The resulting 95% confidence interval is

This should be

cii proportions 20 0, agresti

$$\hat{\pi} \pm 1.96(se \quad), \quad \text{or} \quad (-0.03, 0.19).$$

A proportion can interval as (0.0, 0.19).

We can also find this interval using some software, or on the Internet[14]. Here is how you can do it using Stata[15], by applying the *cii* command to n and the count in the category of interest:

```
-----------------------------------------------------------------
. cii 20 0, agresti
                                                  -- Agresti-Coull ---
 Variable |      Obs      Mean     Std. Err.     [95% Conf. Interval]
          |       20         0            0             0    .1898096
------------------------------------------------------------------
```

We can be 95% confident that the proportion of vegetarians at the University of Florida is no greater than 0.19.

Why do we add 2 to the counts of the two types? The reason is that the confidence interval then approximates one based on a more complex method (described in Exercise 5.77) that does not require estimating the standard error.

5.5 ESTIMATION METHODS: MAXIMUM LIKELIHOOD AND THE BOOTSTRAP*

We've focused on estimating means and proportions, but Chapter 3 showed that other statistics are also useful for describing data. These other statistics also have sampling distributions. In this section, we introduce a standard method, called *maximum likelihood*, that statisticians use to find good estimators of parameters. We also introduce a newer method, called the *bootstrap*, that uses modern computational power to find confidence intervals in cases in which it is difficult to derive the sampling distribution.

Maximum Likelihood Method of Estimation

The most important contributions to modern statistical science were made by a British statistician and geneticist, R. A. Fisher (1890–1962). While working at an agricultural research station north of London, he developed much of the theory of point estimation as well as methodology for the design of experiments and data analysis. For point estimation, Fisher advocated the ***maximum likelihood estimate***. This estimate is the value of the parameter that is most

[14]For example, at `https://istats.shinyapps.io/Inference_prop` and `http://epitools.ausvet.com.au/content.php?page=CIProportion`

[15]Stata calls it the *Agresti-Coull* confidence interval, because it was proposed in an article by A. Agresti and B. Coull, *American Statistician*, vol. 52, pp. 119-126, 1998.

EXAMPLE 6.4: Mean Weight Change in Anorexic Girls

Example 5.5 in Chapter 5 (page 161) analyzed data from a study comparing treatments for teenage girls suffering from anorexia. For each girl, the study observed her change in weight while receiving the therapy. Let μ denote the population mean change in weight for the cognitive behavioral treatment. If this treatment has beneficial effect, as expected, then μ is positive. To test for no treatment effect versus a positive mean weight change, we test H_0: $\mu = 0$ against H_a: $\mu > 0$.

In the Chapter 5 analysis, we found that the $n = 29$ girls had a sample mean weight change of 3.007 pounds, a standard deviation of 7.309 pounds, and an estimated standard error of $se = 1.357$. The test statistic equals

$$t = \frac{\bar{y} - \mu_0}{se} = \frac{3.007 - 0}{1.357} = 2.22.$$

For this one-sided H_a, the P-value is the right-tail probability above 2.22. Why do we use the right tail? Because H_a: $\mu > 0$ has values *above* (that is, to the right of) the null hypothesis value of 0. It's the positive values of t that support this alternative hypothesis.

Now, for $n = 29$, $df = n - 1 = 28$. The P-value equals 0.02. Software can do the calculation for you. For instance, for the one-sided and two-sided alternatives with a data file with variable *change* for weight change, R reports:

```
-----------------------------------------------------------------
> t.test(change, mu=0, alternative='greater')$p.value
[1] 0.0175113
> t.test(change, mu=0, alternative='two.sided')$p.value
[1] 0.0350226
------------------------------------------------------------------
```

