Introduction to R for *Applied Statistical Methods*

Larry Winner
University of Florida

Reading In External Data (Fixed Width)

- Data are in external ASCII file with each variable assigned to a “fixed field”, with one line per unit
- Variable Names are NOT in first row
- You will need to give the following information:
  - The directory and name of the data file (file=)
  - The width of the field for each variable (width=)
  - The name of each variable (col.names=)
- 2 Important Notes for R:
  - The Command: c(1,2,3) strings the numbers 1,2,3 into a vector (concatenates them) and c(“A”,”B”,”C”) does same for the characters A,B,C
  - To assign a function or data to a variable name, the command: <- is used as opposed to =
  - y <- c(20,60,40) creates a vector y with 3 rows (cases)

Example Data File

- File Contains 6 Variables in Fixed Column format

- Var1 ≡ Student Name in Columns 1-30
- Var2 ≡ College/Year Code in Columns 31-33
- Var3 ≡ Exam 1 Score in Columns 34-41
- Var4 ≡ Exam 2 Score in Columns 42-49
- Var5 ≡ Project 1 Score in Columns 50-57
- Var6 ≡ Project 2 Score in Columns 58-65

- Note Var1 is of Length 30, Var2 3, All Others 8

```
**************************************************************************
00000000011111111112222222222333333333344444444455555555566666
12345678901234567890123456789012345678901234567890123456789012345
**************************************************************************
SMITH ROBERT                    7TD      71       83       14       19
WILSON JENNIFER                 8RZ      92       89       18       20
NGUYEN QI                       9YX      84       79       17       15
```

Note: The first 4 rows are only there to show the fixed width format, and would not be part of the data file.

Assume the following information:

File Name: sta666.dat

Data File Directory: C:\data

R Program Directory: C:\Rmisc

In the R program sta666.r, we want to read in this data.

We will assign it the name grades666 within the program, and assign names to the variables as well as their field lengths.

R Command to Read in Data File

```
grades666 <- read.fwf(
file="C:\\data\\sta6166.dat",
width=c(30,3,8,8,8,8),
col.names=c("Student", "Collyear", "Exam1", "Exam2", "Project1", "Project2"))
attach(grades666)
```

- Note that grades666 is now a “data frame” containing 6 variables.
- You have “told” R where the dataset is (C:\data\sta6166.dat, but note you had to use 2 back slashes, you could have also used a single forward slash)
- You “told” R there were 6 variables in fields of widths 30,3,8,8,8,8, respectively
- You have assigned names to the variables
- Originally, to R, the exam1 score is: grades666$Exam1
- By adding the line: **attach(grades666)** you can use the shorter label Exam1 instead of grades666$Exam1 in future commands

Handling Categorical (Factor) Variables

- In many applications, we have categorical variables that are coded as 1,2,…as opposed to their actual levels (names).

- In R, you need to inform the program that these are factor levels, not the corresponding numeric levels (so it can differentiate between a 1-Way ANOVA and Simple Regression, for instance).

- You also may wish to give the levels character names to improve interpreting output.

Example Inoculating Amoebae

- 5 Methods of Inoculation (Conditions)
  1. None (Control)
  2. Heat at 70C for 10 minutes (Heat)
  3. Addition of 10% Formalin (Form)
  4. Heat followed by Formalin (HF)
  5. Formalin, followed after 1 hour by heat (FH)
- n=10 Replicates per Condition (N=50)
- Y=Yield (10^{-4})
- Method is in Columns 1-8
- Yield is in Columns 9-16
- Data is in C:\data\amoeba.dat
- Program in C:\Rmisc\amoeba.r

1	265
1	292
1	268
1	251
1	245
1	192
1	228
1	291
1	185
1	247
2	204
2	234
2	197
2	176
2	240
2	190
2	171
2	190
2	222
2	211
3	191
3	207
3	218
3	201
3	192
3	192
3	214
3	206
3	185
3	163
4	221
4	205
4	178
4	167
4	224
4	225
4	171
4	214
4	283
4	277
5	259
5	206
5	179
5	199
5	180
5	146
5	182
5	147
5	182
5	223

Reading in Data and Declaring Factor

```
amoeba <- read.fwf(
file="http://www.stat.ufl.edu/~winner/data/entozamoeba.dat",
width=c(8,8), col.names=c("method", "yield"))
attach(amoeba)
method <- factor(method, levels=1:5,
 labels=c("Control", "Heat", "Form", "HF", "FH"))
method
```

- method was originally to be treated as interval scale (like if it had been a dose)
- Within the FACTOR command, we change it to being a categorical variable by giving its levels (1,2,3,4,5) and assigning names to those levels (Control, Heat, Form, HF, FH)
- In Place of **levels=1:5** we could have used **levels=c(1,2,3,4,5)**

Obtaining Summary Statistics

- Directly obtain summary statistics for variables in R
  - You can simply have the statistic computed
  - You can save the value of statistic to variable for future use
- Common Statistical Functions (x is the variable):
  - Average: **mean(x)**
  - Standard Deviation: **sd(x)**
  - Variance: **var(x)**
  - Median: **median(x)**
  - Quantile (0<p<1): **quantile(x,p)**
  - Correlation (Between vars x and y): **cor(x,y)**
  - Coefficient of Variation: **100*sd(x)/mean(x)**
  - Min, LQ, Median, Mean, UQ, Max: **summary(x)**
- Obtained by Group: **tapply(x,groupvar,mean)** where mean can be replaced by **sd**, **var**, …

Program – PGA/LPGA Driving Distance/Accuracy

```
pdf("pgalpga1.pdf")

pgalpga <- read.fwf(
                file="http://www.stat.ufl.edu/~winner/data/pgalpga2008.dat",
                width=c(8,8,8),
                col.names=c("Distance", "Accuracy", "Gender"))
attach(pgalpga)

Gender <- factor(Gender, levels=1:2, labels=c("Female", "Male"))

mean(Distance);  mean(Accuracy)
sd(Distance); sd(Accuracy)

tapply(Distance, Gender, mean);  tapply(Accuracy, Gender, mean)
tapply(Distance, Gender, sd); tapply(Accuracy, Gender, sd)

dev.off
```

Output – PGA/LPGA Driving Distance/Accuracy

```
> mean(Distance);  mean(Accuracy)
[1] 269.5116
[1] 65.23927
> sd(Distance); sd(Accuracy)
[1] 22.19603
[1] 5.973796
>
> tapply(Distance, Gender, mean);  tapply(Accuracy, Gender,
mean)
  Female     Male
246.8013 287.6107
  Female     Male
67.59108 63.36497
> tapply(Distance, Gender, sd); tapply(Accuracy, Gender,
sd)
  Female     Male
9.493797 8.554456
  Female     Male
5.768708 5.461109
```

Tables for Categorical Data

- For categorical variables, we use frequency (and relative frequency) tabulations to describe data
  - Frequency Tabulation: **table(x)**
  - Relative Frequencies: **table(x)/sum(table(x))**
- For contingency tables (pairs of variables)
  - Frequency Cross-tabulation: **table(x,y)**
  - Row Marginal totals for x: **margin.table(table(x,y),1)**
  - Column totals for y: **margin.table(table(x,y),2)**
  - Relative Freq Cross-tab: **table(x,y)/sum(table(x,y))**
- Extends to more than 2 Dimensions

Program – UFO Encounters

```
pdf("ufocase.pdf")

ufo <- read.fwf(file="http://www.stat.ufl.edu/~winner/data/ufocase.dat",
  width=c(4,52,16,3,4,4,4),
  col.names=c("Year", "Event", "Loc", "Effect", "Photo", "Contact", "Abduct"))
attach(ufo)

Effect <- factor(Effect,levels=0:1,labels=c("No", "Yes"))
Photo <- factor(Photo,levels=0:1,labels=c("No", "Yes"))
Contact <- factor(Contact,levels=0:1,labels=c("No", "Yes"))
Abduct <- factor(Abduct,levels=0:1,labels=c("No", "Yes"))

table(Effect); table(Photo); table(Contact); table(Abduct);

table(Photo,Contact)

margin.table(table(Photo,Contact),1)      # Row Totals Computed
margin.table(table(Photo,Contact),2)      # Column Totals Computed

table(Photo,Contact)/sum(table(Photo,Contact))

dev.off()
```

Output – UFO Encounters

```
> table(Effect); table(Photo); table(Contact);
table(Abduct);
Effect
 No Yes
 74 129
Photo
 No Yes
141  62
Contact
 No Yes
146  57
Abduct
 No Yes
182  21
>
> table(Photo,Contact);
     Contact
Photo  No Yes
  No  100  41
  Yes  46  16
```

Output – UFO Encounters

```
> margin.table(table(Photo,Contact),1)
# Row Totals Included
Photo
 No Yes
141  62
> margin.table(table(Photo,Contact),2)
# Column Totals Included
Contact
 No Yes
146  57
>
> table(Photo,Contact)/sum(table(Photo,Contact))
     Contact
Photo         No        Yes
  No  0.49261084 0.20197044
  Yes 0.22660099 0.07881773
```

Summarizing a Single Numeric Variable

- Index Plot – Plot of Response against its position in dataset
- Boxplot – Plot Representing Minimum, Lower Quartile, Median, Upper Quartile, Maximum (and possibly outliers – observations more than 1.5*IQR below (above) the lower (upper) quartile
- Histogram – Plot of Frequency across range of values of response (More detail below)
- QQ-Plot – Plot of sample quantiles versus corresponding quantiles of a probability distribution
- Shapiro-Wilk Test – Used to test for normality

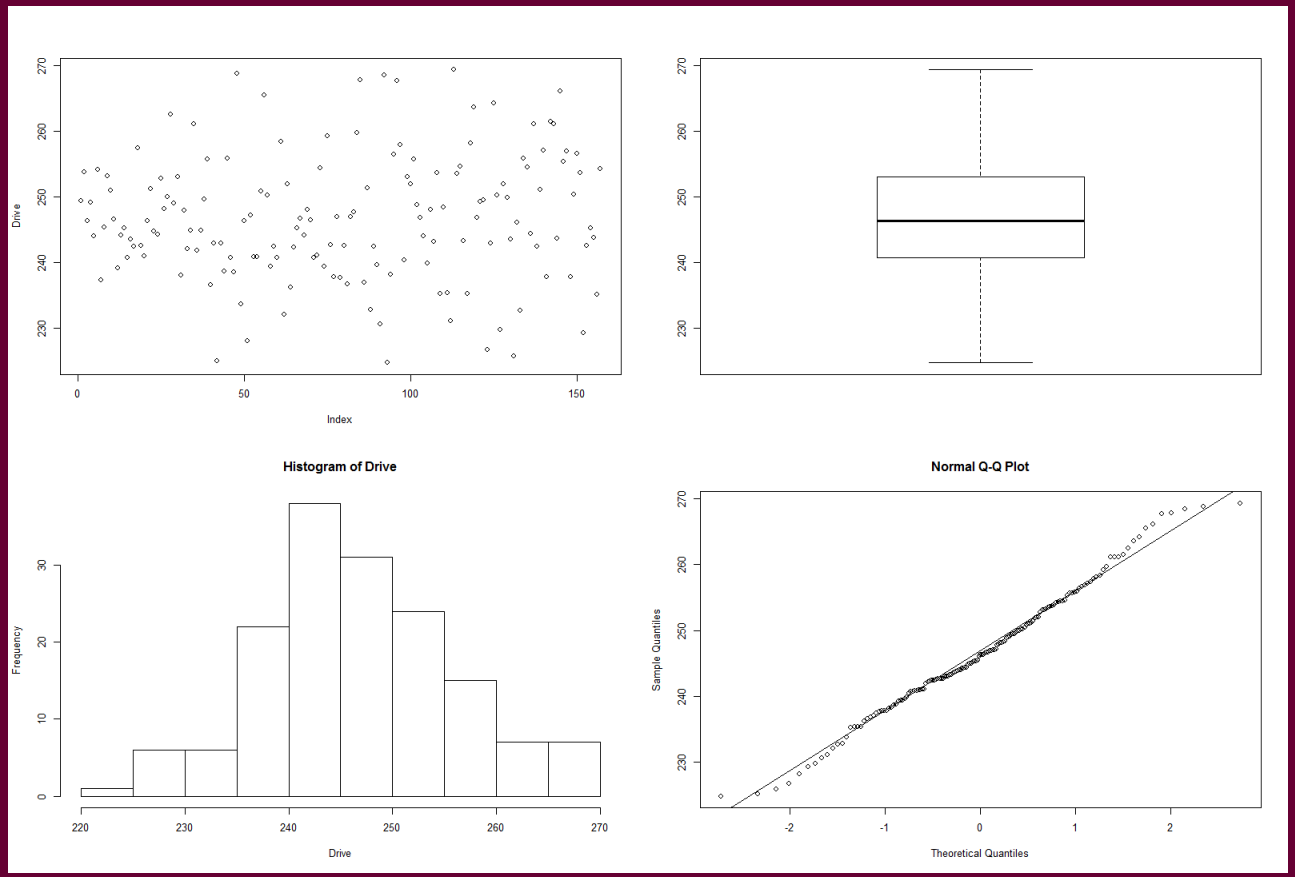
Summarizing LPGA Driving Distances

```
png("lpga2008plots.png")

lpga2008 <- read.fwf("http://www.stat.ufl.edu/~winner/data/lpga2008.dat",
  width=c(30,8,8,8,8,8,8,8,8,8,8),
  col.names=c("Name", "Drive", "Accuracy", "Greens", "AvePutts", "SandAttmpts",
          "PctSandSv", "PrzRnd", "lnPrzRnd", "Rounds", "GolferID"))

attach(lpga2008)

par(mfrow=c(2,2))   # Put plots in 2 rows/cols on single page
plot(Drive)   # Index plot
boxplot(Drive)
hist(Drive)
qqnorm(Drive); qqline(Drive)   # Normal QQ-Plot with straight line added

shapiro.test(Drive)  # Shapiro-Wilk Stat/P-value  H0: Data are Normal

dev.off()
```

LPGA Driving Distance Output

Shapiro-Wilk normality test

data: Drive
W = 0.99, p-value = 0.3362

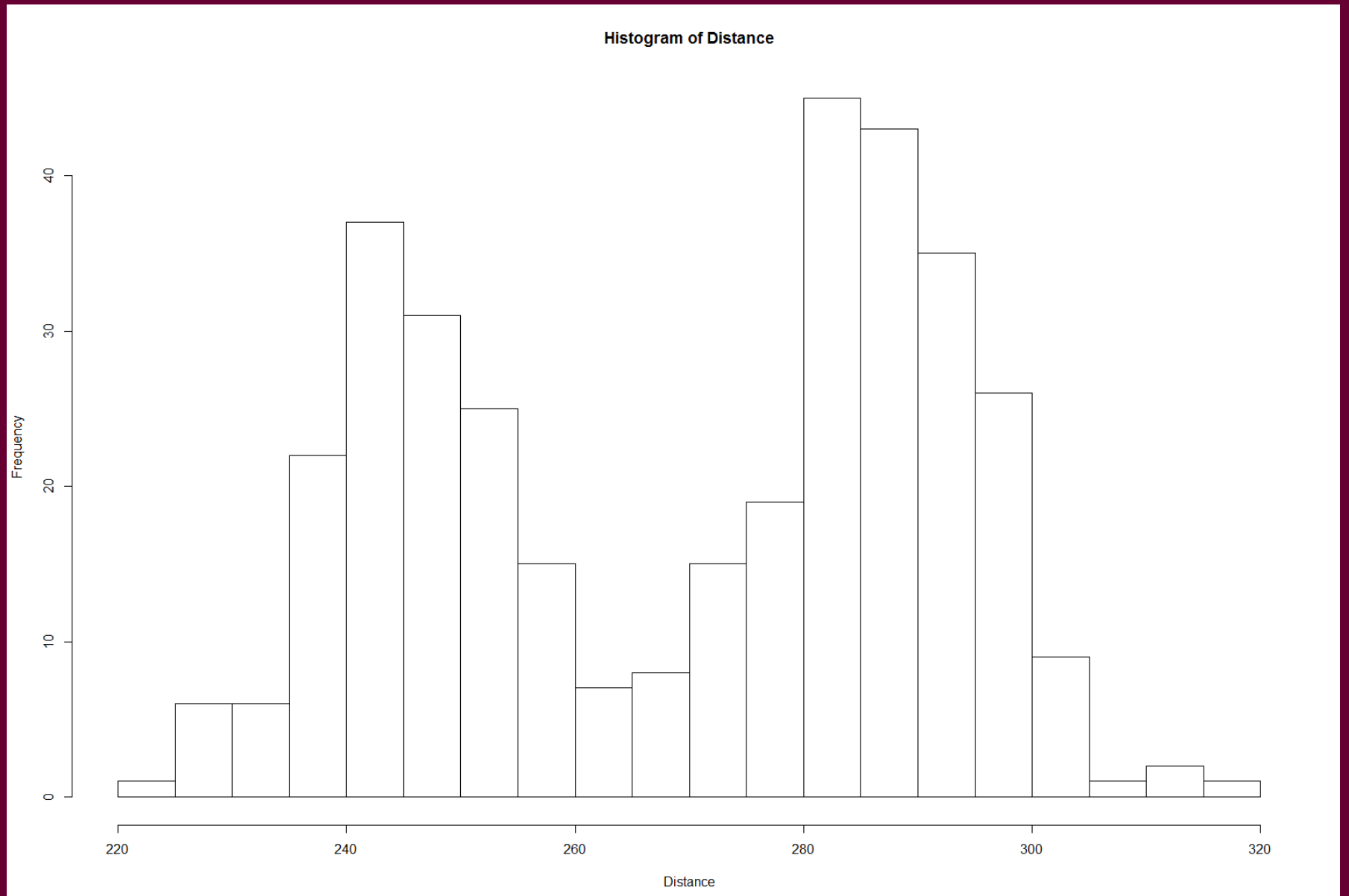


Histograms

- Histograms for Numeric Data (x is the variable):
  - Frequency Histogram (Default # of bins): **hist(x)**
  - Fixing the number of bins: **hist(x, breaks=10)**
  - Fixing the Break Points: **hist(x, breaks=c(5,10,15,20))**
  - Relative Freq.: **hist(x, probability=T)** or **hist(x,freq=F)**
  - Density Curves can be superimposed as well (see example on upcoming slide)
  - Side-by-Side histograms by Group can also be created (see below)

