ANALYSIS OF DISCRETE DATA
STATA CODES

1. LOGISTIC REGRESSION

Logistic regression: general form

Syntax:

    logit depvar [indepvars] [if] [in] [weight] [, options]

Options (select):

Standard errors/Robust:
    vce(vcetype): vcetype may be, for example, robust, cluster clustvar or bootstrap.

Reporting
    level(#): set confidence level; default is level(95).
    or: report odds ratios.

Other
    if at the end of the logit command means that the command will only use the data specified.
    in at the end of the logit command means that the command will only use the observations in the range specified.

Examples:

The following model regresses “fracture” (coded as 1 if a runner has a fracture and 0 otherwise) on calcium, dairy, and fiber intake.

    logit fracture calcium dairy fiber

The following model repeats the first regression, but reports odds ratios instead of log odds and 90% confidence intervals instead of 95% intervals.

    logit fracture calcium dairy fiber, or level(90)

Predicted probabilities

Syntax:

    predict new_var

Example:
Immediately after running a logit regression, *predict* creates a “new_var” with predicted probabilities. The following code does this for the *fracture* example.

```plaintext
/* Runs the logit model
logit fracture calcium dairy fiber
*/ Computes the predicted probabilities
predict fracture_predicted
```

**Predicted values of the dependent variable for specific values of independent variables**

*Syntax:*

```
margins, at(indep_var=value)
```

*Example 1:*

Immediately after running a logit regression, *margins* calculates the predicted probability of the dependent variable at specific values of the independent variables. In the following example, *margins* calculates the predicted probability of fracture if the calcium intake of a runner is 0.15:

```plaintext
/* Runs the logit model
logit fracture calcium dairy fiber
*/ Computes the predicted probability of fracture when calcium is 0.15
margins, at(calcium =0.15)
```

*Example 2:*

The following code repeats the example above, but calculates the predicted probability of *fracture* when all the independent variables are at their means.

```plaintext
/* Runs the logit model
logit fracture calcium dairy fiber
*/ Computes the predicted probability of fracture when the independent variables are at their means
margins, atmeans
```

**Predicted residuals**

*Syntax:*

```
predict new_var, residuals
```
**Example:**

Immediately after running a logit regression, `predict new_var, residuals` creates a “new_var” with the predicted residuals. The following codes illustrates this:

```plaintext
*/Runs the logit model
logit fracture calcium dairy fiber
*/Computes the predicted residuals
predict fracture_predicted_residuals, residuals
```

**ROC curve**

**Syntax:**

```plaintext
lroc
```

**Example:**

Immediately after running a logit model, `lroc` creates the ROC curve for the model. The following code does this for the runners example:

```plaintext
*/Runs the logit model
logit fracture calcium dairy fiber
*/Obtains the ROC curve
lroc
```

**Interactions**

To include an interaction in the logit model, first create the variable with the interacted terms and then run the logit model with that variable.

**Example:**

The following example creates an interaction between calcium and dairy, and then runs the logit model with that interaction:

```plaintext
*/Generates the interaction variable
gen calcium_dairy = calcium*dairy
*/Runs the logit with the interaction variable
logit fracture calcium dairy fiber calcium_dairy
```
2. PRINCIPAL COMPONENT ANALYSIS (PCA)

Note

In general, STATA performs PCA using the `pca` command. However, PCA can also be performed from the `factor analysis` module, which will yield results that are consistent with the SAS results in the class. The following commands therefore use the `factor analysis` module for PCA.

**Syntax:**

```
factor variables [, [method] [options[ ]
```

**Method (select):**

- `pcf`: performs principal component factor analysis.

**Options (select):**

- `factors(#)`: maximum number of factors to be retained.
- `mineigen(#)`: minimum value of eigenvalues to be retained.

**Example 1:**

In the following example, the `factor` module performs principal component analysis for eight variables: `aniprotein`, `calcium`, `dairy`, `fat`, `fiber`, `fruitveg`, `vegprotein`, and `vitc`. In this example, the command retains only principal components with an eigenvalue equal or greater than 0.95.

```
factor aniprotein calcium dairy fat fiber fruitveg vegprotein vitc, pcf mineigen(.95)
```

**Example 2:**

This example illustrates the entire process from (i) running PCA to (ii) obtaining a new set of vectors (“factors”) that are linear combinations of the initial variables. In forming these “factors”, the weight of each initial variable is given by the “scores” from the PCA analysis. In other words, the “factors” are computed by multiplying the matrix of “scores” by the matrix of the original variables.

```*/
Step 1: create a matrix ("orig_vars") with all the independent variables (this matrix will be used later)
```

```
mkmat aniprotein calcium dairy fat fiber fruitveg vegprotein vitc, matrix(orig_vars)
```

```*/
See the orig_vars matrix
```
matrix list orig_vars