Using its `ttest` command with the data file, Stata also reports $P = 0.0175$ for the one-sided alternative:

```
-----------------------------------------------------------------------
. ttest change == 0

One-sample t test
Variable |  Obs      Mean   Std. Err.   Std. Dev.   [95% Conf. Interval]
  change |   29  3.006896   1.357155    7.308504    .2268896    5.786902

   mean = mean(change)                                        t =   2.2156
Ho: mean = 0                                    degrees of freedom = 28

   Ha: mean < 0              Ha: mean != 0               Ha: mean > 0
 Pr(T < t) = 0.9825      Pr(|T| > |t|) = 0.0350      Pr(T > t) = 0.0175
------------------------------------------------------------------------
```

I would think this should be 'only', because it is easier to use –ttest– than to use –ttesti–, what with the latter needing positional parameters (or the use of a dialog box).

If you already have summary statistics, Stata can conduct the test using them, with the `ttesti` command, by entering n, $\bar{y}$, s, and μ_0:

Under random sampling, this test statistic has approximately the standard normal distribution, when H_0 is true. Some software also reports a se and/or related P-value that holds only under H_0. The normal approximation holds better with larger n, and is adequate when each of C and D exceed about 50.

EXAMPLE 8.8: Inference about Association between Income and Happiness

For the data in Table 8.16 on family income and happiness, Table 8.17 shows software output for the analysis of the data. The $\hat{\gamma} = 0.145$ value has $se = 0.079$. (It is labeled as ASE, where the A stands for "asymptotic," meaning it is an approximate large-sample standard error.) With some software, if we already have the cell counts we can enter them and get gamma and its standard error, such as in Stata with the command

```
tabi 37 90 45 \ 25 93 56 \ 6 18 13, gamma
```

A 95% confidence interval for γ in the population is

$$\hat{\gamma} \pm 1.96(se), \quad \text{or } 0.145 \pm 1.96(0.079), \quad \text{or } 0.146 \pm 0.155,$$

which equals $(-0.01, 0.30)$. We can be 95% confident that γ is no less than -0.01 and no greater than 0.30. It is plausible that essentially no association exists between income and happiness, but it is also plausible that a moderate positive association exists. We need a larger sample size to estimate this more precisely.

Table 8.17. Part of Software Output for Analyzing Table 8.16

	Value	DF	P-Value
Pearson Chi-Square	4.092	4	0.394

Statistic	Value	ASE	P-Value
Gamma	0.1454	0.0789	0.064

Stata gives a different value than these for the above table:
chi2: 4.1266   Pvalue: 0.389

For testing independence between family income and happiness, the chi-squared test of independence has $X^2 = 4.09$ with $df = 4$, for which the P-value equals 0.39. This test does not show any evidence of an association. The chi-squared test treats the variables as nominal, however, and ordinal-level methods are more powerful if there is a positive or negative trend. The ordinal test statistic using gamma equals

$$z = \frac{\hat{\gamma} - 0}{se} = \frac{0.145 - 0}{0.079} = 1.84.$$

The P-value for H_a: $\gamma \neq 0$ equals 0.065. This test shows some evidence of an

You can fit multilevel models for quantitative response variables using linear mixed models, as shown in the table on page 683. See also the `multilevel` package.

For event history (survival) models, the `survival` package can fit Cox models. For a variety of analyses in R, see the book *Event History Analysis with R* by G. Broström (CRC Press, 2012).

You can conduct factor analysis using the `factanal` function or the `factor.pa` function in the `psych` package. Packages for structural equations modeling include `sem` and `lavaan`.

The `markovchain` package is available for Markov chains.

For Bayesian inference, Table 16.10 shows that you can fit normal regression models with the `bayesglm` function in the `arm` package. It uses t distribution priors, which are normal when you take $df = \infty$ for the t distribution and take the prior scale parameter to be infinite. The `MCMCregress` function in the `MCMCpack` package can also fit normal regression models. Select improper uniform priors for $\boldsymbol{\beta}$ by taking the normal prior to have precision (which is the reciprocal of the variance) equal to 0. It uses a gamma prior distribution for $1/\sigma^2$, and that prior is practically uniform over the real line for $\log(\sigma^2)$ when you take tiny values for the two parameters of a gamma prior distribution ($c0$ and $d0$). For logistic regression you can use the `MCMClogit` function in the `MCMCpack` package or the `bayesglm` function.