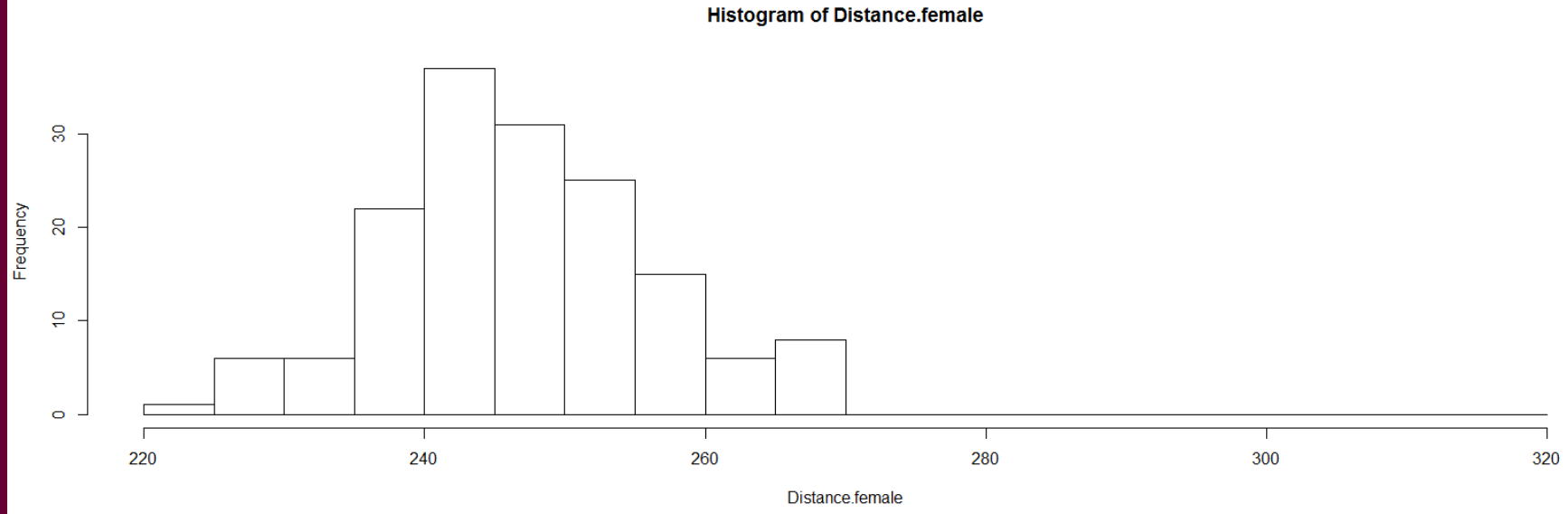
Program – Driving Distance Histogram

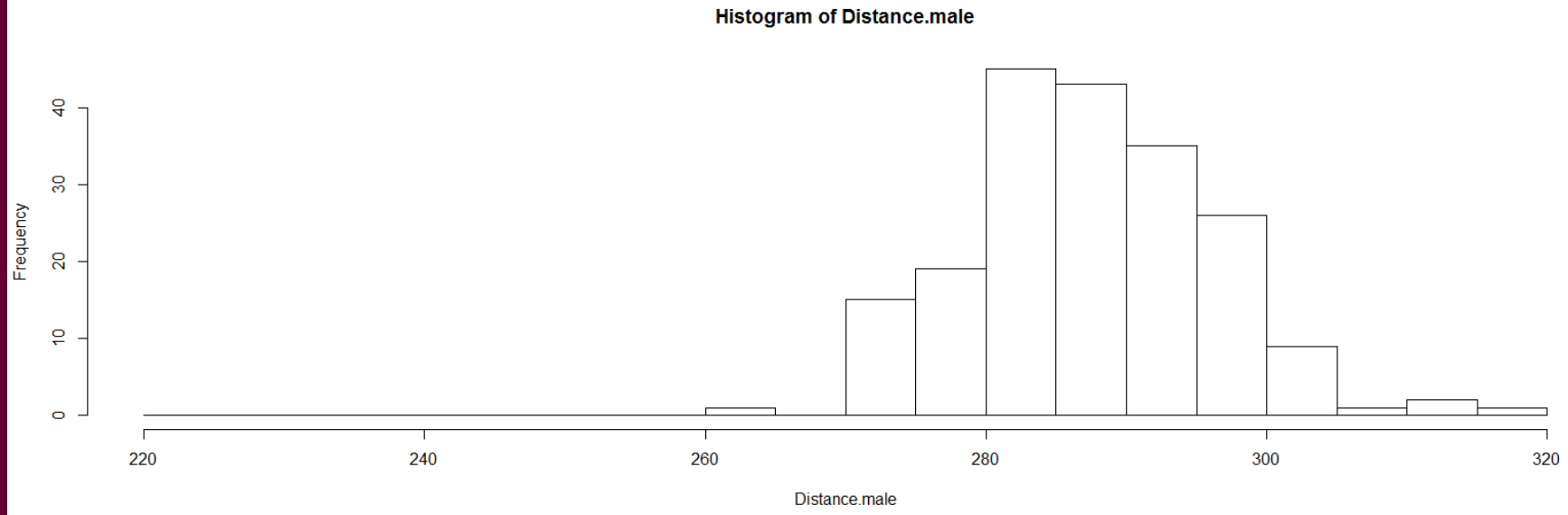
```
pdf("pgalpga1.pdf")



pgalpga <- read.fwf(
        file="http://www.stat.ufl.edu/~winner/data/pgalpga2008.dat",
                width=c(8,8,8),
                col.names=c("Distance", "Accuracy", "Gender"))
attach(pgalpga)

Gender <- factor(Gender, levels=1:2, labels=c("Female", "Male"))

hist(Distance, breaks=seq(220,320,5))

dev.off
```


Histogram of Distance
Frequency
0
10
20
30
40
220
240
260
280
300
320
Distance

Separate by Gender (Same Range)

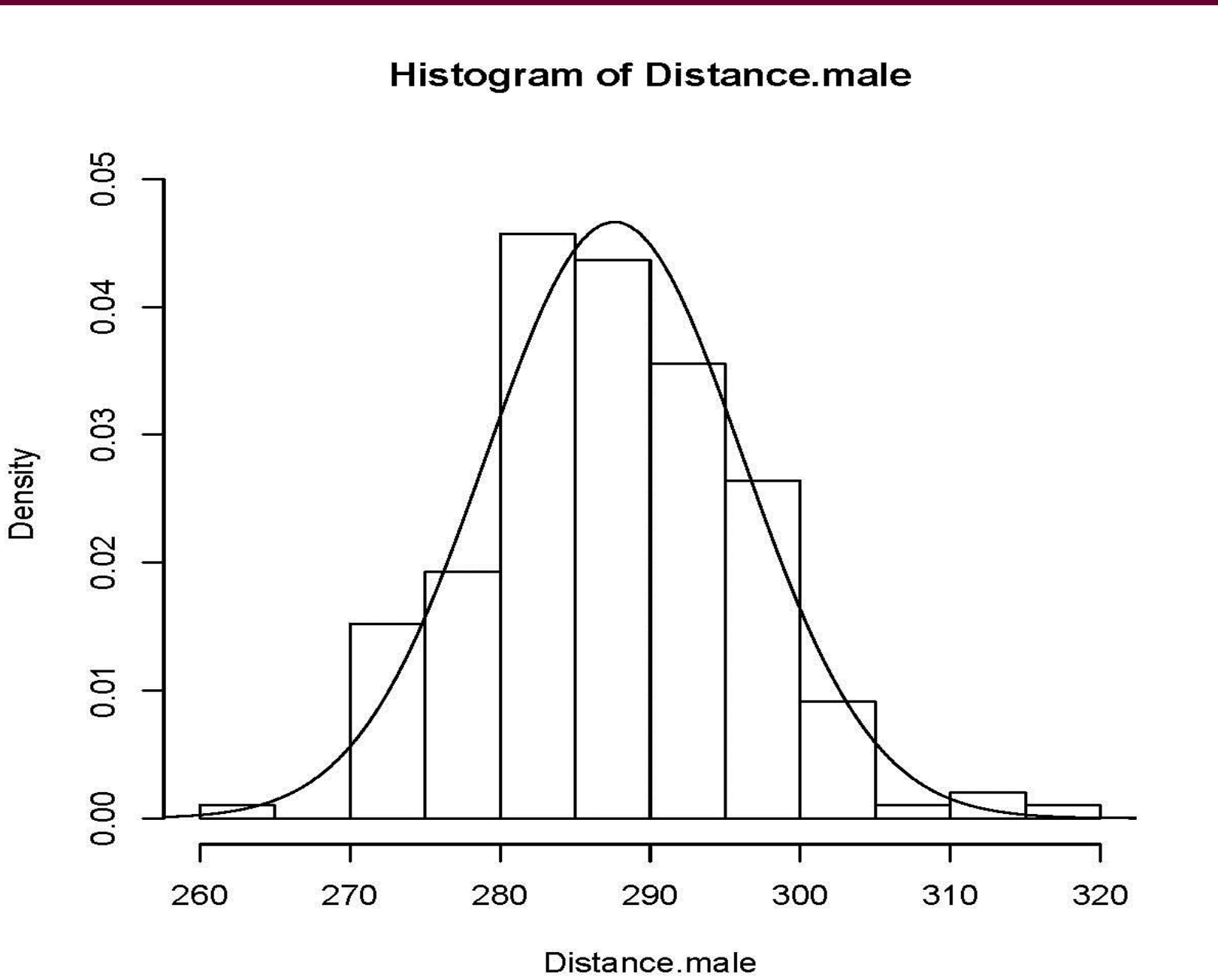
```
pdf("pgalpga2.pdf")
pgalpga <- read.fwf(
        file="http://www.stat.ufl.edu/~winner/data/pgalpga2008.dat",
                width=c(8,8,8),
                col.names=c("Distance", "Accuracy", "Gender"))
attach(pgalpga)
Gender <- factor(Gender, levels=1:2, labels=c("Female", "Male"))

# Create Distance variables seperately by gender
Distance.female <- Distance[Gender=="Female"]
Distance.male <- Distance[Gender=="Male"]

#There will be 2 “rows” and 1 “column” of graphs
# Force each axis to have bins from 220 to 320 by 5 (original scale)
par(mfrow=c(2,1))
hist(Distance.female,breaks=seq(220,320,5))
hist(Distance.male,breaks=seq(220,320,5))
par(mfrow=c(1,1))
dev.off
```


Histogram of Distance.female
Frequency
0
10
20
30
220
240
260
280
300
320
Distance.female


Histogram of Distance.male
Frequency
0
10
20
30
40
220
240
260
280
300
320
Distance.male

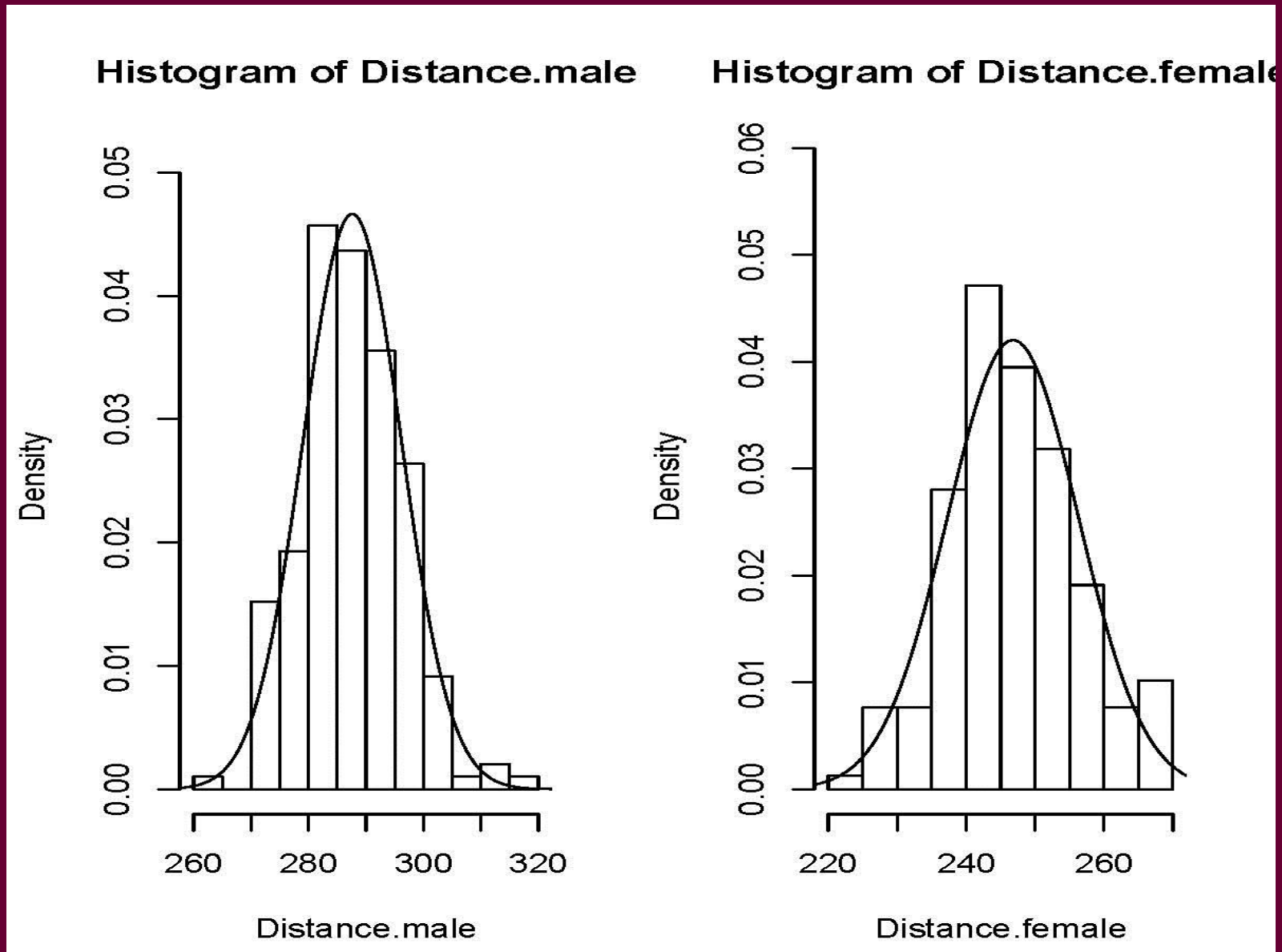
Program – Histogram with Normal Curve

```
pdf("pgalpga3.pdf")
pgalpga <- read.fwf(
        file="http://www.stat.ufl.edu/~winner/data/pgalpga2008.dat",
                width=c(8,8,8),
                col.names=c("Distance", "Accuracy", "Gender"))
attach(pgalpga)
Gender <- factor(Gender, levels=1:2, labels=c("Female", "Male"))
Distance.female <- Distance[Gender=="Female"]
Distance.male <- Distance[Gender=="Male"]

# Histogram for Male Distances
# ylim command "frames" plot area to avoid “cutting off” curve
# Curve command adds a Normal Density with μ=287.61,σ=8.55
h <- hist(Distance.male,plot=F,breaks=seq(260,320,3))
ylim <- range(0,h$density,dnorm(287.61,287.61,8.55))
hist(Distance.male,freq=F,ylim=ylim)
curve(dnorm(x,287.61,8.55),add=T)
dev.off()
```


