0 + \gamma_1 \exp\left(\gamma_2\left(\frac{0.693}{|\gamma_2|}\right)\right) = \gamma_0 + \gamma_1 \exp(-0.693) = \gamma_0 + \left(\frac{\gamma_1}{2}\right)$

Example – Orlistat for Fat Reduction

Source: Zhi, J., Melia, A.T., Guerciolini, R., et al, (1994). "Retrospective Population-Based Analysis of the Dose-Response (Fecal Fat Excretion) Relationship of Orlistat in Normal and Obese Volunteers," Clinical Pharmacology and Therapeutics, 56: 82-85.

- 163 Patients assigned to one of the following doses (mg/day) of orlistat: 0, 60,120,150,240,300,480,600,1200
- Response measured was fecal fat excretion (purpose is to inhibit fat absorption, so higher levels of response are considered favorable)
- Plot of raw data displays a generally increasing but nonlinear pattern and large amount of variation across subjects



$$Y = \gamma_0 + \frac{\gamma_1 x}{\gamma_2 + x} + \varepsilon$$

Simple Maximum Effect (E_{max}) model:

- γ_0 **≡ Mean Response at Dose 0**
- γ_1 **≡ Maximal Effect of Orlistat ($\gamma_0 + \gamma_1$ = Maximum Mean Response)**
- γ_2 **≡ Dose providing 50% of maximal effect (ED_{50})**

Nonlinear Least Squares

$$f\left(\mathbf{X_i},\boldsymbol{\gamma}\right)=f_i\left(\boldsymbol{\gamma}\right)=f\left(\gamma_0,\gamma_1,\gamma_2\right)=\gamma_0+\frac{\gamma_1 X_i}{\gamma_2+X_i}$$

$$\frac{\partial f\left(\mathbf{X_i},\boldsymbol{\gamma}\right)}{\partial \boldsymbol{\gamma}'}=F_i\left(\boldsymbol{\gamma}\right)=\frac{\partial f_i\left(\boldsymbol{\gamma}\right)}{\partial \boldsymbol{\gamma}'}=\begin{bmatrix}1 & \dfrac{X_i}{\gamma_2+X_i} & \dfrac{-\gamma_1 X_i}{\left(\gamma_2+X_i\right)^2}\end{bmatrix}$$

$$\mathbf{Y}=\begin{bmatrix}Y_1\\ \vdots \\ Y_n\end{bmatrix} \qquad \mathbf{f}\left(\boldsymbol{\gamma}\right)=\begin{bmatrix}f_1\left(\gamma\right)\\ \vdots \\ f_n\left(\gamma\right)\end{bmatrix}=\begin{bmatrix}\gamma_0+\dfrac{\gamma_1 X_1}{\gamma_2+X_1}\\ \vdots \\ \gamma_0+\dfrac{\gamma_1 X_n}{\gamma_2+X_n}\end{bmatrix}$$

$$\mathbf{F}\left(\boldsymbol{\gamma}\right)=\begin{bmatrix}F_1\left(\gamma\right)\\ \vdots \\ F_n\left(\gamma\right)\end{bmatrix}=\begin{bmatrix}1 & \dfrac{X_1}{\gamma_2+X_1} & \dfrac{-\gamma_1 X_1}{\left(\gamma_2+X_n\right)^2}\\ \vdots & \vdots & \vdots \\ 1 & \dfrac{X_n}{\gamma_2+X_n} & \dfrac{-\gamma_1 X_n}{\left(\gamma_2+X_n\right)^2}\end{bmatrix}$$

F acts like the **X** matrix in linear regression (but depends on parameters)

Goal: Choose $\hat{\gamma}_0,\hat{\gamma}_1,\hat{\gamma}_2$ that minimize error sum of squares:

$$Q=SSE\left(\boldsymbol{\gamma}\right)=\sum_{i=1}^{n}\left(Y_i-\left[\gamma_0+\frac{\gamma_1 X_i}{\gamma_2+X_i}\right]\right)^2=$$

$$=\left(\mathbf{Y}-\mathbf{f}\left(\boldsymbol{\gamma}\right)\right)'\left(\mathbf{Y}-\mathbf{f}\left(\boldsymbol{\gamma}\right)\right)$$

$$\frac{\partial Q}{\partial \gamma_j}=-2\sum_{i=1}^{n}\left(Y_i-\left[\gamma_0+\frac{\gamma_1 X_i}{\gamma_2+X_i}\right]\right)F_i\left(\gamma_j\right) \quad j=0,1,2$$

$$\frac{\partial Q}{\partial \boldsymbol{\gamma}'}=-2\left[\mathbf{Y}-\mathbf{f}\left(\boldsymbol{\gamma}\right)\right]^{\mathbf{T}}\mathbf{F}\left(\boldsymbol{\gamma}\right)\overset{set}{=}\begin{bmatrix}0 & 0 & 0\end{bmatrix}$$