INTRODUCTION TO STATA

Basic support information is available from Stata at

`www.stat.com/support`

Many Internet sites can help you learn how to use Stata, such as the many resources listed at

`www.stata.com/links/resources-for-learning-stata`.

Examples are

`www.ats.ucla.edu/stat/stata`

Repeated references (although the UCLA site is good enough for such praise ;<))

`www.princeton.edu/~otorres/Stata`

`data.princeton.edu/stata`

`www.cpc.unc.edu/research/tools/data_analysis/statatutorial`

`homepages.rpi.edu/~simonk/pdf/UsefulStataCommands.pdf`

`www.ats.ucla.edu/stat/stata`

Also, examples are shown for a previous edition of this textbook at

`www.ats.ucla.edu/stat/examples/smss`

These tutorials and the discussion below show commands to enter to perform various statistical analyses. Commands are case-sensitive. To get information about a command, use the `help` command; e.g., to get information about the `histogram` command, enter

It would be better to leave off the extension here. Using a .txt extension still saves the file as a .smcl file, but obscures the meaning of the file. If you meant to use the .log extension, I would advise against this, also, because smcl files can be translated to text, but also can be translated to html and can be post-processed much more easily.

```
help histogram
```

For many purposes, once you load a data file, it is simpler to use the Statistics and Graphics menus that Stata provides.

Reading Data Files and Using Stata

After starting Stata, it is helpful to create a log file that keeps a record of the commands you enter and the output. To do this, use a command such as

```
log using exampleoutput.txt
```

which will create this file at the directory Stata tells you.

There are various ways to enter data or access a data file. See, for example,

`www.stata.com/manuals14/u21.pdf`

`www.ats.ucla.edu/stat/stata/modules/input.htm`

`www.ats.ucla.edu/stat/stata/notes/entering.htm`.

Your instructor can help you for the way your class will enter data. The text website has Stata data files (with extension *.dta*) for many examples and exercises. For example, to load the *Crime* data file that is used extensively in Chapter 3, you can enter the command

```
use "http://www.stat.ufl.edu/~aa/smss/data/Stata/Crime.dta"
```

Chapter 3: Descriptive Techniques

To form a histogram of a variable named y, use the command

```
histogram y
```

You can obtain basic descri
y with the command

If people find the manuals on the web via searching, that's great. Directing people to the web, however, is less useful, because the PDF docs come with the software. Also, trying to guess the URL is much harder for other commands than simply using the PDF docs.

```
summarize y, detail
```

For further options, see

`www.stata.com/manuals14/rsummarize.pdf`

To find the median and other percentiles, use the `centile` command. For example, for a variable called y, we get the quartiles by:

```
centile(y), centile(25, 50, 75)
```

You can find correlations for each pair of a set of variables with the `corr` command, such as

While this works, this is not standard Stata syntax, as it makes a command look like a function. It would be better to have

centile y, centile(25, 50, 75)

-pwcorr- is better to use than -corr-.

```
corr GDP GII Fertility
```

You can find the prediction line for a regression analysis with the `regress` command, such as

```
regress Fertility GDP
```

to predict Fertility using GDP.

Chapter 4: Probability Distributions

To find a normal cumulative probability for a particular z-value, use the `display normal(z)` command, such as

```
display normal(2.0)
```

to find the probability falling below 2.0 for a standard normal curve. To find the z-value having a cumulative probability p, use the `display invnormal(p)` command, such as

```
display invnormal(0.975)
```

to find the z value having cumulative probability 0.975 and thus right-tail probability 0.025.