Histogram of Distance.male
Density
0.00
0.01
0.02
0.03
0.04
0.05
260
270
280
290
300
310
320
Distance.male

Side-by-Side with Normal Curves

```
pdf("pgalpga4.pdf")
pgalpga <- read.fwf(
        file="http://www.stat.ufl.edu/~winner/data/pgalpga2008.dat",
                width=c(8,8,8),
                col.names=c("Distance", "Accuracy", "Gender"))
attach(pgalpga)
Gender <- factor(Gender, levels=1:2, labels=c("Female", "Male"))
Distance.female <- Distance[Gender=="Female"]
Distance.male <- Distance[Gender=="Male"]

par(mfrow=c(1,2))
# Histogram for Male Distances
hm <- hist(Distance.male,plot=F,breaks=seq(260,320,3))
ylim <- range(0,hm$density,dnorm(287.61,287.61,8.55))
hist(Distance.male,freq=F,ylim=ylim)
curve(dnorm(x,287.61,8.55),add=T)

# Histogram for Female Distances
hf <- hist(Distance.female,plot=F,breaks=seq(220,270,2.5))
ylim <- range(0,hf$density,dnorm(246.80,246.80,9.49))
hist(Distance.female,freq=F,ylim=ylim)
curve(dnorm(x,246.80,9.49),add=T)
dev.off()
```


Histogram of Distance.male
Density
0.00
0.01
0.02
0.03
0.04
0.05
260
280
300
320
Distance.male
Histogram of Distance.female
Density
0.00
0.01
0.02
0.03
0.04
0.05
0.06
220
240
260
Distance.female

Barplots and Pie Charts

- Used to show frequencies for categorical variables
- Barplots
  - Simplest version (single variable): **barplot(table(x))**
  - Used to show cross-tabulations: **barplot(table(x,y))**
  - Can be structured for side-by-side or stacked (see below)
- Pie Charts
  - Simplest version (single variable): **pie(table(x))**
  - Used to show cross-tabulations: **pie(table(x,y))**
  - Can be used for Categorical or numeric data

Barplots – UFO Data

```
pdf("ufocase1.pdf")

ufo <- read.fwf(file="http://www.stat.ufl.edu/~winner/data/ufocase.dat",
  width=c(4,52,16,3,4,4,4),
  col.names=c("Year", "Event", "Loc", "Effect", "Photo", "Contact", "Abduct"))
attach(ufo)

Effect <- factor(Effect,levels=0:1,labels=c("No", "Yes"))
Photo <- factor(Photo,levels=0:1,labels=c("No", "Yes"))
Contact <- factor(Contact,levels=0:1,labels=c("No", "Yes"))
Abduct <- factor(Abduct,levels=0:1,labels=c("No", "Yes"))


barplot(table(Photo))
dev.off()
```

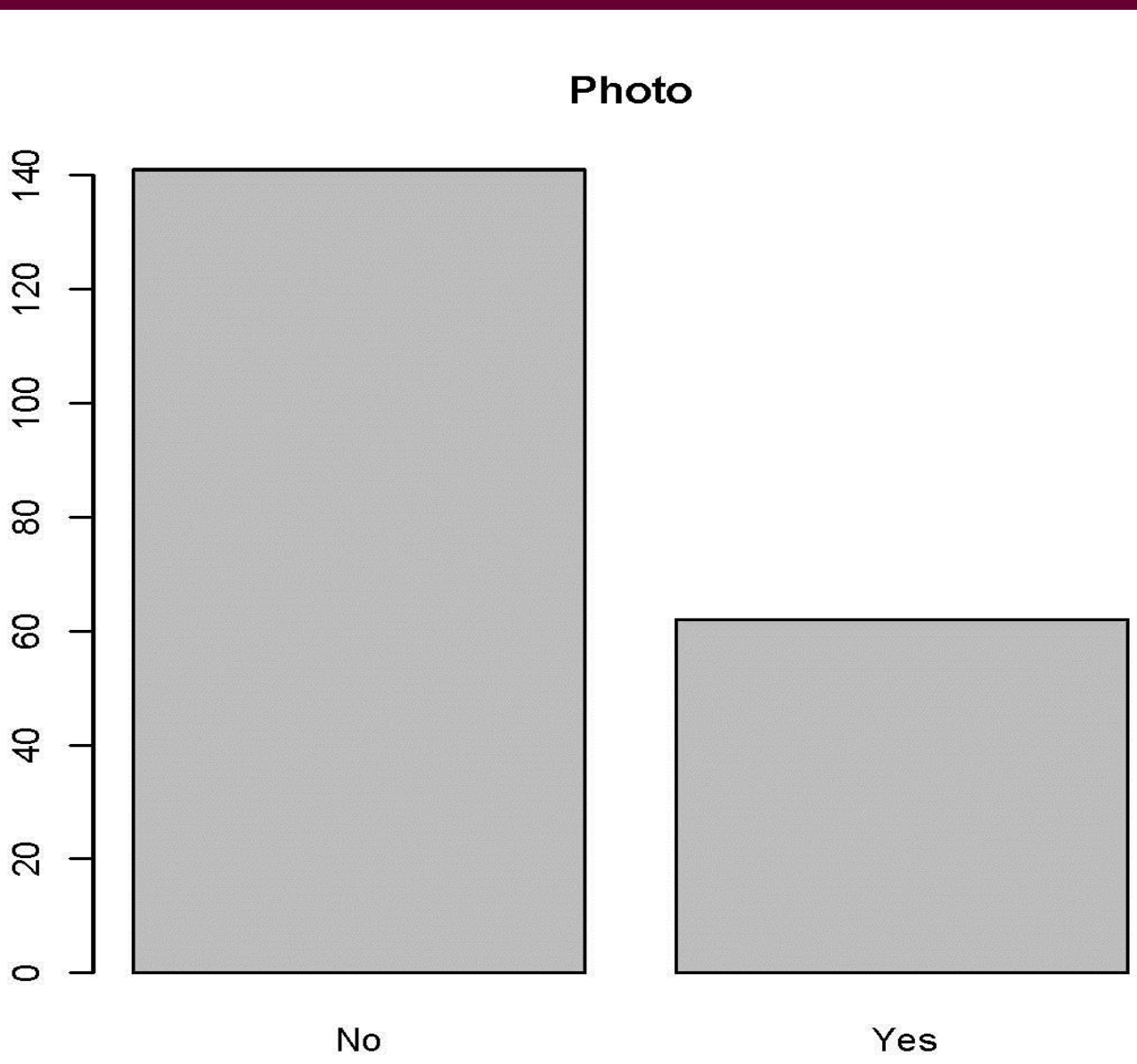
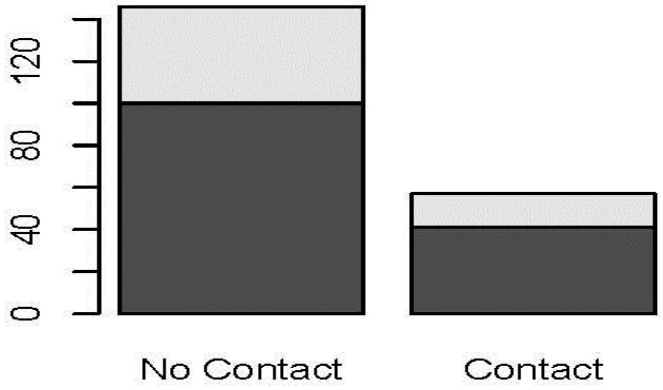
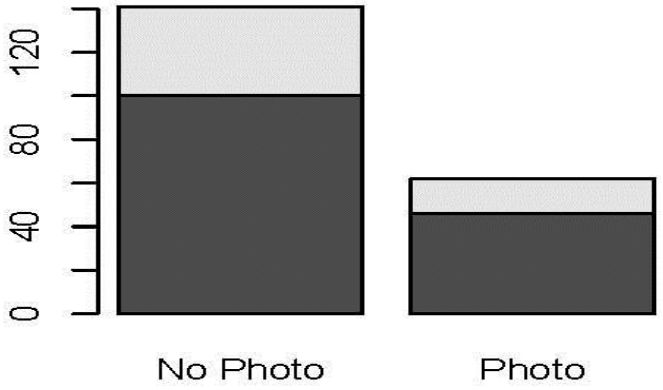


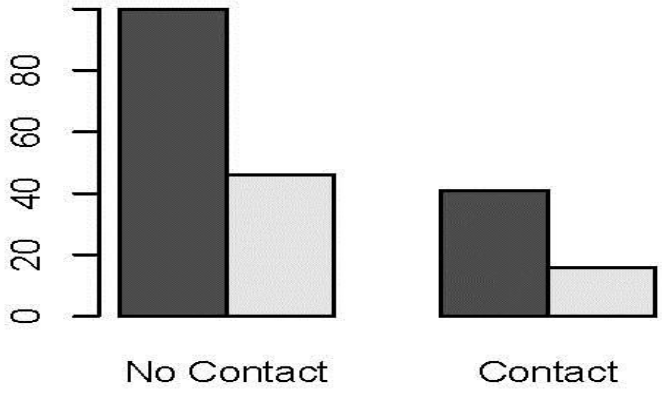
Photo
0
20
40
60
80
100
120
140
No
Yes

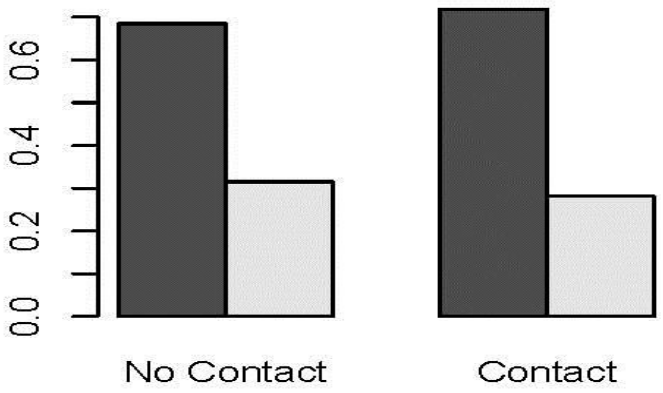
4 Plots of a 2-Way Contingency Table

```
pdf("ufocase2.pdf")
ufo <- read.fwf(file="http://www.stat.ufl.edu/~winner/data/ufocase.dat",
  width=c(4,52,16,3,4,4,4),
  col.names=c("Year", "Event", "Loc", "Effect", "Photo", "Contact", "Abduct"))
attach(ufo)
Effect <- factor(Effect,levels=0:1,labels=c("No Effect", "Effect"))
Photo <- factor(Photo,levels=0:1,labels=c("No Photo", "Photo"))
Contact <- factor(Contact,levels=0:1,labels=c("No Contact", "Contact"))
Abduct <- factor(Abduct,levels=0:1,labels=c("No Abduct", "Abduct"))

pht.cntct <- table(Photo,Contact);
# t(x) transposes the table x (interchanges rows/cols)
#prop.table(pht.cntct,2) obtains the proportions separately within columns
par(mfrow=c(2,2))
barplot(pht.cntct)
barplot(t(pht.cntct))
barplot(pht.cntct,beside=T)
barplot(prop.table(pht.cntct,2),beside=T)
par(mfrow=c(1,1))
dev.off()
```


0
40
80
120
No Contact
Contact


0
40
80
120
No Photo
Photo

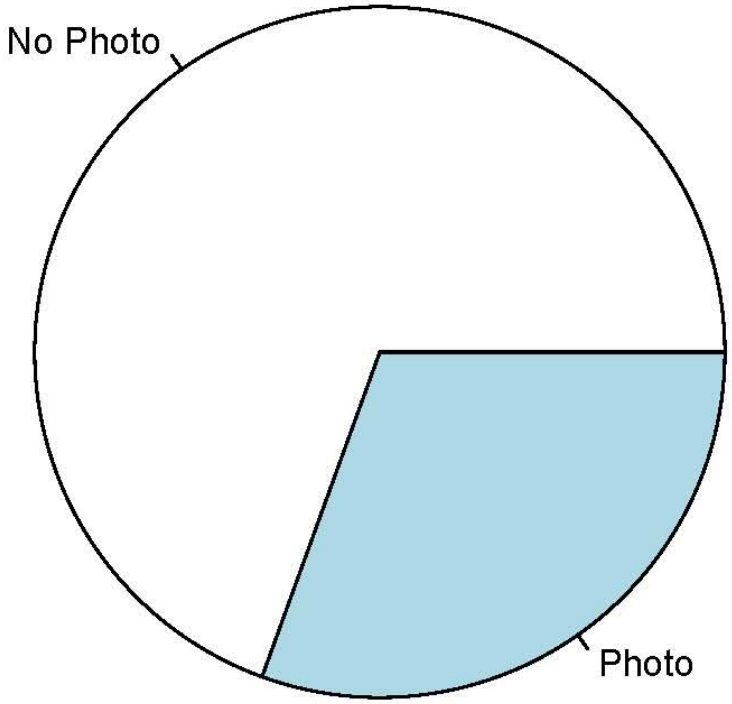

0
20
40
60
80
No Contact
Contact


0.0
0.2
0.4
0.6
No Contact
Contact

Pie Chart – UFO Data

```
pdf("ufocase3.pdf")
ufo <- read.fwf(file="http://www.stat.ufl.edu/~winner/data/ufocase.dat",
  width=c(4,52,16,3,4,4,4),
  col.names=c("Year", "Event", "Loc", "Effect", "Photo", "Contact",
"Abduct"))
attach(ufo)
Effect <- factor(Effect,levels=0:1,labels=c("No Effect", "Effect"))
Photo <- factor(Photo,levels=0:1,labels=c("No Photo", "Photo"))
Contact <- factor(Contact,levels=0:1,labels=c("No Contact", "Contact"))
Abduct <- factor(Abduct,levels=0:1,labels=c("No Abduct", "Abduct"))

table(Effect); table(Photo); table(Contact); table(Abduct);

pie(table(Photo),main="Photo Evidence")
dev.off
```

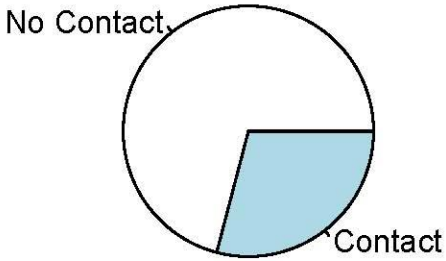
Photo Evidence



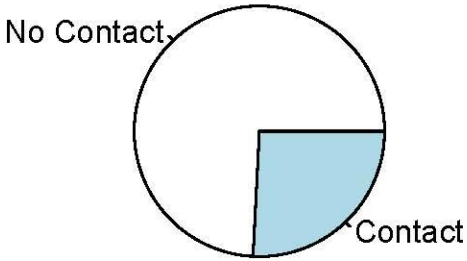
UFO Contact (Column) by Photo (Row) Status

```
pdf("ufocase4.pdf")
ufo <- read.fwf(file="http://www.stat.ufl.edu/~winner/data/ufocase.dat",
  width=c(4,52,16,3,4,4,4),
  col.names=c("Year", "Event", "Loc", "Effect", "Photo", "Contact", "Abduct"))
attach(ufo)
Effect <- factor(Effect,levels=0:1,labels=c("No Effect", "Effect"))
Photo <- factor(Photo,levels=0:1,labels=c("No Photo", "Photo"))
Contact <- factor(Contact,levels=0:1,labels=c("No Contact", "Contact"))
Abduct <- factor(Abduct,levels=0:1,labels=c("No Abduct", "Abduct"))

pht.cntct <- table(Photo,Contact);

par(mfrow=c(1,2))
pie(pht.cntct["No Photo",],main="No Photo")
pie(pht.cntct["Photo",],main="Photo")
par(mfrow=c(1,1))

dev.off()
```