*/ Step 2: perform PCA analysis (in this case, only principal components with an eigenvalue equal to or greater than 0.95 are retained)

factor aniprotein calcium dairy fat fiber fruitveg vegprotein vitc, pcf mineigen(.95)

*/ Step 3: apply the varimax rotation (using the kaiser option)

rotate, varimax kaiser

*/ Step 4: obtain the “scores” of each principal component after the rotation

predict scores

*/ Step 5: form “factors” or “component patterns” as a linear combination of the original variables. The weight of each factor is the “score” from Step 4.

matrix factors = orig_vars* r(scoef)

*/ See the resulting “factors”

matrix list factors

*/ Rename each column as “Factor” and the corresponding number of the factor

svmat factors, names(Factor)

The “factors” or “component patterns” formed from the above process can then be used to study whether and how they determine the outcome variable of interest (in this case, fracture). For example, it is possible to use a logit regression model to study the relationship between each factor and fracture (note that the “factors” are uncorrelated, so a multiple regression framework is not necessary):

*/ Runs a logit regression model of fracture on each “factor” obtained from the process above

logit fracture Factor1 Factor2 Factor3
3. PROPENSITY SCORE MATCHING

There are several algorithms and options to perform propensity score matching in STATA. The following codes only focus on the general methods discussed in class.

Direct method

Syntax:

psmatch2 dependent_variable independent_variables [, options]

Example:

In the following example, the primary outcome of interest is “MACE” (major adverse cardiovascular event), which is a binary variable. The main predictor of interest is “cilostazol”, which is also binary. To make the patients more comparable, they are first matched on age and gender:

psmatch2 cilostazol age gender, logit

The propensity score obtained from match2 can then be used in several ways to study the relationship between the independent variables and MACE. One way, for example, is to weight the logit regression of MACE on the independent variables by the propensity score [weighted regressions in this form are not discussed in class; this example is only for illustration purposes]:

logit MACE cilostazol age gender [weight=_pscore]

Another way to use the propensity score (discussed in class) is to include it as a covariate in the regression of the outcome of interest (in this case, MACE) on the other covariates. For example, the model could take the following form:

logit MACE cilostazol _pscore

Indirect method

The propensity score can also be obtained by (i) running a logit model of the treatment variable, and then (ii) obtaining the predicted probabilities. For example, the following commands compute propensity scores similar to the propensity scores obtained from the psmatch2 command in the example above:

*/ Runs the logit model
logit cilostazol age gender

*/ Computes the predicted probabilities (labeled here as "propensity scores")
predict psscore
4. **BOOTSTRAP REGRESSION**

**Syntax**

```bash
bootstrap exp_list [, options eform_option] : command
```

The `bootstrap` code executes `command` multiple times, bootstrapping the statistics in `exp_list` by resampling observations (with replacement) from the data.

`command` defines the statistical command to be executed (for example, logit regression).

`exp_list` specifies the statistics to be collected from the execution of `command`.

In order to be able to reproduce the results, set the random-number seed by specifying the `seed(#)` option.

**Options (select):**

**General**
- `reps(#)`: perform # bootstrap replications; default is reps(50).
- `strata(varlist)`: variables identifying strata.
- `size(#)`: draw samples of size #.
- `cluster(varlist)`: variables identifying resampling clusters.
- `idcluster(newvar)`: create new cluster ID variable.
- `saving(filename, ...)`: save results to filename; save results to filename every # replications.
- `seed(#)`: set random-number seed to #.

**Reporting**
- `level(#)`: set confidence level; default is level(95).
- `eform`: report exponentiated coefficients.

**Example:**

```bash
*/ Computes bootstrap estimates for a logit regression of fracture on dairy using 200 replications.

bootstrap, reps(200) : logit fracture dairy
```
5. AUTOMATED SELECTION: STEPWISE REGRESSION (FORWARD AND BACKWARD)

Syntax:

\texttt{stepwise [, options] : command}

Options (select):

- \texttt{pr(#)\textbackslash : significance level for removal from the model.}
- \texttt{pe(#)\textbackslash : significance level for addition to the model.}
- \texttt{forward\textbackslash : perform forward-stepwise selection.}
- \texttt{lockterm1\textbackslash : keep the first term.}

Example:

- /* Performs backward selection, using 0.10 significance as the cutoff for removal from the model */
  
  \texttt{stepwise, pr(.10): logit fracture aniprotein calcium dairy fat fiber fruitveg vegprotein vitc}

- /* Performs forward selection, using 0.10 significance as the cutoff for addition to the model */
  
  \texttt{stepwise, pe(.10): logit fracture aniprotein calcium dairy fat fiber fruitveg vegprotein vitc}