Chapter 5: Estimation

To construct confidence intervals for mean and proportions, use the `ci` command. For example, for the mean of a variable called y,

```
ci y
```

It would help here to mention that there are dialog boxes for this, quickly accessible via
-db cii-
These are much easier to use than remembering obscure positional parameters.

If you already have sum... them, with the `cii` com... 3.007, $s = 7.309$):

```
cii 29 3.007 7.309
```

For the confidence interval for a proportion for a binary variable y that is a column in the data file that takes the values 0 and 1,

```
ci y, binomial wald
```

Or you can find it directly from the sample size and count in the category of interest, such as

```
cii 1200 396, wald
```

[5] Here, *i* following *ci* stands for *immediate*.

for the example in the text with 396 people out of $n = 1200$ sampled who favored restricting access to abortion. For further details and options, see

`www.stata.com/manuals14/rci.pdf`

The `mean` command also provides a confidence interval for the mean. For example, for the mean of a variable called y,

```
mean y
```

For further details and options, see

`www.stata.com/manuals14/rmean.pdf`

To find the t value having a cumulative probability p, use the `display invt(df, p)` command, such as

```
display invt(28, 0.975)
```

to find the t value having cumulative probability 0.975 and thus right-tail probability 0.025 when $df = 28$. To find a cumulative probability for a particular t value, use the `display tprob(df, t)` command, such as

```
display tprob(28, 2.0)
```

to find the probability falling below 2.0 for a t distribution with $df = 28$.

For information on using Stata for the bootstrap, see

`www.stata.com/features/overview/bootstrap-sampling-and-estimation`.

Chapter 6: Significance Tests

To conduct a t-test of whether a variable y in the data file has a mean of 0, use the `ttest` command:

```
ttest y == 0
```

If you already have summary statistics, you can use the `ttesti` command[6], by entering n, $\bar{y}$, s, and μ_0, such as for the text anorexia example:

```
ttesti 29 3.007 7.309 0
```

For further details and options, see

`www.stata.com/manuals14/rttest.pdf`.

To conduct a significance test of whether a categorical variable y that takes values 0 and 1 in the data file has a population proportion of 0.50 with the value 1:

```
prtest y == 0.50
```

[6]Here, *i* following *ttest* stands for *immediate*.

If you already have summary statistics, you can use the `prtesti` command, by entering n, $\hat{\pi}$, and π_0, such as for the example on page 210:

```
prtesti 1200 0.52 0.50
```

For further details and options, see

`www.stata.com/manuals14/rprtest.pdf`.

To find a right-tail probability for a particular t value with a certain *df* value, such as to find a one-sided P-value, use

```
display tprob(df, t)
```

to find the cumulative probability, and then subtract from 1.

This should be an incredibly rare usage nowadays, because it assumes that the observations within the two variables are totally unrelated. I wouldn't even show it, because it assumes a really bad dataset.

Chapter 7: Comparison of Two Groups

Stata can construct confidence intervals and tests comparing two proportions using the command `prtest`. To test equality of proportions between binary variables $y1$ and $y2$ that each take values 0 and 1 in a data file, you can use

```
prtest y1 == y2
```

This also shows the 95% confidence interval for the difference. To test equality of proportions for a variable y between two groups defined by a variable called *group* (such as gender), you can use

```
prtest y, by(group)
```

If you have summary statistics, you can find the inferences directly from the sample size and count in the category of interest for each group, such as

```
604 0.522 597 0.509
```

for the prayer example on pages 248, 250 and 252 in the text.

To construct inference for means, use the `ttest` command. For example, to test that the mean of a variable called y is equal between two groups defined by a categorical variable called *group*, use

```
ttest y, by(group)
```

to use the method of Section 7.5 that assumes $\sigma_1 = \sigma_2$. Use

```
ttest y, by(group) unequal
```

to allow unequal population standard deviations as in Section 7.3. The commands also yield confidence intervals comparing the group means.