No Photo



Photo

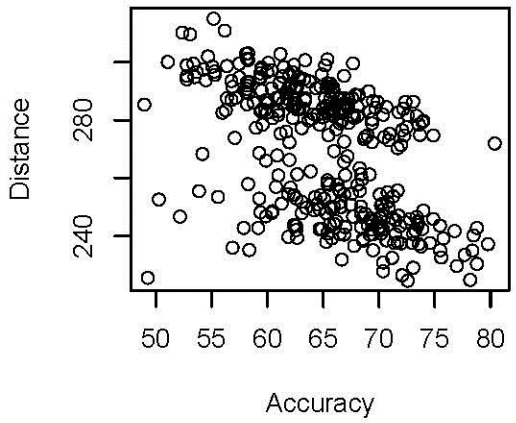


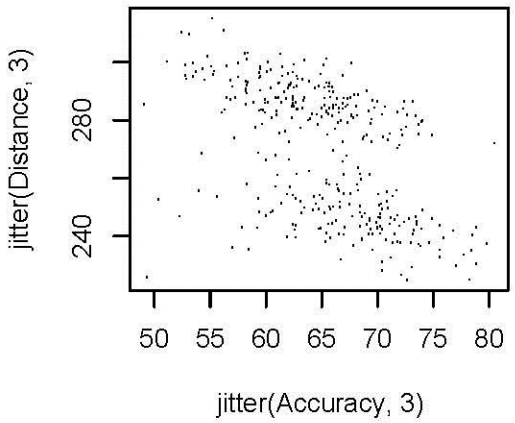
Scatterplots

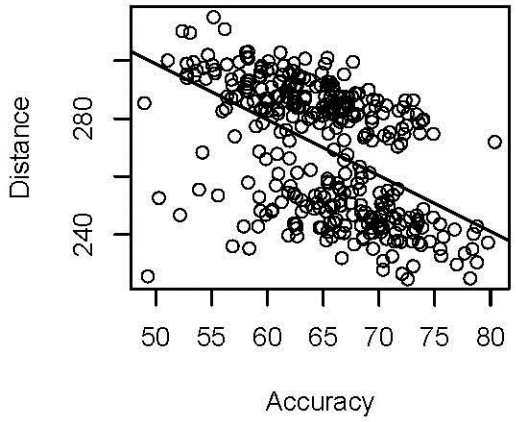
- Plots of 2 (or more) Quantitative Variables
  - For Basic Plot (default symbol is circle): **plot(x,y)**
  - Can change plotting symbol to **l**ines or **b**oth lines and points (appropriate if time series data) **plot(x,y,type="l")** for lines or **plot(x,y,type="b")** for both
  - Can change plotting character to dots: **plot(x,y,pch=".")**
  - Can add "jitter" when data are discrete and have ties: **plot(jitter(x,3),jitter(y,3))** where the number in second position of jitter command reflects amount of jitter
  - Can add lines to plots: **plot(x,y); abline(lm(y~x));**
  - Can do plots by groups of 3rd variable (coplots): **coplot(x~y|z)**

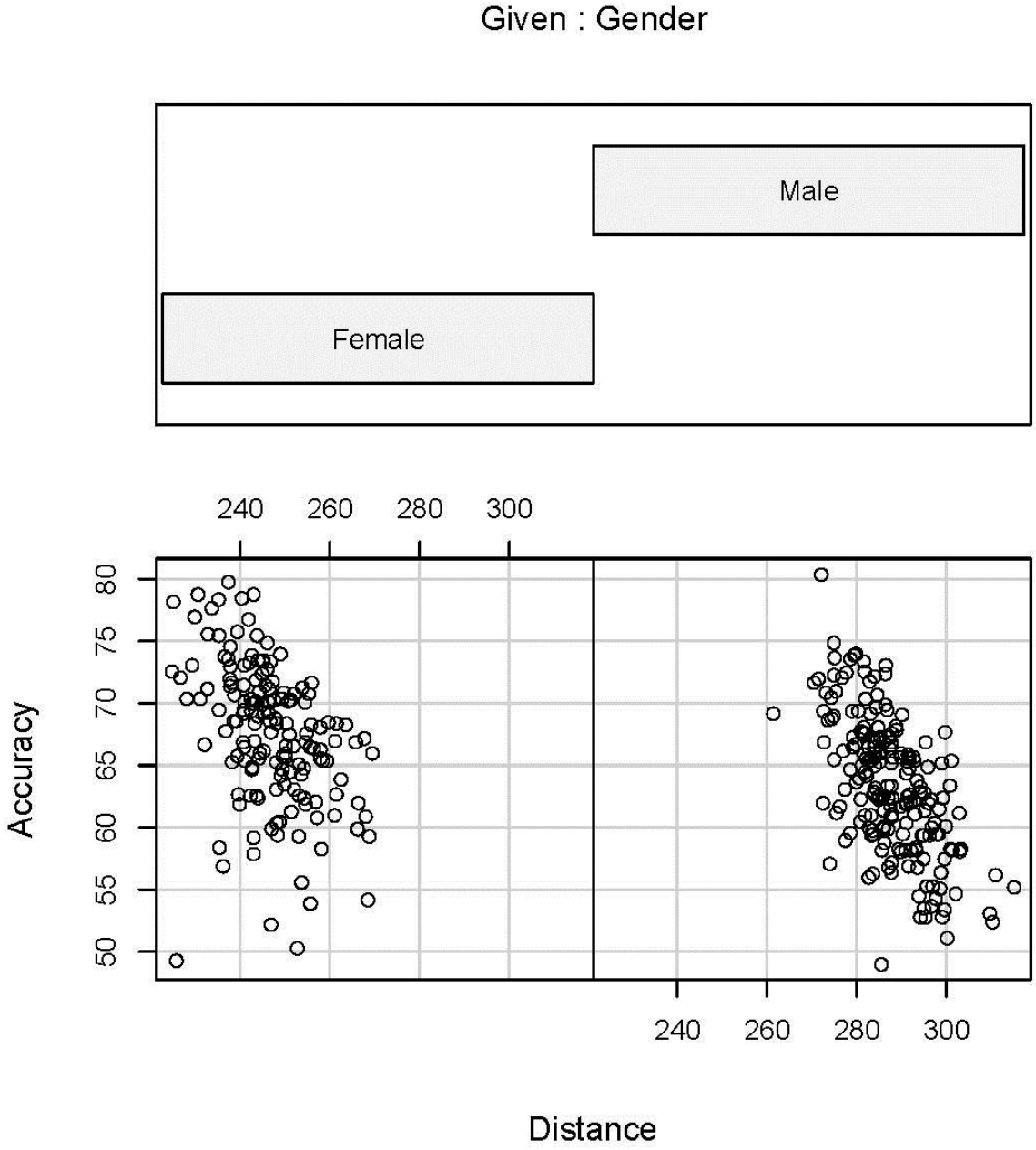
PGA/LPGA Data

```
pdf("pgalpga5.pdf")
pgalpga <- read.fwf(
        file="http://www.stat.ufl.edu/~winner/data/pgalpga2008.dat",
                width=c(8,8,8),
                col.names=c("Distance", "Accuracy", "Gender"))
attach(pgalpga)
Gender <- factor(Gender, levels=1:2, labels=c("Female", "Male"))

par(mfrow=c(2,2))
plot(Accuracy,Distance)

plot(jitter(Accuracy,3),jitter(Distance,3),pch=".")

plot(Accuracy,Distance); abline(lm(Distance~Accuracy))

coplot(Accuracy~Distance|Gender)

dev.off()
```


Distance
240
280
50 55 60 65 70 75 80
Accuracy


jitter(Distance, 3)
240
280
50 55 60 65 70 75 80
jitter(Accuracy, 3)


Distance
240
280
50 55 60 65 70 75 80
Accuracy


Given : Gender
Male
Female
240
260
280
300
80
75
70
65
60
55
50
Accuracy
240
260
280
300
Distance

Comparing 2 Populations

- Sampling Designs: Independent/Dependent
- Distributional Assumptions: Normal/Non-Normal
  - Means: Independent Samples/Normal Distributions
    - Student's t-test (Equal and Unequal Variance cases)
  - Medians: Independent Samples/Non-Normal Distributions
    - Wilcoxon Rank-Sum Test
  - Means: Dependent Samples/Normal Distributions
    - Paired t-test
  - Medians: Dependent Samples/Non-Normal Distributions
    - Wilcoxon Signed-Rank Test
  - Variances: Independent Samples/Normal Distributions
    - F-test

Program -Testing for Accuracy by Gender

```
pdf("pgalpga5.pdf")
pgalpga <- read.fwf(
        file="http://www.stat.ufl.edu/~winner/data/pgalpga2008.dat",
                width=c(8,8,8),
                col.names=c("Distance", "Accuracy", "Gender"))
attach(pgalpga)
Gender <- factor(Gender, levels=1:2, labels=c("Female", "Male"))
Distance.female <- Distance[Gender=="Female"]
Distance.male <- Distance[Gender=="Male"]
Accuracy.female <- Accuracy[Gender=="Female"]
Accuracy.male <- Accuracy[Gender=="Male"]

# 2 ways of getting t-test: As a “Model” and as 2 Groups of Responses
# var.test tests the equal variance assumption
var.test(Accuracy ~ Gender)
t.test(Accuracy ~ Gender,var.equal=T)

t.test(Accuracy.female,Accuracy.male)

wilcox.test(Accuracy ~ Gender)
dev.off()
```

Output -Testing for Accuracy by Gender

```
> var.test(Accuracy ~ Gender)

        F test to compare two variances
data:  Accuracy by Gender
F = 1.1158, num df = 156, denom df = 196, p-value = 0.4664
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.8300783 1.5076372
sample estimates:
ratio of variances
          1.115823

> t.test(Accuracy ~ Gender, var.equal=T)
        Two Sample t-test

data:  Accuracy by Gender
t = 7.0546, df = 352, p-value = 9.239e-12
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 3.047924 5.404292
sample estimates:
mean in group Female   mean in group Male
            67.59108             63.36497
```

Output -Testing for Accuracy by Gender

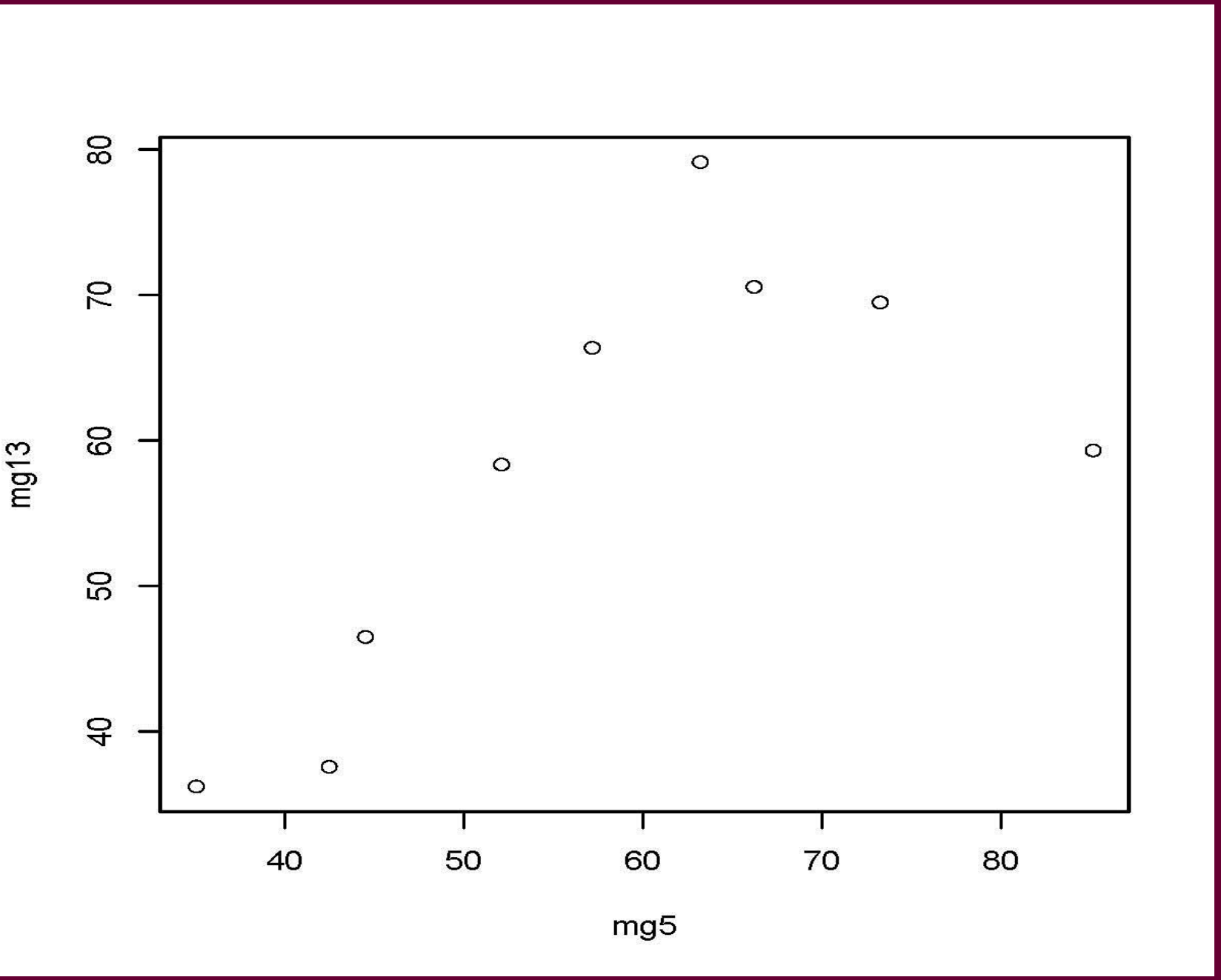
```
> t.test(Accuracy.female,Accuracy.male)

        Welch Two Sample t-test

data:  Accuracy.female and Accuracy.male
t = 7.011, df = 326.041, p-value = 1.369e-11
alternative hypothesis: true difference in means is not
equal to 0
95 percent confidence interval:
 3.040267 5.411949
sample estimates:
mean of x mean of y
 67.59108  63.36497

> wilcox.test(Accuracy ~ Gender)

        Wilcoxon rank sum test with continuity correction

data:  Accuracy by Gender
W = 22016.5, p-value = 7.417e-12
alternative hypothesis: true mu is not equal to 0
```

Program – Caffeine/Endurance – Paired Data

```
pdf("caffeine.pdf")

caffeine <-
read.fwf("http://www.stat.ufl.edu/~winner/data/caffeine1.dat",
width=c(8,8,8,8,8),
col.names=c("subject", "mg0", "mg5", "mg9", "mg13"))

attach(caffeine)

# Plot of 13mg endurance versus 5mg
#    endurance for the 9 cyclists
plot(mg5,mg13)


# Conduct Paired t-test to determine if Dose effect exists
t.test(mg13, mg5, paired=TRUE)

# Conduct Wilcoxon Signed-Rank Test
wilcox.test(mg13, mg5, paired=TRUE)

dev.off()
```

1	37.55	42.47
2	59.30	85.15
3	79.12	63.20
4	58.33	52.10
5	70.54	66.20
6	69.47	73.25
7	46.48	44.50
8	66.35	57.17
9	36.20	35.05