Estimated Variance-Covariance Matrix

$$s^2\left\{\hat{\boldsymbol{\gamma}}\right\} = s^2\left(\hat{\mathbf{F}}'\hat{\mathbf{F}}\right)^{-1}$$

$$s^2 = \frac{\left(\mathbf{Y}-\hat{\mathbf{f}}\right)^{\mathbf{T}}\left(\mathbf{Y}-\hat{\mathbf{f}}\right)}{n-p}$$

$$s\left\{\hat{\gamma}_i\right\} = s\sqrt{\left(\hat{\mathbf{F}}'\hat{\mathbf{F}}\right)^{-1}_{(i+1,i+1)}}$$

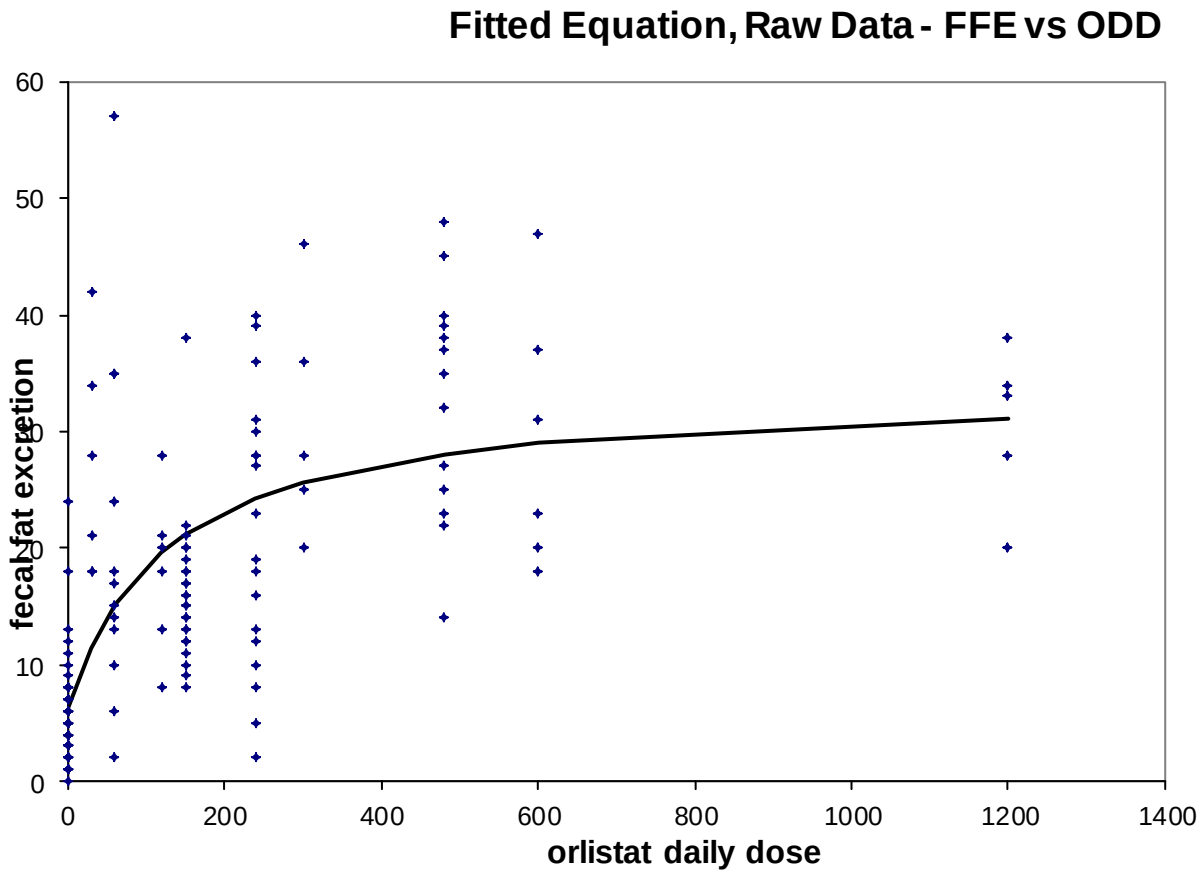
Note: KNNL uses $\mathbf{g}$ for $\hat{\boldsymbol{\gamma}}$ and $\mathbf{D}$ for $\hat{\mathbf{F}}$