If you already have summary statistics, you can conduct the inferences with the `ttesti` command, by entering n, $\bar{y}$, and s for each group, such as

```
ttesti 583 8.3 9.4 693 11.9 12.7, unequal
```

for the housework example on pages 255 and 256 in the text.

For the paired-difference t analyses with matched-pairs data in variables called $y1$ and $y2$, use

```
ttest y1 == y2
```

Alternatively, you can create a new variable of difference scores, and use the t methods described for Chapters 5 and 6. When $y1$ and $y2$ are binary, you can get McNemar's test using

```
mcc y1 y2
```

The output shows a *chi-squared statistic* that is the square of the z statistic we present in the text. The P-value for the chi-squared test is the two-sided P-value for the z statistic. Using the summary counts in the contingency table that cross classifies $y1$ and $y2$, you can get McNemar's test for the example on page 267 using

```
mcci 875 162 9 168
```

For two categorical variables $y1$ and $y2$, you can construct a contingency table and perform Fisher's exact test using the command

```
tab y1 y2, exact
```

You can enter the counts yourself from the contingency table that cross classifies $y1$ and $y2$ and request this test. For the example on page 270, we use

```
tabi 10 18 \ 1 22, exact
```

To conduct the Wilcoxon test with a response variable y and groups defined by a variable x, use

```
ranksum y, by(x) porder
```

The *porder* option requests an estimate of the probability that one group is higher than the other.

Chapter 8: Analyzing Association between Categorical Variables

With the `tabulate` command (`tab` for short), you can contruct contingency tables, find percentages in the conditional distributions (within-row relative frequencies), get expected frequencies for H_0: independence, get the chi-squared statistic and its P-value, and conduct Fisher's exact test. See

`www.stata.com/help.cgi?tabulate_twoway`

`www.stata.com/manuals14/rtabulatetwoway.pdf`

for a summary and a list of options. For categorical variables x and y in a data file, for instance, you can use

```
tab x y, row expected chi2 exact gamma
```

If you already have the cell counts, you can enter them by row. For the example on page 302, use

```
tabi 495 590 272 \ 330 498 265, row expected chi2 exact gamma
```

To get standardized residuals, you currently must download a routine written by Nicholas Cox. Within Stata, use the command

```
ssc install tab\chi
```

This should be tab_chi

then followed (if you have the cell counts) by

```
tabchii 495 590 272 \ 330 498 265, adjust
```

to get the standardized (adjusted) residuals.

The name of the dataset is "Crime2" (no space)

Chapter 9: Linear Regression and Correlation

The examples at UCLA use a very different dataset from that used in this book. This could be quite confusing to a reader.

For the use of Stata for regression analyses for the `Crime 2` data file analyzed in Chapters 9 and 14, see

`www.ats.ucla.edu/stat/stata/webbooks/reg/chapter2/statareg2.htm`

One can conduct a basic linear regression analysis for a response variable y and explanatory variable x with the command:

```
regress y x
```

For a scatterplot, use:

```
scatter y x
```

For the correlation, use the command

```
corr y x
```

It would help to tell the readers how to install the package, as you did above: ssc install corrci

One can get a confidence interval for the correlation with the package `corrci`, using the command

```
corrci x y
```

for variables x and y.

Chapter 11: Multiple Regression and Correlation

For a scatterplot matrix, use the command of form

```
graph matrix w x y
```

entering variable names such as w, x, y, e.g.,

```
graph matrix impair life ses
```

for the `Mental` data set at the text website that is analyzed in this chapter. To construct a correlation matrix, use the command of form

```
corr w x y z
```

One can conduct a basic linear regression analysis for a response variable y and explanatory variables with the command:

```
regress y x1 x2 x3
```

For a partial regression plot of the response variable y against each explanatory variable, follow this by the command `avplots` (*Added-variable plot* is an alternative name for partial regression plot).