Data


mg13
40
50
60
70
80
40
50
60
70
80
mg5

Output – Caffeine/Endurance – Paired Data

```
        Paired t-test

data:  mg13 and mg5
t = 0.1205, df = 8, p-value = 0.907
alternative hypothesis: true difference in means is not equal
to 0
95 percent confidence interval:
 -8.562982  9.507426
sample estimates:
mean of the differences
              0.4722222


>
> # Conduct Wilcoxon Signed-Rank Test
> wilcox.test(mg13, mg5, paired=TRUE)

        Wilcoxon signed rank test

data:  mg13 and mg5
V = 28, p-value = 0.5703
alternative hypothesis: true mu is not equal to 0
```

Pearson Chi-Square Test

- Used to Test for Association Between 2 or More Categorical Variables
- Arranges Counts of Outcomes in Contingency Table (Cross-Tab)
- Null Hypothesis: Distribution of Column (Row) Proportions is Constant Across Rows (Columns)
- Data Can be Entered Directly into Contingency Table or Table Formed From Raw Data at the Individual Level
- fisher.test(table) Can be used for small samples (exact for 2x2 tables)

Case 1 – Direct Entry of Contingency Table

Diet\Complete 1 Year	Yes	No
Atkins	21	19
Zone	26	14
Weight Watchers	26	14
Ornish	20	20

```
pdf("dietcomp.pdf")

# Create Table as string of 8 counts, tell R it's entered by rows
#  (as opposed to columns) and tell R there are 4 rows (thus 2 columns)

diet.table <- matrix(c(21,19,26,14,26,14,20,20),byrow=T,nrow=4)

diet.table

# Conduct Pearson's chi-square test
chisq.test(diet.table)

# Conduct Fisher's exact test
fisher.test(diet.table)



dev.off()
```

Case 1 – Direct Entry of Contingency Table

```
> # Conduct Pearson's chi-square test
> chisq.test(diet.table)

        Pearson's Chi-squared test

data:  diet.table
X-squared = 3.1584, df = 3, p-value = 0.3678

>
> # Conduct Fisher's exact test
> fisher.test(diet.table)

        Fisher's Exact Test for Count Data

data:  diet.table
p-value = 0.3904
alternative hypothesis: two.sided
```

Case 2 – Raw Data

Atkins	C
Atkins	C
…	…
Atkins	F
Atkins	F
Zone	C
Zone	C
…	…
Zone	F
Zone	F
WW	C
WW	C
…	…
WW	F
WW	F
Ornish	C
Ornish	C
…	…
Ornish	F
Ornish	F

160Subjects

40 per diet

```
pdf("rawdiet.pdf")

rawdiet <- read.fwf("http://www.stat.ufl.edu/~winner/data/rawdiet.dat",
  width=c(8,8), col.names=c("diet", "comp1yr"))
attach(rawdiet)

diet <- factor(diet,levels=1:4,labels=c("A", "Z", "W", "O"))
comp1yr <- factor(comp1yr,levels=0:1,labels=c("Fail", "Comp"))

dietcomp <- table(diet,comp1yr)
dietcomp

# Conditional Distributions of Outcome by Diet (Rows)
#  The “1” gets row distributions, “2” would give columns

prop.table(dietcomp,1)

# Pearson Chi-Square Test

chisq.test(dietcomp)

dev.off()
```

Case 2 – Raw Data

```
> dietcomp
    complyr
diet Fail Comp
   A   19   21
   Z   14   26
   W   14   26
   O   20   20
> # Conditional Distributions of Outcome by Diet (Rows)
> prop.table(dietcomp,1)
    complyr
diet  Fail  Comp
   A 0.475 0.525
   Z 0.350 0.650
   W 0.350 0.650
   O 0.500 0.500
> # Pearson Chi-Square Test
> chisq.test(dietcomp)
        Pearson's Chi-squared test
data:  dietcomp
X-squared = 3.1584, df = 3, p-value = 0.3678
```

Comparison of k>2 Means

- Sampling Designs:
  - Independent – Completely Randomized Design
  - Dependent – Randomized Block Design
- Distributional Assumptions: Normal/Non-Normal
  - Independent Samples/Normal Distributions
    - ANOVA F-Test for CRD
  - Independent Samples/Non-Normal Distributions
    - Kruskal-Wallis Test
  - Dependent Samples/Normal Distributions
    - ANOVA F-Test for RBD
  - Dependent Samples/Non-Normal Distributions
    - Friedman's Test
- Post-Hoc Comparison's (Normal Data)
  - Bonferroni's Method
  - Tukey's Method

Program – Amoebae Innoculi - F-test, Post-Hoc Tests

```
amoeba <- read.fwf(
file="http://www.stat.ufl.edu/~winner/data/entozamoeba.dat",
width=c(8,8), col.names=c("Trt", "Yield"))
attach(amoeba)

fTrt <- factor(method, levels=1:5,
 labels=c("Control", "Heat", "Form", "HF", "FH"))

# Obtain the 1-Way ANOVA using aov, not lm to obtain Tukey's HSD
amoeba1.aov <- aov(Yield~fTrt)
summary(amoeba1.aov)

# Obtain Tukey's Comparisons among levels of treatment
TukeyHSD(amoeba1.aov, "fTrt")

# Obtain Bonferroni Comparisons among levels of treatment
# Does NOT use pooled Mean Square Error Across all Treatments
pairwise.t.test(Yield, fTrt, p.adj="bonf")
dev.off()
```

Output – 5 Amoebae Innoculi – F-Test

```
> amoeba1.aov <- aov(Yield~fTrt)
>
> summary(amoeba1.aov)
            Df Sum Sq Mean Sq F value   Pr(>F)
fTrt         4  19666    4916  5.0444 0.001911 **
Residuals   45  43858     975
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
0.1 ' ' 1
>
>
> # Obtain Tukey's Comparisons among levels of
treatment
>
> TukeyHSD(amoeba1.aov, "fTrt")
  Tukey multiple comparisons of means
    95% family-wise confidence level
```

Output Continued on next slide

Output – 5 Amoebae Innoculi – Post-Hoc Tests

```
Fit: aov(Yield ~ fTrt)

$fTrt
          diff       lwr         upr     p adj
H-None   -42.9 -82.57118  -3.228817 0.0281550
F-None   -49.5 -89.17118  -9.828817 0.0078618
HF-None  -29.9 -69.57118   9.771183 0.2208842
FH-None  -56.1 -95.77118 -16.428817 0.0019658
F-H       -6.6 -46.27118  33.071183 0.9894377
HF-H      13.0 -26.67118  52.671183 0.8832844
FH-H     -13.2 -52.87118  26.471183 0.8774716
HF-F      19.6 -20.07118  59.271183 0.6284290
FH-F      -6.6 -46.27118  33.071183 0.9894377
FH-HF    -26.2 -65.87118  13.471183 0.3444453

 pairwise.t.test(Yield, fTrt, p.adj="bonf")
Pairwise comparisons using t tests with pooled SD
data:  Yield and fTrt
   None   H      F      HF
H  0.0360 -      -      -
F  0.0093 1.0000 -      -
HF 0.3768 1.0000 1.0000 -
FH 0.0022 1.0000 1.0000 0.6707
P value adjustment method: bonferroni
```

Program/Output – Kruskal-Wallis Test

```
amoeba <- read.fwf(
file="http://www.stat.ufl.edu/~winner/data/entozamoeba.dat",
width=c(8,8), col.names=c("Trt", "Yield"))
attach(amoeba)

fTrt <- factor(method, levels=1:5,
 labels=c("Control", "Heat", "Form", "HF", "FH"))

# Conduct the Kruskal-Wallis Test (Rank-Based)
kruskal.test(Yield~fTrt)
```

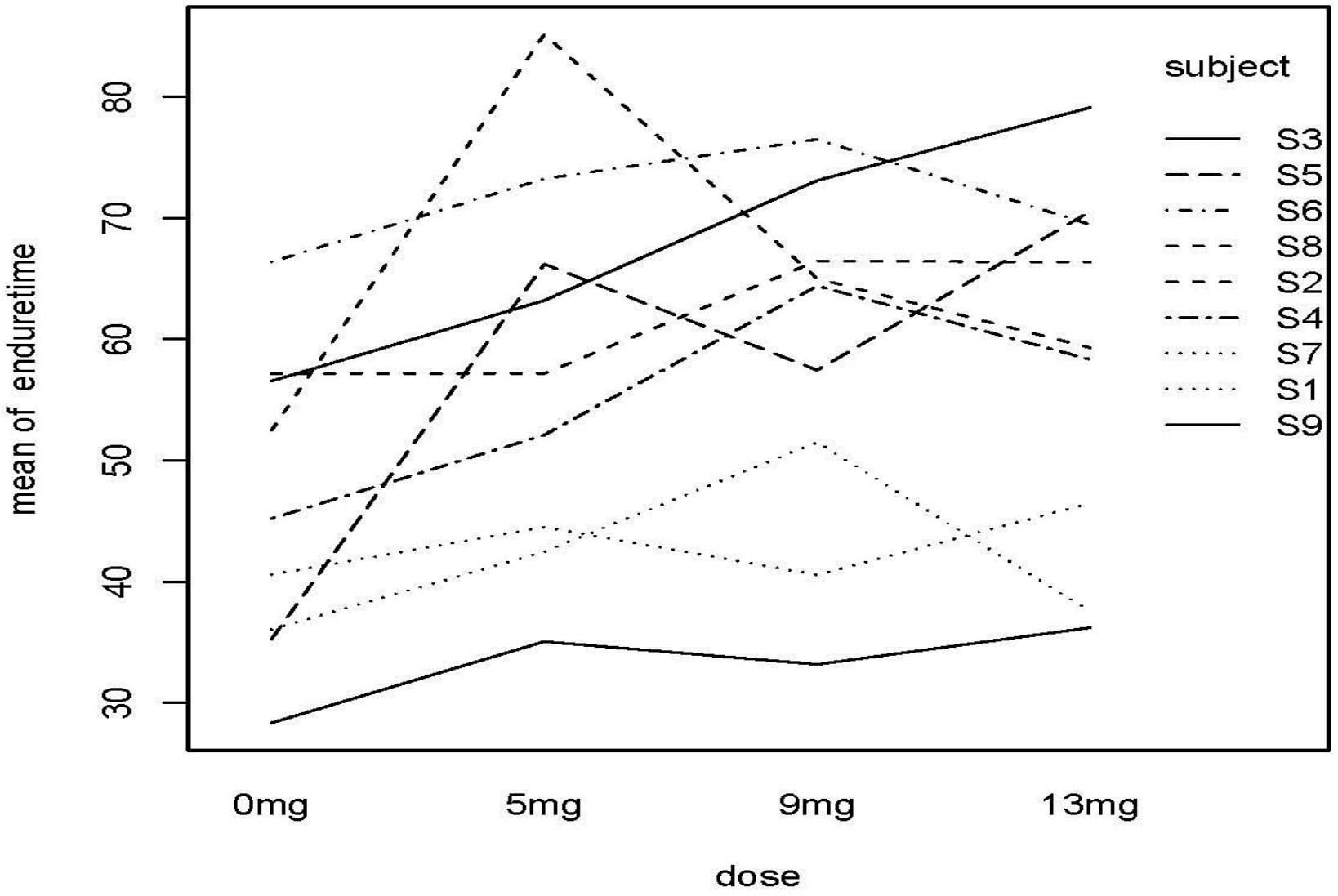
```
> kruskal.test(Yield~fTrt)

        Kruskal-Wallis rank sum test

data:  Yield by fTrt
Kruskal-Wallis chi-squared = 12.6894,
df = 4, p-value = 0.01290
```

RBD Program – Caffeine and Endurance

```
pdf("caffrbd.pdf")
caffrbd <- read.fwf("http://www.stat.ufl.edu/~winner/data/caffeine.dat",
width=c(1,5,8),  col.names=c("subject", "dose", "enduretime"))
attach(caffrbd)
# Create Factors from subject and dose
subject <- factor(subject, levels=1:9,
 labels=c("S1", "S2", "S3", "S4", "S5", "S6", "S7", "S8", "S9"))
dose <- factor(dose, levels=c(0,5,9,13),
labels=c("0mg", "5mg", "9mg", "13mg"))

# Plot of endurance versus dose separately for the 9 cyclists
# The order of vars is: X-Variable, Separate Line Factor, Y-Variable
interaction.plot(dose, subject, enduretime)

# Obtain Analysis of Variance for RBD
caffeine.aov <- aov(enduretime ~ dose + subject)
summary(caffeine.aov)

# Conduct Tukey's HSD for Doses
TukeyHSD(caffeine.aov, "dose")
```

Output – Caffeine and Endurance

```
> caffeine.aov <- aov(enduretime ~ dose + subject)
> summary(caffeine.aov)
            Df Sum Sq Mean Sq F value    Pr(>F)
dose         3  933.1   311.0  5.9168  0.003591 **
subject      8 5558.0   694.7 13.2159 4.174e-07 ***
Residuals   24 1261.7    52.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> # Conduct Tukey's HSD for Doses
> TukeyHSD(caffeine.aov, "dose")
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = enduretime ~ dose + subject)
$dose
               diff       lwr       upr     p adj
5mg-0mg  11.2366667  1.808030 20.665303 0.0153292
9mg-0mg  12.2411111  2.812474 21.669748 0.0076616
13mg-0mg 11.7088889  2.280252 21.137526 0.0110929
9mg-5mg   1.0044444 -8.424192 10.433081 0.9909369
13mg-5mg  0.4722222 -8.956414  9.900859 0.9990313
13mg-9mg -0.5322222 -9.960859  8.896414 0.9986162
```


mean of enduretime
30
40
50
60
70
80
0mg
5mg
9mg
13mg
dose
subject
S3
S5
S6
S8
S2
S4
S7
S1
S9

Program/Output for Friedman's Test

```
pdf("caffrbd.pdf")
caffrbd <- read.fwf(" http://www.stat.ufl.edu/~winner/data/caffeine.dat ",
width=c(1,5,8), col.names=c("subject","dose","enduretime"))
attach(caffrbd)
subject <- factor(subject, levels=1:9,
 labels=c("S1","S2","S3","S4","S5","S6","S7","S8","S9"))
dose <- factor(dose, levels=c(0,5,9,13),
labels=c("0mg", "5mg", "9mg", "13mg"))

# Conduct Friedman's Test (DepVar ~ Trt | Block)
friedman.test(enduretime ~ dose | subject)
```

```
> friedman.test(enduretime ~ dose | subject)

        Friedman rank sum test

data:  enduretime and dose and subject
Friedman chi-squared = 14.2, df = 3, p-value = 0.002645
```

Multi-Factor ANOVA

- Used to Measure Main Effects and Interactions among Several Factors Simultaneously
- Can be Conducted Within CRD or RBD frameworks
- Can Make Post-Hoc Comparisons among Main Effects of Factor Levels (Assuming Interaction is Not Present
- Factors can be Crossed or Nested
  - Crossed → Levels of One Factor Observed at all levels of other factor(s)
  - Nested → Levels of One Factor Are Different Within Various levels of other factor(s)