Example – Orlistat E_{max} Model

- Reasonable Starting Values:
  - γ_0: Mean of 0 Dose Group: 5
  - γ_1: Difference between highest mean and dose 0 mean: 33-5=28
  - γ_2: Dose with mean halfway between 5 and 33: 160
- Create Vectors $\boldsymbol{Y}$ and $f(\boldsymbol{\gamma}^0)$
- Generate matrix $\boldsymbol{F}(\boldsymbol{\gamma}^0)$
- Obtain first "new" estimate of $\boldsymbol{\gamma}$
- Continue to Convergence

iteration	g0	g1	g2	SSE	Delta(g)
0	5.0000	28.0000	160.0	13541.6	
1	6.2379	28.5863	140.9	12945.5	365.5745418
2	6.1771	28.1281	133.7	12942.9	52.82513814
3	6.1507	27.9163	129.9	12942.2	14.44158887
4	6.1361	27.7967	127.8	12942.0	4.506448063
5	6.1277	27.7272	126.5	12941.9	1.510150161
6	6.1227	27.6861	125.8	12941.9	0.526393989
7	6.1197	27.6615	125.4	12941.9	0.187692352
8	6.1180	27.6467	125.1	12941.9	0.067822683
9	6.1169	27.6377	125.0	12941.9	0.024703325
10	6.1162	27.6323	124.9	12941.9	0.009040833
11	6.1158	27.6291	124.8	12941.9	0.003318268
12	6.1156	27.6271	124.8	12941.9	0.001220029
13	6.1155	27.6259	124.8	12941.9	0.000449042
14	6.1154	27.6251	124.7	12941.9	0.000165379
15	6.1153	27.6247	124.7	12941.9	6.09317E-05

$$\hat{Y} = 6.12 + \frac{27.62X}{124.7 + X}$$



Variance Estimates/Confidence Intervals

$$s^2 = \frac{\sum_{i=1}^{163}\left(Y_i - f_i\left(\hat{\boldsymbol{\gamma}}\right)\right)^2}{163-3} = 80.89$$

$$\mathbf{s^2}\left\{\hat{\boldsymbol{\gamma}}\right\} = s^2\left(\hat{\mathbf{F}}'\hat{\mathbf{F}}\right)^{-1} = \begin{bmatrix} 1.1594 & -0.7219 & 15.609 \\ -0.7219 & 12.081 & 130.14 \\ 15.609 & 130.14 & 2238.76 \end{bmatrix}$$

Parameter	Estimate	Std. Error	95% CI
γ_0	6.12	1.08	(3.96 , 8.28)
γ_1	27.62	3.48	(20.66 , 34.58)
γ_2	124.7	47.31	(30.08 , 219.32)

Notes on Nonlinear Least Squares

Large-Sample Theory:

When $\varepsilon_i \sim N\left(0,\sigma^2\right)$ independent , for large *n*: $\hat{\boldsymbol{\gamma}}$ is approximately normal

$E\left\{\hat{\boldsymbol{\gamma}}\right\} \approx \boldsymbol{\gamma}$ Approximate $\sigma^2\left\{\hat{\boldsymbol{\gamma}}\right\}$ estimated by $\mathbf{s}^2\left\{\hat{\boldsymbol{\gamma}}\right\} = MSE\left(\hat{\mathbf{F}}'\hat{\mathbf{F}}\right)^{-1}$

$\Rightarrow \quad \hat{\boldsymbol{\gamma}} \sim N\left(\boldsymbol{\gamma}, \sigma^2\left(\mathbf{F}'\mathbf{F}\right)^{-1}\right)$ (Approximately)

For small samples:

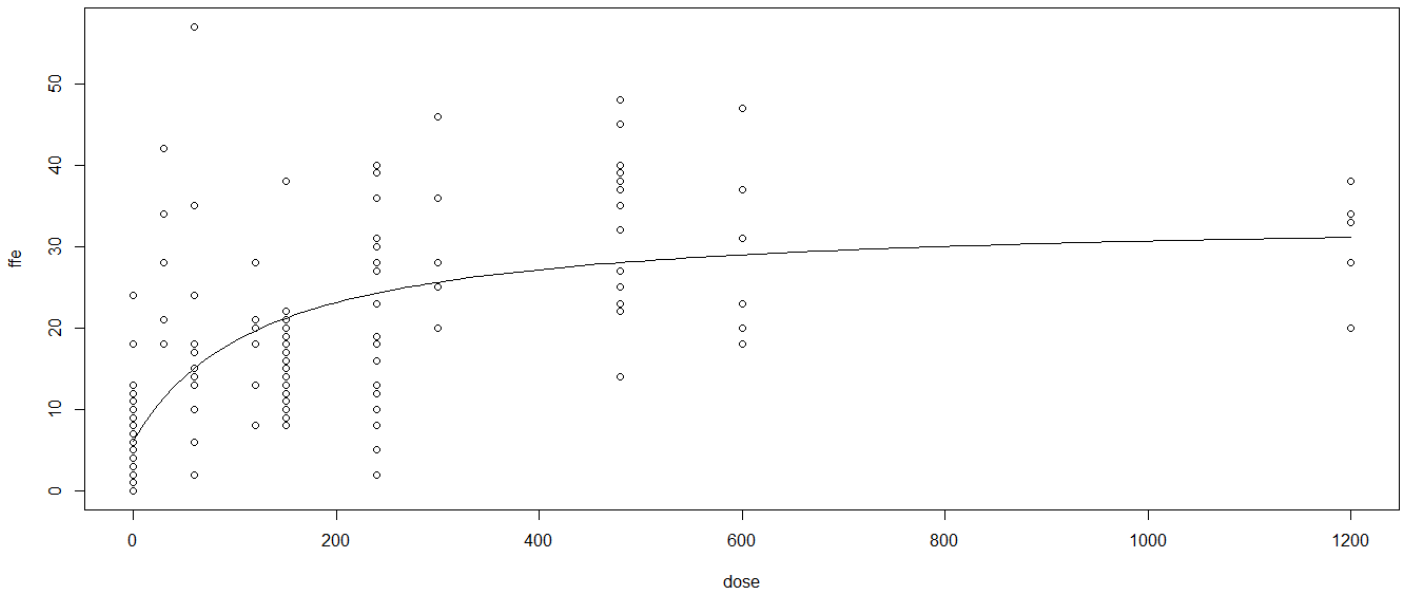
- When errors are normal, independent, with constant variance, we can often use the t-distribution for tests and confidence intervals (software packages do this implicitly)
- When the extent of nonlinearity is extreme, or normality assumptions do not hold, should use bootstrap to estimate standard errors of regression coefficients

R Program

```
orl1 <- read.fwf("http://www.stat.ufl.edu/~winner/data/orlistat.dat", width=c(8,8),
col.names=c("ffe", "orl_dose"))
attach(orl1)

orl.nl1 <- nls(ffe ~ b0 + ((b1*orl_dose)/(b2+orl_dose)),
start=c(b0=1, b1=1, b2=1))
summary(orl.nl1)
vcov(orl.nl1)

plot(orl_dose,ffe,xlab="dose",ylab="ffe")
dose1 <- seq(0,1200,10)
lines(dose1,predict(orl.nl1,list(orl_dose=dose1)))
```

R Output

```
> summary(orl.nl1)

Formula: ffe ~ b0 + ((b1 * orl_dose)/(b2 + orl_dose))

Parameters:
   Estimate Std. Error t value Pr(>|t|)
b0    6.115      1.077   5.679 6.22e-08 ***
b1   27.625      3.476   7.948 3.21e-13 ***
b2  124.731     47.788   2.610  0.00991 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.994 on 160 degrees of freedom

Number of iterations to convergence: 13
Achieved convergence tolerance: 7.273e-06

> vcov(orl.nl1)
          b0         b1          b2
b0  1.159448  -0.721871   15.60886
b1 -0.721871  12.080970  130.13596
b2 15.608859 130.135962 2283.72884
>
```



Selected Regression and R Texts

Kutner, M.H., C.J. Nachtsheim, J. Neter, and W. Li (2005). *Applied Linear Statistical Models*, 5th Ed., McGraw-Hill Irwin, New York.

Faraway, J.J. (2005). *Linear Models with R*, Chapman & Hall/CRC, Boca Raton, FL.

Faraway, J.J. (2006). *Extending the Linear Model with R*, Chapman & Hall/CRC, Boca Raton, FL.

Rawlings, J.O., S.G. Pantula, and D.A. Dickey (1998). *Applied Regression Analysis: A Research Tool*, 2nd Ed., Springer-Verlag, New York.

Cameron, A.C. and P.K. Trivedi (2005). *Microeconometrics:Methods and Applications*, Cambridge University Press, Cambridge.

Greene, W.H. (2008). *Econometric Analysis*, 6th Ed. Prentice-Hall, Upper Saddle River, NJ.

Crawley, M.J. (2013). *The R Book*, 2nd Ed., Wiley, Chichester, West Sussex, UK.