I understand the point of this in the body of the book, because it illustrates the mathematical meaning of interaction terms. In an appendix illustrating the use of Stata, however, it introduces a bad habit, because there is no need to ever generate a new variable for an interaction.

You can add an interaction term to a model by generating the new cross-product variable

```
generate int12 = x1*x2
```

and then including it in a regression model statement, such as

```
regress y x1 x2 int12 x3
```

Would it help to show people the ## notation, since one typically never has interactions without their main effects?

Or, you can put a # between the variables and ad[...] that the variable is continuous (actually, merely q[...]

```
regress y x1 x2 c.x1#c.x2 x3
```

To get partial correlations (and semipartial correlations) of y with each explanatory variable, controlling for the others in the model, use a command of form

```
pcorr y x1 x2 x3
```

To obtain the standardized regression coefficients, use

```
regress impair life ses, beta
```

The name, and the heading *Beta* in the output, reflects the alternative name *beta weights* for these coefficients.

Two things:
The -i- in -ib- is not needed.
The terminology in the Stata manuals is 'base level', not 'baseline'. I wouldn't bet on it, but I believe that the Stata terminology is clearer, because 'baseline' usually refers to an initial measurement, not a base reference class.

Chapter 12: Regression with Categorical Predictors: Analysis of Variance Methods

To use the `regress` function with a categorical variable, declaring it to be a factor using the i. prefix to create indicator (dummy) variables, such as for a variable called *party*,

```
regress y i.party
```

The first category is deleted for the dummy variables. To instead use category 3 for the baseline, for instance, enter the categorical variable as *ib3.party* (for *ib* = indicator baseline).

For pairwise multiple comparisons of means for a factor called *party*, use

```
pwmean y, over(party) mcompare(bonferroni)
```

This is a bit dated. I would recommend
  regress y i.party
  pwcompare party
because -pwcompare- can be used after any type of regression with a categorical predictor.

substituting *tukey* for *bonferroni* to get the less conservative Tukey intervals.

To conduct a one-way ANOVA with a response variable y and a factor A, use the command

```
anova y A
```

The variable A is assumed to be categorical. One can also do one-way ANOVA with the `oneway` command,

```
oneway y A
```

For a two-way ANOVA with response variable y and factors A and B, without interaction, use

```
anova y A B
```

To allow for interaction, use

Or... better still
anova y A##B

```
anova y A B A#B
```

Entering `regress` after requesting an ANOVA fit yields the model fit for the corresponding regression model with dummy variables.

Alternatively, you can do a factorial anova by applying the `regress` function to the factors, declaring them to be factors using the i. prefix to create indicator (dummy) variables, such as

```
regress y i.A i.B
```

To conduct a repeated-measures ANOVA, the data must be in the "long" form with the repeated measurements on separate lines of the data file, as shown above in the R section for Chapter 12. If a data file has all observations for a subject on one row, one can use the `reshape` command in Stata to put it in the required form. For example, if a row of the data file showed all the observations for a particular person, with variable labels *trt*, *y1*, and *y2*, then use the command

```
reshape long y, i(person) j(time)
```

The one-way repeated-measures analysis of Se[...] tainment is obtained by

What does 'R section' mean?
Do you mean the [R] manual?
I'm confused.

```
anova y person type, repeated(type)
```

The two-way analysis of Section 12.6, for "long" data file as shown above in the R section, is obtained by

```
anova y group / person|group time group#time, repeated(time)
```

Chapter 13: Multiple Regression with Quantitative and Categorical Predictors

Stata can fit regression models having both quantitative and categorical explanatory variables using the `regress` function. Prefix a categorical factor with i. to specify indicators for each category of the variable, such as

```
regress y education i.race
```

By default, Stata

~~Stata~~ takes the first category as the baseline that does not have its own dummy variable.