2-Way ANOVA (CRD) – Lacrosse Helmets

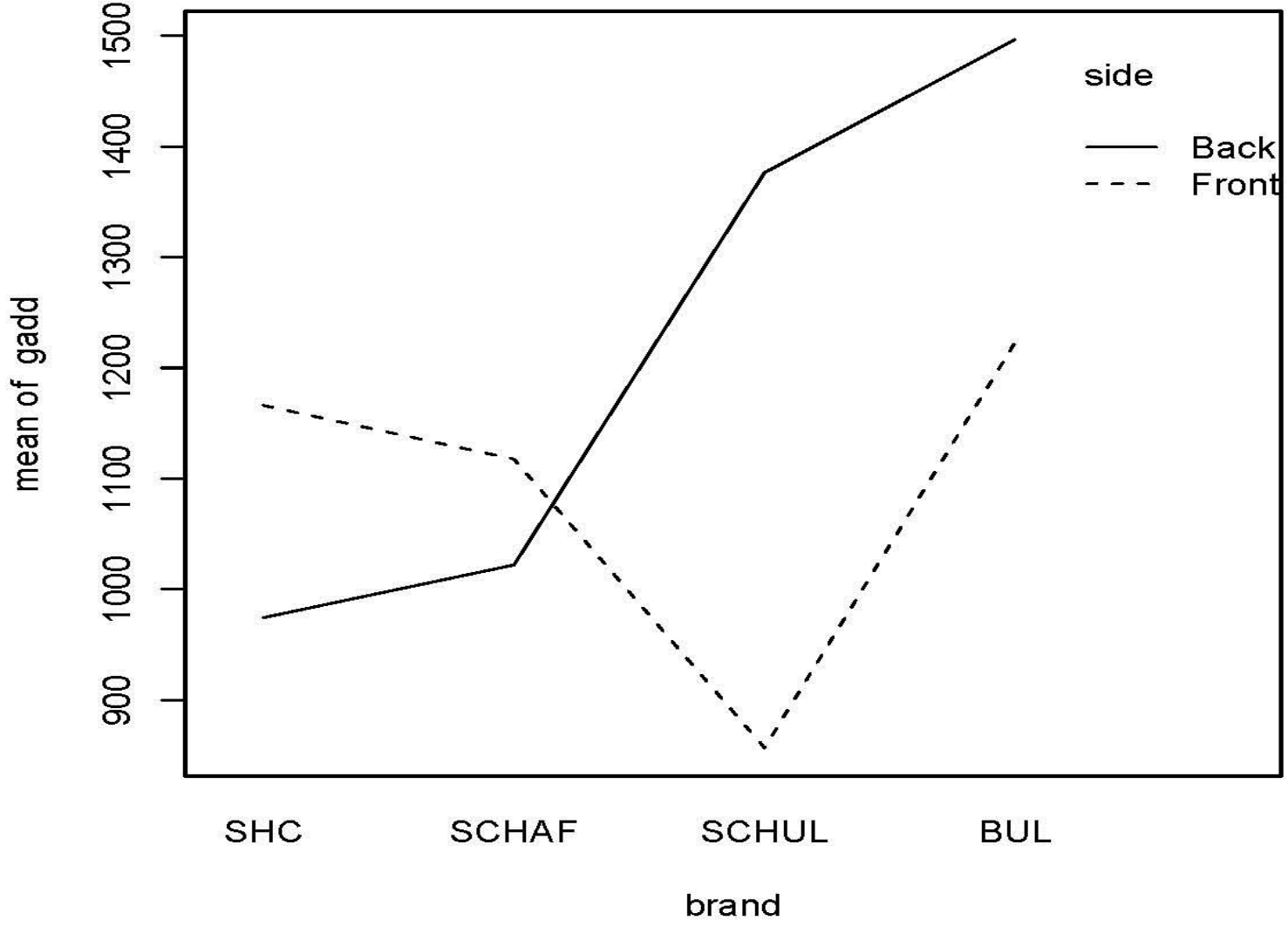
```
pdf("lacrosse.pdf")
lacrosse1 <- read.fwf("http://www.stat.ufl.edu/~winner/data/lacrosse.dat",
width=c(8,8,14), col.names=c("brand", "side", "gadd"))
attach(lacrosse1)

brand <- factor(brand, levels=1:4,
labels=c("SHC", "SCHAF", "SCHUL", "BUL"))
side <- factor(side, levels=1:2,
labels=c("Front", "Back"))

# Descriptive Statistics
tapply(gadd, brand, mean) # marginal mean for brand
tapply(gadd, side, mean)  # marginal mean for side
tapply(gadd, list(brand,side), mean) # cell means
tapply(gadd, list(brand,side), sd)   # cell SDs

# Fit Model with Interaction (A*B says to include main effects and interactions)
lacrosse1.aov <- anova(lm(gadd ~ brand*side))
lacrosse1.aov

# Plot Means of Front and Back versus Brand
interaction.plot(brand,side,gadd)
```

Output – Part 1

```
> tapply(gadd, brand, mean) # marginal mean for brand
     SHC    SCHAF    SCHUL      BUL
1070.300 1069.850 1116.650 1359.650
>
> tapply(gadd, side, mean)  # marginal mean for side
   Front     Back
1090.875 1217.350
>
> tapply(gadd, list(brand,side), mean) # cell means
          Front      Back
SHC   1166.0999  974.4998
SCHAF 1117.5999 1022.1000
SCHUL  857.0001 1376.3000
BUL   1222.8001 1496.5001
>
> tapply(gadd, list(brand,side), sd)   # cell SDs
         Front      Back
SHC   152.3998  72.99994
SCHAF 216.2000 105.09994
SCHUL 151.4998 211.40010
BUL   123.1000 183.99987
```

Output – Part 2

```
> # Fit Model with Interaction
> # (A*B says to include main effects and interactions)
>
> lacrosse1.aov <- anova(lm(gadd ~ brand*side))
>
> lacrosse1.aov
Analysis of Variance Table

Response: gadd
            Df  Sum Sq Mean Sq F value    Pr(>F)
brand        3 1155478  385159  15.179 9.459e-08 ***
side         1  319918  319918  12.608 0.0006816 ***
brand:side   3 1632156  544052  21.441 4.988e-10 ***
Residuals   72 1826954   25374
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


mean of gadd
900
1000
1100
1200
1300
1400
1500
side
Back
Front
SHC
SCHAF
SCHUL
BUL
brand

2-Way ANOVA (RBD) Fabric Hairiness

```
pdf("hairiness.pdf")
hair1 <- read.fwf("http://www.stat.ufl.edu/~winner/data/hairyarn.dat",
width=c(8,8,8,8), col.names=c("twstlvl", "tstspd", "bobbin", "hairiness"))
attach(hair1)
twstlvl <- factor(twstlvl, levels=1:3,labels=c("373tpm", "563tpm", "665tpm"))
tstspd <- factor(tstspd, levels=1:3,labels=c("25mpm", "100mpm", "400mpm"))
bobbin <- factor(bobbin, levels=1:6)

tapply(hairiness, bobbin, mean)                  # marginal mean for bobbin
tapply(hairiness, list(twstlvl,tstspd), mean)     # Trt cell means
interaction.plot(tstspd,twstlvl,hairiness)        # Interaction Plot

# Fit Model with Interaction and bobbin effect
hair1.aov <- anova(lm(hairiness ~ twstlvl*tstspd + bobbin)); hair1.aov
# Fit Additive Model with  bobbin effect (Use aov to get TukeyHSD)
hair2.aov <- aov(hairiness ~ twstlvl + tstspd + bobbin); summary(hair2.aov)

# Tukey's HSD For Main Effects
TukeyHSD(hair2.aov, "twstlvl")
TukeyHSD(hair2.aov, "tstspd")
```

Output – Part 1

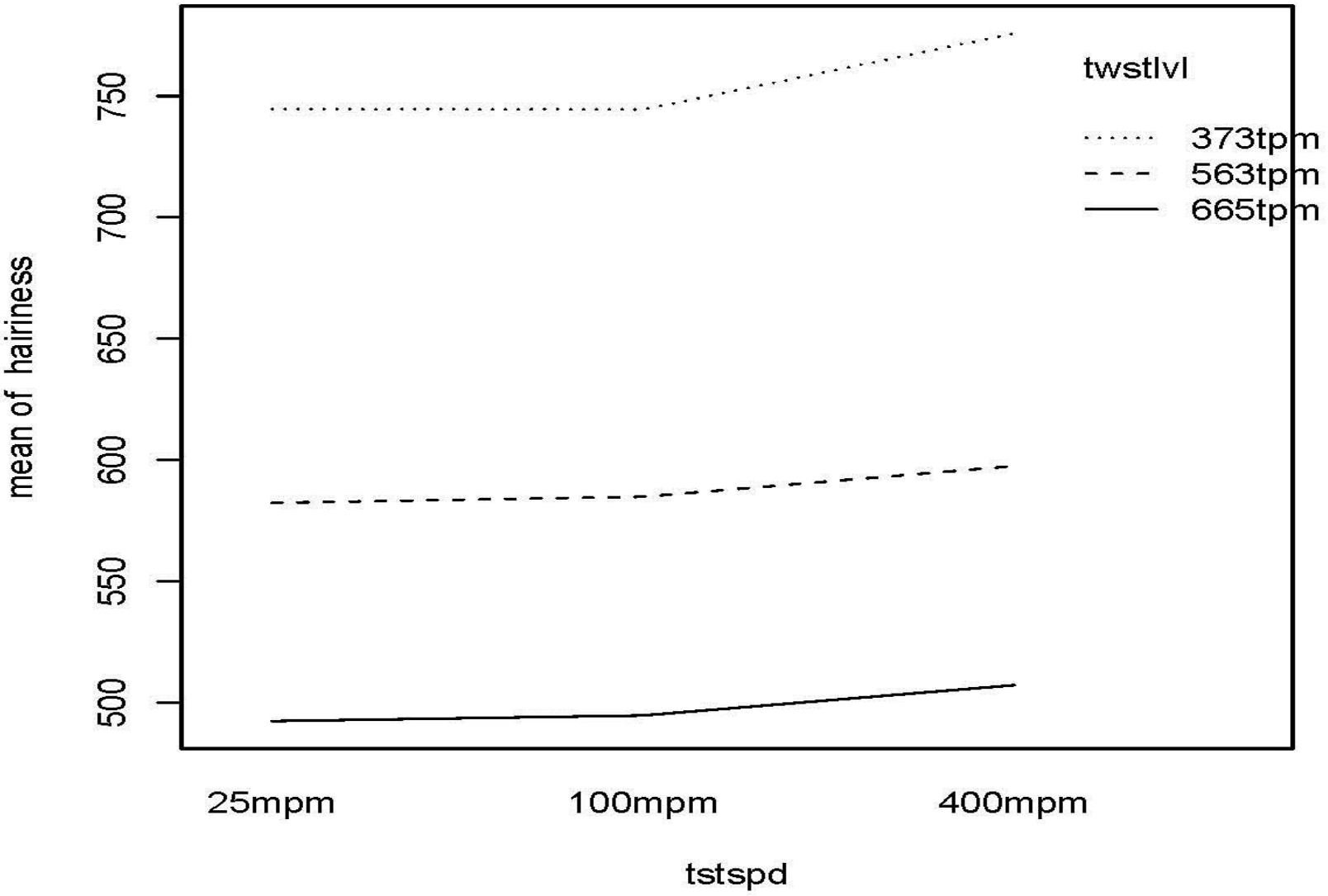
```
> tapply(hairiness, bobbin, mean) # marginal mean for bobbin
       1        2        3        4        5        6
632.5556 607.5556 608.5556 605.3333 614.4444 614.1111
> tapply(hairiness, list(twstlvl,tstspd), mean) #Trt cell means
          25mpm   100mpm   400mpm
373tpm 744.6667 744.5000 775.8333
563tpm 582.3333 584.8333 597.5000
665tpm 492.3333 494.6667 507.1667
>
> # Fit Model with Interaction and bobbin effect
> hair1.aov <- anova(lm(hairiness ~ twstlvl*tstspd + bobbin))
> hair1.aov

Analysis of Variance Table
Response: hairiness
               Df Sum Sq Mean Sq   F value    Pr(>F)
twstlvl         2 611792  305896 1324.6920 < 2.2e-16 ***
tstspd          2   4637    2318   10.0402 0.0002928 ***
bobbin          5   4414     883    3.8231 0.0063403 **
twstlvl:tstspd  4    826     207    0.8946 0.4761743
Residuals      40   9237     231
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Output – Part 2

```
> # Fit Additive Model with  bobbin effect
> hair2.aov <- aov(hairiness ~ twstlvl + tstspd + bobbin)
> summary(hair2.aov)
            Df Sum Sq Mean Sq   F value    Pr(>F)
twstlvl      2 611792  305896 1337.5107 < 2.2e-16 ***
tstspd       2   4637    2318   10.1373 0.0002394 ***
bobbin       5   4414     883    3.8601 0.0054948 **
Residuals   44  10063     229
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> # Tukey's HSD For Main Effects
>
> TukeyHSD(hair2.aov, "twstlvl")
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = hairiness ~ twstlvl + tstspd + bobbin)
```

Output – Part 3

```
Fit: aov(formula = hairiness ~ twstlvl + tstspd + bobbin)

$twstlvl
                     diff       lwr       upr p adj
563tpm-373tpm -166.77778 -179.0046 -154.5509     0
665tpm-373tpm -256.94444 -269.1713 -244.7176     0
665tpm-563tpm  -90.16667 -102.3935  -77.9398     0

> TukeyHSD(hair2.aov, "tstspd")
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = hairiness ~ twstlvl + tstspd + bobbin)

$tstspd
                     diff        lwr      upr     p adj
100mpm-25mpm    1.555556 -10.671306 13.78242 0.9489254
400mpm-25mpm   20.388889   8.162027 32.61575 0.0005994
400mpm-100mpm  18.833333   6.606472 31.06020 0.0015237
```


mean of hairiness
500
550
600
650
700
750
25mpm
100mpm
400mpm
tstspd
twstlvl
373tpm
563tpm
665tpm

Nested Design – Florida Swamp Depths

Response=Watlev Nesting Factor=size Nested Factor = swampid

```
pdf("swamp.pdf")
swamp <- read.fwf("http://www.stat.ufl.edu/~winner/data/swamp1.dat",
width=c(8,8,8), col.names=c("size", "swampid", "watlev"))
attach(swamp)

size <- factor(size, levels=1:3, labels=c("small", "medium", "large"))
swampid <- factor(swampid, levels=1:9)
tapply(watlev, size, mean)

# Fit the ANOVA with size and swamp nested within size
swamp.aov1 <- aov(watlev ~ size/swampid)
# This provides ANOVA, not appropriate F-test for size
summary(swamp.aov1)

swamp.aov2 <- aov(watlev ~ size + Error(swampid))
# This provides appropriate F-test for size

summary(swamp.aov2)
```

Output – Swamp Depths

```
> tapply(watlev, size, mean)
    small    medium     large
 52.16667 106.43444 152.76654
>
>
> # Fit the ANOVA with size and swamp nested within size
>
> swamp.aov1 <- aov(watlev ~ size/swampid)
>
> # This provides ANOVA, not appropriate F-test for size
>
> summary(swamp.aov1)
              Df Sum Sq Mean Sq F value    Pr(>F)
size           2 410724  205362 888.385 < 2.2e-16 ***
size:swampid   6  83058   13843  59.884 < 2.2e-16 ***
Residuals    234  54092     231
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Output – Swamp Depths

```
> swamp.aov2 <- aov(watlev ~ size + Error(swampid))
>
> # This provides appropriate F-test for size
>
> summary(swamp.aov2)

Error: swampid
          Df Sum Sq Mean Sq F value   Pr(>F)
size       2 410724  205362  14.835 0.004759 **
Residuals  6  83058   13843
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: Within
           Df Sum Sq Mean Sq F value Pr(>F)
Residuals 234  54092     231
```

Split-Plot Design – Wool Shrinkage

Response = shrink
Whole Plot Factor = trt
Subplot Factor = revs
Block = runnum

```
pdf("woolfrict.pdf")

wf1 <- read.fwf("http://www.stat.ufl.edu/~winner/data/woolfrict1.dat",
width=c(8,8,8,8),
col.names=c("runnum", "trt", "revs", "shrink"))

attach(wf1)

trt <- factor(trt, levels=1:4,
labels=c("Untreated", "AC15sec", "AC4min", "AC15min"))

runnum <- factor(runnum, levels=1:4)
revs <- factor(revs,levels=seq(200,1400,200),ordered=TRUE)
```

Split-Plot Design – Wool Shrinkage

```
# Fit full ANOVA with all terms, F-test for trt is inappropriate
woolfrict.sp1 <- aov(shrink ~ trt + runnum + trt:runnum + revs + revs:trt)
summary(woolfrict.sp1)

# This model uses correct error terms for trt and revs and interaction
woolfrict.sp2 <- aov(shrink ~ trt*revs + Error(runnum/trt))
summary(woolfrict.sp2)

# Partitions the Revs SS into orthogonal polynomials
summary(woolfrict.sp2,split=list(revs=list(linear=1, quadratic=2,
cubic=3, quartic=4, quintic=5, sextic=6)))
```

Output – Wool Shrinkage