To interact a quantitative variable with a categorical factor, prefix the quantitative variable with c. (for continuous), such as

Here, too, I would suggest

education##c.race instead of specifying the interaction terms by hand.

```
regress y education i.race c.education#i.race
```

Having fitted a model with no interaction, as just shown, one can follow-up with adjusted means by

```
margins i.race, at( (mean) education)
```

To fit the linear mixed model for the clustered family data, with a data file containing values for *family*, *y*, *x1*, and *x2*, use the command

mixed

-xtmixed- is out of date as of Stata 13.

```
xtmixed y x1 x2 || family:, residuals(ex, t(family)) reml
```

or else

```
xtmixed y x1 x2 || family:, covariance(exchangeable) reml
```

both of which yield an exchangeable structure for correlations within families for the model.

Chapter 14: Model Building with Multiple Regression

One can conduct automatic variable selection methods using the `stepwise` command. For backward elimination, with 0.10 as the α-level in tests, use a command such as (with five potential explanatory variables)

It's not really my place to say this, but -stepwise- is truly a bad thing to have, unless the user is (over)fitting on a portion of the dataset and testing on another portion. Stepwise regression on full datasets is a great way to get biased estimators.

```
stepwise, pr(0.10): regress y x1 x2 x3 x4 x5
```

where *pr* stands for the probability needed to be exceeded for removal. For forward selection, use

```
stepwise, pe(0.10): regress y x1 x2 x3 x4 x5
```

where *pe* stands for the probability needed to be below to be eligible for addition. The command

```
stepwise, pr(0.10) pe(0.10) forward: regress y x1 x2 x3 x4 x5
```

uses the stepwise variation of forward selection that removes a previously entered term if it is no longer significant.

After fitting a model with the `regress` command, to obtain the residuals and plot them against the model's fitted values, use

```
rvfplot, yline(0)
```

(Here, *rvf* stands for *residual-versus-fitted* plot.) Use `rvpplot` to plot them against a predictor x,

```
rvpplot x, yline(0)
```

Use the `predict` command with the `rstudent` option to generate the studentized residuals. Here, we name them r and then form a histogram and plot them against a predictor.

```
------------------------
. predict r, rstudent
. histogram r
. scatter r x1
------------------------
```

With the `dfbeta` and `dfits` commands, Stata will form DFBETA values for all the model parameters and DFFIT values for all the observations. Use DFBETA(x1) with the variable name in parentheses to inspect DFBETA values for a particular parameter.

After fitting a model, to assess multicollinearity you can obtain VIF values with the command

It would be better to use Stata's factor variables, and fit the model using

regress y x##x

because this will then give proper predictive margins and because it doesn't require using a derived variable.

```
vif
```

To fit GLMs, use the `glm` command. For details about models that can be fitted and options, see

`www.stata.com/manuals14/rglm.pdf`.

For instance, to fit a gamma regression model with the identity link, such as done in the text for the home selling price example, use a command such as

```
glm y  x1  x2, family(gamma) link(identity)
```

To fit a quadratic regression model, you can generate the new square variable

```
generate x2 = x^2
```

(or you can use *generate x2* $= x$**x*) and then include it in a regression model statement,

```
regress y x x2
```

You can fit exponential regression models by fitting the normal GLM with log link, using the `glm` command, such as by

```
glm y  x, family(gaussian) link(log)
```

Chapter 15: Logistic Regression: Modeling Categorical Responses

Stata can fit logistic regression models with the `logit` command or the `logistic` command. It can also fit them with the `glm` command, treating the model as a generalized linear model. For `logit`, the standard output is the model parameter estimates, whereas for `logistic` it is the odds ratios obtained by exponentiating the estimates. For example, the command for a binary response variable with three explanatory variables is

Do you see datasets like this in the wild, anymore? This seems to be a bit outdated.

```
logit y x1 x2 x3
```

Adding the *or* option to this command requests the odds ratio form of estimate.

If the data are counts in a contingency table, and each row of the data file has a value for each explanatory variable, the 0 or the 1 value for y, and a variable (say, called *count*) containing the cell counts, you can use the command

store

-save- has the connotation of saving to disk (as in -estimates save-). -store- means to store in memory, which is what -estimates store- does.

```
logit y x1 x2 x3 [fweight = count]
```

Here, $fweight = count$ indicates that the data file has data grouped according to the variable called *count*.

To do a likelihood-ratio test about an explanatory variable, ~~save~~ the results for the full model, fit the simpler model without that variable, and then request the likelihood-ratio test comparing the models. For example, to test the effect of defendant's race for the death penalty data of Table 15.3,

```
-----------------------------------------------------------------------
. logit y d v [fweight = count]
. estimates store full
. logit y v [fweight = count]
. lrtest full
-----------------------------------------------------------------------
```

A command such as

```
test race
```

conducts the Wald test about an explanatory variable (i.e., the square of the z test statistic), which is not as reliable a test as the likelihood-ratio test.

Probit models are fitted like logistic regression models, merely substituting *probit* for *logit* in the command. Propensity-score matching is obtained with the command `teffects psmatch`, such as

It might be helpful to say that -groups- can define 2 and only 2 groups here. From the wording, it looks like there could be any number of groups.

```
teffects psmatch (y) (group x1 x2 x3 x4)
```

to compare the groups identified by the variable *group* in their response on y after using logistic regression to get propensity scores for predicting *group* using $x1$, $x2$, $x3$, and $x4$.

Stata fits the cumulative logit model with the `ologit` (ordinal logit) command. If the data file contains grouped data (i.e., cell counts in the response categories), such as columns labeled *party* (a 1/0 indicator), *response* (giving the response category), *count*, fit the model with the command

Do you see datasets like this very often?

```
ologit response party [fweight = count]
```

Stata fits the baseline-category logit model with the `mlogit` (multinomial logit) command. If the data file contains grouped data (i.e., cell counts in the response categories), such as columns labeled *race*, *sex*, *response* (giving the response category), *count*, fit the model with the command

Again: this style of grouped data seems artificial.

Perhaps I'm missing something, but couldn't this be specified directly using the -poisson- command? -glm- has some extra postestimation tools, but they don't seem to be used in the book.


```
mlogit response sex race [fweight = count], base(3)
```

where *base(3)* indicates the baseline category.

Stata can fit loglinear models by regarding them as generalized linear models with response count having a Poisson distribution, using the log link. For example, for a 2×2×2 table such as Table 15.13 constructed from a data file with three columns of levels for the variables and a column of cell counts, fit the homogeneous association model with the command

```
glm count i.a i.c i.m i.a#i.c i.a#i.m i.c#i.m, family(poisson) link(log)
```

Chapter 16: Introduction to Advanced Methodology

Here is an illustration of how to conduct multiple imputation for the mental impairment data set discussed on page 680 in the text:

```
-----------------------------------------------------------------------
. use "http://www.stat.ufl.edu/~aa/smss/data/Stata/mental_missing.dta"
. regress impair life ses
. misstable summarize
. mi set mlong
. mi register imputed ses
(10 m=0 obs. now marked as incomplete)
. mi misstable summarize, all
. mi impute regress impair life ses, add(100)
. mi estimate: regress impair life ses
-----------------------------------------------------------------------
```

For details, see

`www.stata.com/manuals14/mi.pdf`.

For an example for logistic regression. watch the demonstration at

`www.youtube.com/watch?v=i6SOlq0mjuc`.

In the *Statistics* menu, the *Multilevel Mixed-Effects Models* suboption has many choices, including linear regression and GLMs. The *Survival Analysis* option and *Regression models* suboption has many choices, including the Cox proportional hazards model (with the `stcox` function). There is also a *SEM* (structural equation modeling) option.

INTRODUCTION TO SPSS

SPSS has a windows-with-menus structure that makes requesting statistical procedures simple. Our discussion below applies to version 23. It can help to look at the online manual at

`www.spss-tutorials.com`