```
> > # Fit full ANOVA with all terms, F-test for trt is
inappropriate
> woolfrict.sp1 <- aov(shrink ~ trt + runnum + trt:runnum +
revs + revs:trt)
> summary(woolfrict.sp1)
             Df  Sum Sq Mean Sq   F value    Pr(>F)
trt           3  3012.5  1004.2  729.1367 < 2.2e-16 ***
runnum        3   124.3    41.4   30.0815 1.024e-12 ***
revs          6 11051.8  1842.0 1337.4577 < 2.2e-16 ***
trt:runnum    9   114.6    12.7    9.2487 3.546e-09 ***
trt:revs     18   269.5    15.0   10.8718 4.620e-14 ***
Residuals    72    99.2     1.4
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Output – Wool Shrinkage

```
> # This model uses correct error terms for trt and revs and
interaction
> woolfrict.sp2 <- aov(shrink ~ trt*revs + Error(runnum/trt))
> summary(woolfrict.sp2)
Error: runnum
          Df  Sum Sq Mean Sq F value Pr(>F)
Residuals  3 124.286  41.429

Error: runnum:trt
          Df  Sum Sq Mean Sq F value   Pr(>F)
trt        3 3012.53 1004.18  78.836 8.81e-07 ***
Residuals  9  114.64   12.74
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: Within
          Df  Sum Sq Mean Sq  F value    Pr(>F)
revs       6 11051.8  1842.0 1337.458 < 2.2e-16 ***
trt:revs  18   269.5    15.0   10.872  4.62e-14 ***
Residuals 72    99.2     1.4
```

Output – Wool Shrinkage

```
> # Partitions the Revs SS into orthogonal polynomials
> summary(woolfrict.sp2,split=list(revs=list(linear=1, quadratic=2,
+ cubic=3, quartic=4, quintic=5, sextic=6)))
Error: runnum
          Df  Sum Sq Mean Sq F value Pr(>F)
Residuals  3 124.286  41.429
Error: runnum:trt
          Df  Sum Sq Mean Sq F value   Pr(>F)
trt        3 3012.53 1004.18  78.836 8.81e-07 ***
Residuals  9  114.64   12.74
Error: Within
                      Df  Sum Sq Mean Sq   F value    Pr(>F)
revs                   6 11051.8  1842.0 1337.4577 < 2.2e-16 ***
  revs: linear         1 10846.8 10846.8 7875.9309 < 2.2e-16 ***
  revs: quadratic      1   198.9   198.9  144.4046 < 2.2e-16 ***
  revs: cubic          1     2.9     2.9    2.0842    0.1532
  revs: quartic        1     1.4     1.4    1.0398    0.3113
  revs: quintic        1     0.1     0.1    0.0885    0.7669
  revs: sextic         1     1.7     1.7    1.1983    0.2773
trt:revs              18   269.5    15.0   10.8718 4.620e-14 ***
  trt:revs: linear     3   154.4    51.5   37.3624 1.133e-14 ***
  trt:revs: quadratic  3   104.8    34.9   25.3581 2.646e-11 ***
  trt:revs: cubic      3     7.4     2.5    1.7944    0.1559
  trt:revs: quartic    3     0.8     0.3    0.1861    0.9055
  trt:revs: quintic    3     0.9     0.3    0.2099    0.8892
  trt:revs: sextic     3     1.3     0.4    0.3199    0.8110
Residuals             72    99.2     1.4
```

Repeated Measures Design – Hair Growth

Response = hair
Treatment Factor = trt
Subject within Treatment Factor = subj
Time Factor = time

```
pdf("rogaine.pdf")

rogaine <- read.fwf("http://www.stat.ufl.edu/~winner/data/rogaine.dat",
width=c(8,8,8,8), col.names=c("trt", "subj", "time", "hair"))

# Only use values where time>0
rogaine2 <- subset(rogaine,rogaine$time>0)


rogaine1 <- data.frame(trt=factor(rogaine2$trt),
subj=factor(rogaine2$subj),
time=factor(rogaine2$time), hair=rogaine2$hair)


attach(rogaine1)
```

Repeated Measures Design – Hair Growth

```
# Give ANOVA with all factors and interactions (inappropriate error for trt)

rogaine1.mod1 <- aov(hair ~ trt + trt/subj + time + time:trt)

summary(rogaine1.mod1)

# Give ANOVA with proper error term for trt

rogaine1.mod2 <- aov(hair ~ trt*time + Error(subj))

summary(rogaine1.mod2)
```

Output – Hair Growth

```
> # Give ANOVA with all factors and interactions
(inappropriate error for trt)
> rogaine1.mod1 <- aov(hair ~ trt + trt/subj + time +
time:trt)
>
> summary(rogaine1.mod1)
            Df Sum Sq Mean Sq F value    Pr(>F)
trt          1   8064    8064 27.7389 5.231e-05 ***
time         3   2268     756  2.5999   0.08391 .
trt:subj     6  55476    9246 31.8027 1.230e-08 ***
trt:time     3   2005     668  2.2985   0.11201
Residuals   18   5233     291
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Output – Hair Growth

```
> # Give ANOVA with proper error term for trt
> rogaine1.mod2 <- aov(hair ~ trt*time + Error(subj))
>
> summary(rogaine1.mod2)

Error: subj
          Df Sum Sq Mean Sq F value Pr(>F)
trt        1   8064    8064  0.8722 0.3864
Residuals  6  55476    9246

Error: Within
          Df Sum Sq Mean Sq F value  Pr(>F)
time       3 2267.6   755.9  2.5999 0.08391 .
trt:time   3 2004.8   668.3  2.2985 0.11201
Residuals 18 5233.1   290.7
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Linear Regression

- Response/Dependent Variable – Y
- Explanatory/Predictor/Independent Variable(s) – $X_1,\ldots,X_p$
- Predictors can me numeric, qualitative (using dummy variables), polynomials, or cross-products
- Methods based on independent and normally distributed errors with constant variance

Simple Linear Regression – Math Scores / LSD Levels

```
mathlsddat <- read.fwf("http://www.stat.ufl.edu/~winner/data/lsd.dat",
  width=c(4,8),  col.names=c("lsd","math"))

mathlsd <- data.frame(mathlsddat)
attach(mathlsd)

# Fit the simple linear regression model with Y = math (score) and X = lsd
(concentration)
mathlsd.reg <- lm(math ~ lsd)

# Plot the data and regression line (Note you enter X first, then Y in plot statement)
plot(lsd,math)
abline(mathlsd.reg)

# Print out the estimates, standard errors and t-tests, R-Square, and F-test
summary(mathlsd.reg)
```

Simple Linear Regression – Math Scores / LSD Levels

```
# Print out the ANOVA table (Sums of Squares and degrees of freedom)
anova(mathlsd.reg)

# Compute the correlation coefficient (r) and test if rho=0
cor(math,lsd)
cor.test(math,lsd)

# Plot Residuals versus Fitted Values
png("mathlsd2.png")
plot(predict(mathlsd.reg),residuals(mathlsd.reg))
abline(h=0)
dev.off()

# Print out standardized, studentized Residuals and Influence Measures
round(rstandard(mathlsd.reg),3)
round(rstudent(mathlsd.reg),3)
influence.measures(mathlsd.reg)
```

Output – Math/LSD

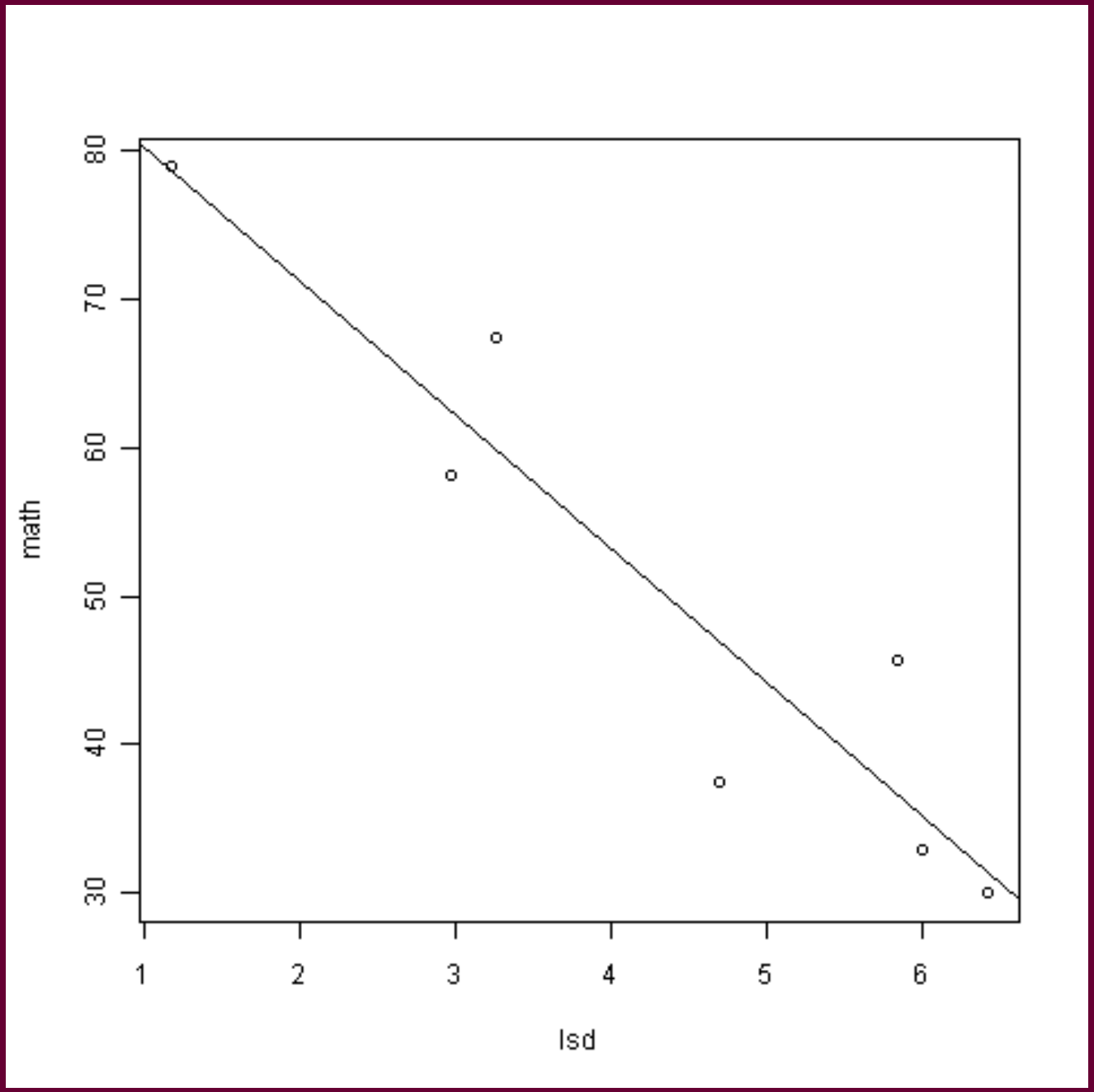
```
> summary(mathlsd.reg)
Call: lm(formula = math ~ lsd)
Residuals:
      1       2       3       4       5       6       7
 0.3472 -4.1658  7.7170 -9.3995  9.0513 -2.1471 -1.4032
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   89.124      7.048  12.646 5.49e-05 ***
lsd           -9.009      1.503  -5.994  0.00185 **
Residual standard error: 7.126 on 5 degrees of freedom
Multiple R-squared: 0.8778,      Adjusted R-squared: 0.8534
F-statistic: 35.93 on 1 and 5 DF,  p-value: 0.001854
> anova(mathlsd.reg)
Analysis of Variance Table
Response: math
          Df  Sum Sq Mean Sq F value   Pr(>F)
lsd        1 1824.30 1824.30  35.928 0.001854 **
Residuals  5  253.88   50.78
```

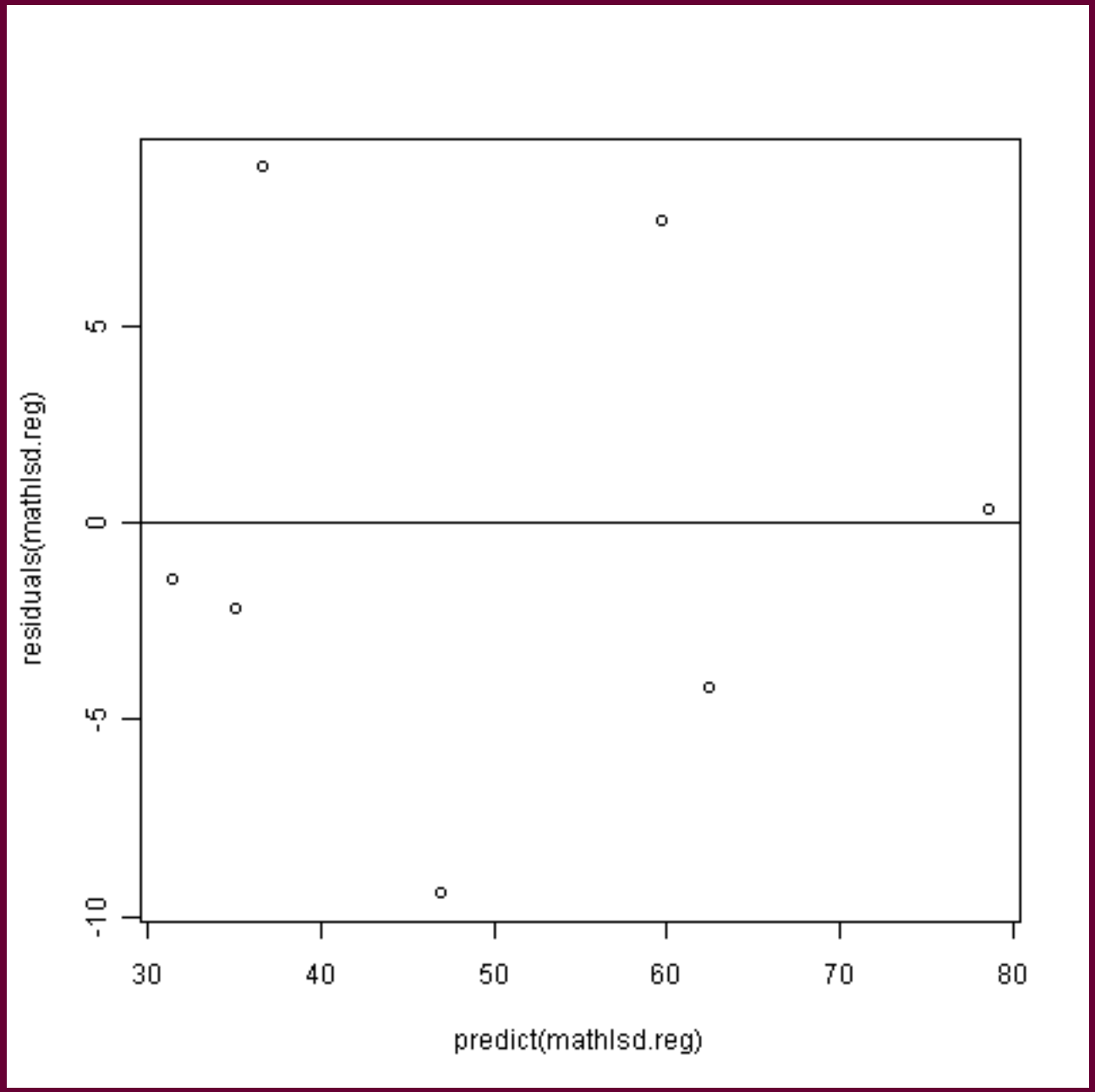
Output – Math/LSD

```
> cor(math,lsd)
[1] -0.9369285
> cor.test(math,lsd)
        Pearson's product-moment correlation
data:  math and lsd
t = -5.994, df = 5, p-value = 0.001854
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.9908681 -0.6244782
sample estimates:
       cor
-0.9369285

> round(rstandard(mathlsd.reg),3)
     1      2      3      4      5      6      7
 0.076 -0.664  1.206 -1.430  1.460 -0.352 -0.241
> round(rstudent(mathlsd.reg),3)
     1      2      3      4      5      6      7
 0.068 -0.622  1.281 -1.663  1.723 -0.319 -0.217
```

Output – Math/LSD

```
> influence.measures(mathlsd.reg)
Influence measures of
        lm(formula = math ~ lsd) :

   dfb.1_ dfb.lsd   dffit cov.r  cook.d   hat inf
1  0.0805 -0.0706  0.0811 3.783 0.00411 0.588   *
2 -0.2900  0.2033 -0.3358 1.677 0.06424 0.225
3  0.5047 -0.3230  0.6288 0.974 0.17521 0.194
4 -0.1348 -0.1358 -0.6945 0.642 0.17824 0.149
5 -0.2918  0.6253  0.9752 0.680 0.34114 0.243
6  0.0672 -0.1308 -0.1921 2.026 0.02249 0.267
7  0.0694 -0.1167 -0.1541 2.295 0.01467 0.335   *
```


math
80
70
60
50
40
30
1
2
3
4
5
6
lsd


residuals(mathlsd.reg)
5
0
-5
-10
30
40
50
60
70
80
predict(mathlsd.reg)

Multiple Regression – PGA/LPGA

Y = Accuracy (pctfrway=Percent Fairways) X_1 = Average Distance X_2 = Male

```
png("C:\\Rmisc\\pgalpga.png")

pgalpga <- read.fwf("http://www.stat.ufl.edu/~winner/data/pgalpga2008.dat",
      width=c(8,8,8),  col.names=c("avedist","pctfrwy","gender"))
attach(pgalpga)
# Gender has levels 1=Female, 2=Male. Create Male "Dummy" variable
male <- gender-1

# Fit additive Model (Common slopes)
add.model <- lm(pctfrwy ~ avedist + male)

# Fit interaction model (Different slopes by gender)
int.model <- lm(pctfrwy ~ avedist*male)

summary(add.model)
summary(int.model)

dev.off()
```

PGA/LPGA Output - I

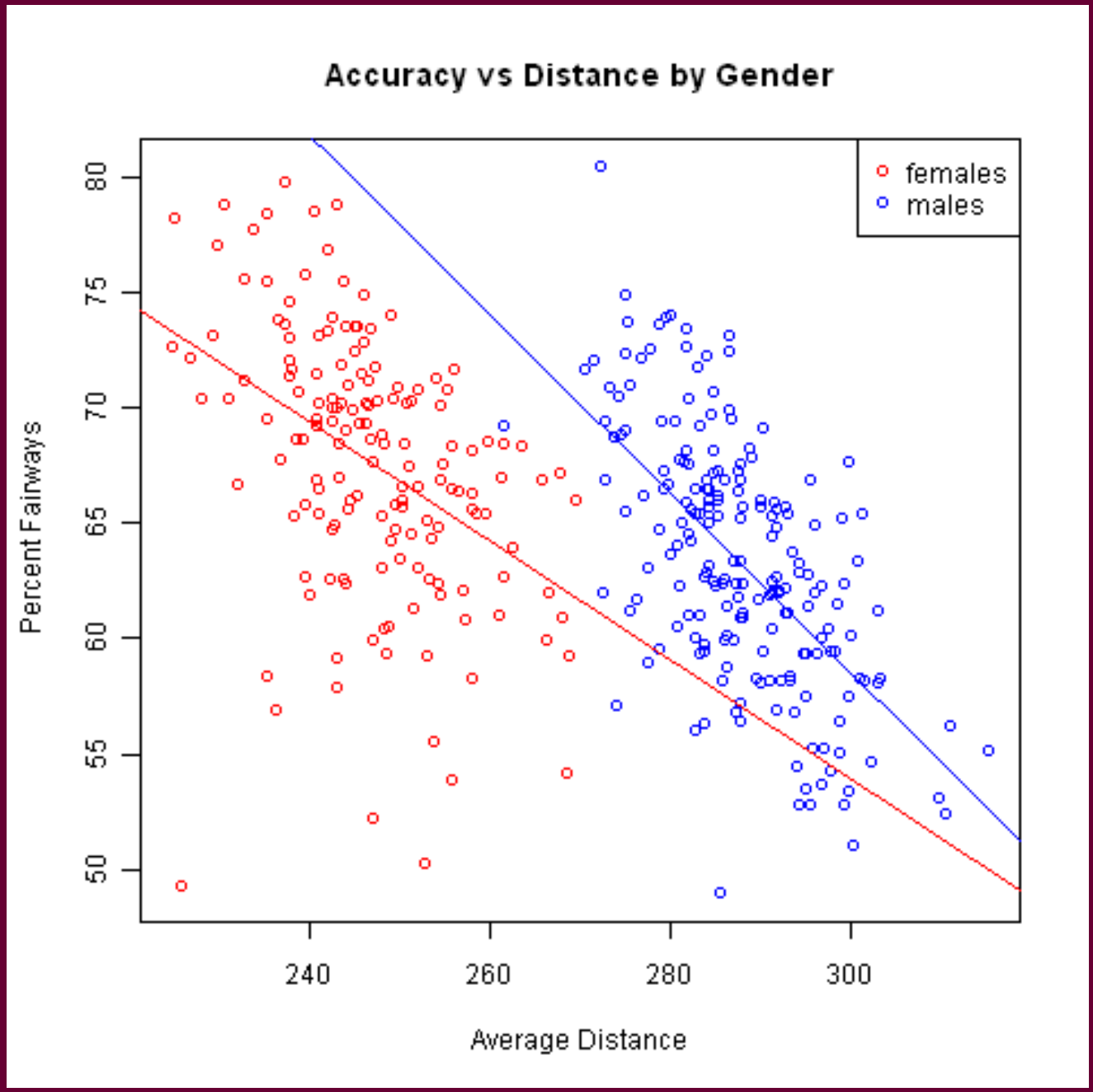
```
> summary(add.model)

Call:
lm(formula = pctfrwy ~ avedist + male)

Residuals:
     Min       1Q   Median       3Q      Max
-25.0712  -2.8263   0.4867   3.3494  12.0275

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 147.26894    7.03492  20.934  < 2e-16 ***
avedist      -0.32284    0.02846 -11.343  < 2e-16 ***
male          8.94888    1.26984   7.047 9.72e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.797 on 351 degrees of freedom
Multiple R-squared: 0.3589,     Adjusted R-squared: 0.3552
F-statistic: 98.24 on 2 and 351 DF,  p-value: < 2.2e-16
```

PGA/LPGA Output - II

```
> summary(int.model)

Call:
lm(formula = pctfrwy ~ avedist * male)

Residuals:
     Min       1Q   Median       3Q      Max
-23.6777  -2.7462   0.3365   3.3635  11.0186

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  130.89331    9.92928  13.183  < 2e-16 ***
avedist       -0.25649    0.04020  -6.380 5.61e-10 ***
male          44.03215   15.15809   2.905  0.00391 **
avedist:male  -0.13140    0.05657  -2.323  0.02078 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.767 on 350 degrees of freedom
Multiple R-squared: 0.3686,     Adjusted R-squared: 0.3632
F-statistic: 68.11 on 3 and 350 DF,  p-value: < 2.2e-16
```

Obtaining Plot by Gender with Regression Lines

```
png("C:\\Rmisc\\pgalpga.png")
pgalpga <- read.fwf("http://www.stat.ufl.edu/~winner/data/pgalpga2008.dat ",
  width=c(8,8,8),  col.names=c("avedist","pctfrwy","gender"))
attach(pgalpga)
male <- gender-1
# Create measures specifcally by gender
ad.female <- avedist[gender == 1]; ad.male <- avedist[gender == 2]
pf.female <- pctfrwy[gender == 1]; pf.male <- pctfrwy[gender == 2]

# First “set up” variables to plot but leave blank (type=“n”). Then add points,
#   lines, and legend
plot(c(ad.female,ad.male),c(pf.female,pf.male),
 xlab="Average Distance", ylab="Percent Fairways",
 main="Accuracy vs Distance by Gender", type="n")

points(ad.female,pf.female,col="red"); points(ad.male,pf.male,col="blue")

abline(lm(pf.female~ad.female),col="red"); abline(lm(pf.male~ad.male),col="blue")

legend("topright",c("females","males"),pch=c(1,1),col=c(2,4))
dev.off()
```


Accuracy vs Distance by Gender
females
males
Percent Fairways
50
55
60
65
70
75
80
240
260
280
300
Average Distance

Generalized Linear Models

- General class of linear models that are made up of 3 components: Random, Systematic, and Link Function
  - Random component: Identifies dependent variable (*Y*) and its probability distribution
    - Binary Responses $\Rightarrow$ Binomial Distribution
    - Count Responses $\Rightarrow$ Poisson Distribution (or Negative Binomial)
    - Positive and Skewed Responses $\Rightarrow$ Gamma Distribution
  - Systematic Component: Identifies the set of explanatory variables (X_1,...,X_k)
  - Link Function: Identifies a function of the mean that is a linear function of the explanatory variables

$$g(\mu) = \alpha + \beta_1 X_1 + \cdots + \beta_k X_k$$

Common Link Functions

- Identity link (form used in *normal* and *gamma* regression models):

$$g(\mu) = \mu$$

- Log link (used when μ cannot be negative as when data are *Poisson* counts):

$$g(\mu) = \log(\mu)$$

- Logit link (used when μ is bounded between 0 and 1 as when data are binary):

$$g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$$

Logistic Regression – NFL Field Goals

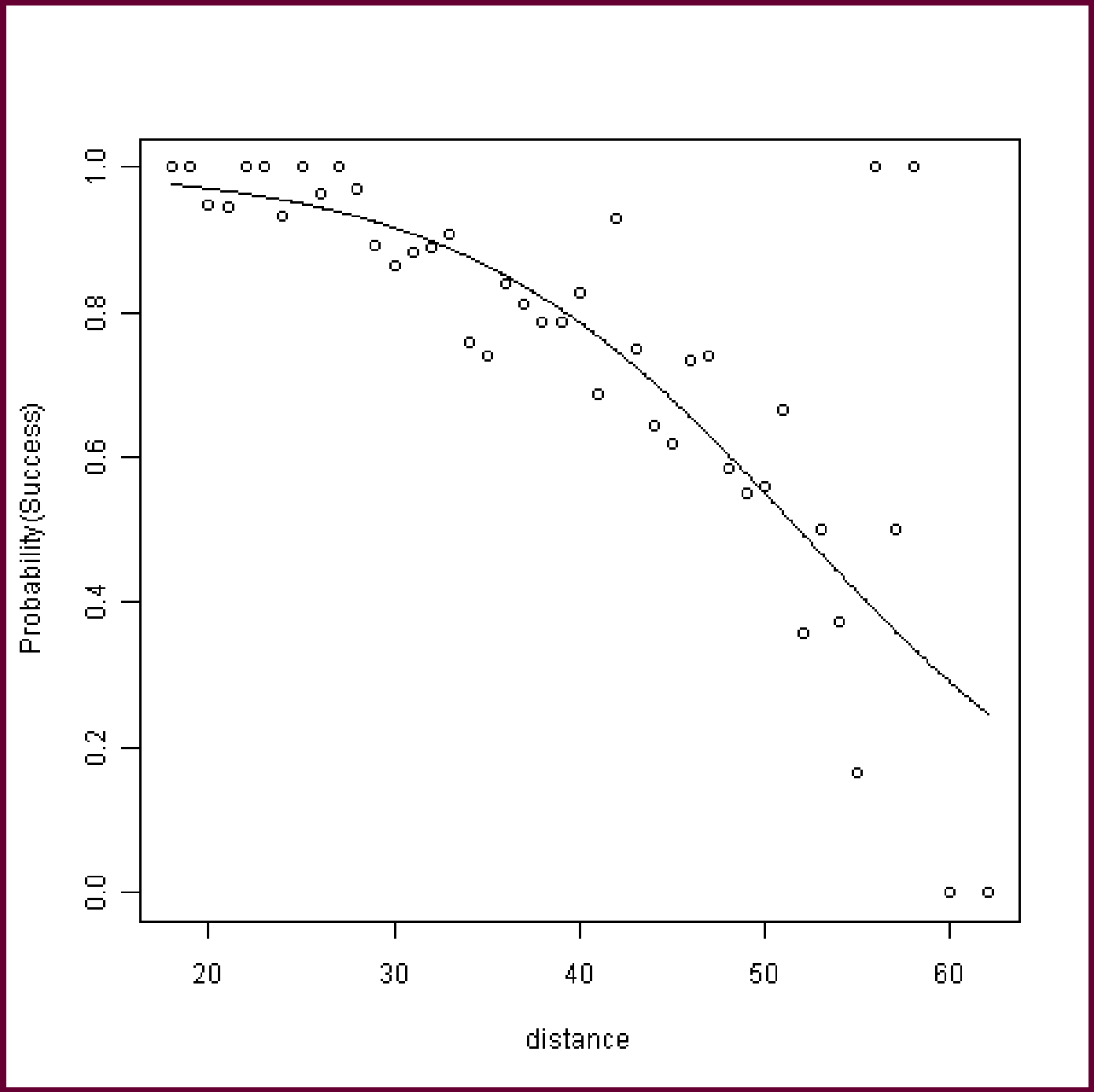
```
png("fieldgoal.png")
fieldgoal <- read.fwf("http://www.stat.ufl.edu/~winner/data/fieldgoal.dat",
  width=c(8,8,8),  col.names=c("distance","success","week"))
attach(fieldgoal)

# Create a contingency Table, Conditional (Row) Distributions
#  and variables for observed proportion of Success (prop.dist)
#  and observed distances (obs.dist) for plot
dstable <- table(distance,success)
prop.dstable <- prop.table(dstable,1)
prop.dist <- prop.dstable[,2]
obs.dist <- as.numeric(rownames(dstable))

# Fit the logistic regression model and print results
fg.reg1 <- glm(success~distance,binomial)
summary(fg.reg1)
```

Logistic Regression – NFL Field Goals

```
# Give a range of Distance values for the smooth fitted/predicted curve
raw.dist <- seq(min(distance),max(distance),0.1)

# Plot the raw data and add line with predicted curve evaluated at raw.dist
#   type="response" backtransforms model to S-shaped response
#   scale (not linear scale)

plot(obs.dist,prop.dist,xlab="distance",ylab="Probability(Success)")

lines(raw.dist,predict(fg.reg1,list(distance=raw.dist),type="response"))

dev.off()
```

Text Output – Field Goal Data

```
> # Fit the logistic regression model and print results
> fg.reg1 <- glm(success~distance,binomial)
> summary(fg.reg1)
Call:
glm(formula = success ~ distance, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6569   0.2718   0.4166   0.6938   1.4750

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  5.69788    0.45110   12.63   <2e-16 ***
distance    -0.10991    0.01058  -10.38   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 955.38  on 947  degrees of freedom
Residual deviance: 817.20  on 946  degrees of freedom
AIC: 821.2

Number of Fisher Scoring iterations: 5
```


Probability(Success)
0.0
0.2
0.4
0.6
0.8
1.0
20
30
40
50
60
distance

Poisson Regression – NASCAR Crashes

```
race1 <- read.fwf("http://www.stat.ufl.edu/~winner/data/race7579.dat",
  width=c(rep(8,9),12,40),  col.names=c("srace", "year", "yrace", "drivers",
  "trklen", "laps", "roadtrk", "cautions", "leadchng", "trkid", "track"))
attach(race1)

## Fit Poisson Regression, Relating Cautions to Drivers, Track Length, and Laps

race.mod <- glm(cautions ~ drivers + trklen + laps, poisson)
summary(race.mod)
anova(race.mod, test="Chisq")      #  Gives sequential tests
```

Output - NASCAR Crashes

```
> summary(race.mod)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.7962699  0.4116942  -1.934  0.05310 .
drivers      0.0365253  0.0124932   2.924  0.00346 **
trklen       0.1144986  0.1684236   0.680  0.49662
laps         0.0025963  0.0007893   3.289  0.00100 **

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 215.49  on 150  degrees of freedom
Residual deviance: 171.22  on 147  degrees of freedom
AIC: 671.11

> anova(race.mod, test="Chisq")      #  Gives sequential tests
Analysis of Deviance Table
Model: poisson, link: log
Response: cautions
Terms added sequentially (first to last)

        Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                     150     215.49
drivers  1   5.1673       149     210.32  0.023016 *
trklen   1  28.1912       148     182.13 1.099e-07 ***
laps     1  10.9167       147     171.22  0.000953 ***
---
```