6. MULTINOMIAL, ORDINAL LOGISTIC AND POISSON REGRESSION

6.1. MULTINOMIAL LOGISTIC

Syntax:

\texttt{mlogit depvar [indepvars] [if] [in] [weight] [, options]}

Options (select):

General
- \texttt{baseoutcome(#): value of depvar that will be the base outcome.}

SE/Robust
- \texttt{vce(vcetype): vcetype may be, for example, oim, robust, cluster clustvar or bootstrap.}

Reporting
- \texttt{level(#): set confidence level; default is level(95).}
- \texttt{rrr: report relative-risk ratios [odds ratios].}
Example:

In the following example, the dependent variable is the hematologic profile of an athlete, which can take three levels: highly abnormal, abnormal, and normal. This outcome variable is therefore ordinal and was coded as profile=3 (for normal), profile=2 (for abnormal), and profile=1 (for highly abnormal). The independent variable is EDIA (0-21 eating disorder scale). If the data are grouped (like in this example), the regressions must be weighted by count. In this case, the regression output reports the odds ratio.

* Runs the multinomial regression model, weighting by count
  mlogit profile edia [weight=count]

* Runs the multinomial regression model, weighting by count and reporting odds ratios
  mlogit anyproblem edia [weight=count], rrr

6.2. ORDINAL LOGISTIC

Syntax:

ologit depvar [indepvars] [if] [in] [weight] [, options]

Options (select):

SE/Robust
  vce(vcetype): vcetype may be, for example, oim, robust, cluster clustvar or bootstrap.

Reporting
  level(#): set confidence level; default is level(95).
  or: report odds ratios.

Example:

Same example as above (multinomial regression), but this code runs an ordinal model instead of the multinomial. The regression is weighted by count and reports odds ratios.

ologit anyproblem place [weight=count], or

6.3. POISSON REGRESSION

Syntax:

poisson depvar [indepvars] [if] [in] [weight] [, options]

Options (select):
SE/Robust
vce(vcetype): vcetype may be, for example, oim, robust, cluster clustvar or bootstrap.

Reporting
level(#): set confidence level; default is level(95).
irr: report incidence-rate ratios.

Example:

In this example, the dependent variable is the number of athletes that had a “highly abnormal” hematologic profile. Each row shows that number and the place of those athletes in a given race. The independent variable is therefore the position of the athletes in the race.

```
poisson hiab place
```

In the data were grouped, it would be necessary to weight by count. The following code illustrates this, and also displays incidence risk ratios:

```
poisson hiab place [weight=count], irr
```

7. CONDITIONAL LOGISTIC AND GENERALIZED ESTIMATING (GEE) MODELS

7.1. CONDITIONAL LOGISTIC

Syntax:

```
clogit depvar [indepvars] [if] [in] [weight], group(varname) [options]
```

Options (select):

Model
group(varname): matched group variable.

SE/Robust
vce(vcetype): vcetype may be, for example, oim, robust, cluster clustvar or bootstrap.

Reporting
level(#): set confidence level; default is level(95).
or: report odds ratios.

Example:

This example is based on a hypothetical randomized trial in which thirty doctors were
randomized to receive either an electronic decision-support system or a control condition (n=15 per group). There are 20 patients per physician.

The outcome variable is InsControl (a binary variable set to 1 for “good insulin control” and 0 for inadequate insulin control). The DoctorId variable identifies the treating doctor; Treatment = 1 for decision-support system and 0 for the control; age is the patient’s age in years; and female is 1 for females and 0 for males. The code in this example reports odds ratios:

clogit inscontrol age female, group(doctorid) or

7.2. GEE

Syntax:

    xtgee depvar [indepvars] [if] [in] [weight] [, options]

In GEE regression, a panel variable must be specified. Correlation structures other than exchangeable and independent require that a time variable also be specified (using xtset varname).

Options (select):

Model
    family(family): distribution of depvar (for dealing with dichotomous dependent variables, the family is binomial).
    link(link): link function.

Correlation
    corr(correlation): within-group correlation structure.

SE/Robust
    vce(vcetype): vcetype may be, for example, oim, robust, cluster clustvar or bootstrap.

Reporting
    level(#): set confidence level; default is level(95).
    eform: report exponentiated coefficients.

Family options:
    gaussian: Gaussian (normal); family(normal) is a synonym.
    igaussian: inverse Gaussian.
    binomial[#|varname]: Bernoulli/binomial.
    poisson: Poisson.
    nbinomial[#]: negative binomial.
    gamma: gamma.

Link options:
identity: identity; y=y.
log: log; ln(y).
logit: logit; ln{y/(1-y)}, natural log of the odds.
probit: probit; inverse Gaussian cumulative.
cloglog: cloglog; ln{-ln(1-y)}.
power[#]: power; y^k with k=#; #=1 if not specified.
opower[#]: odds power; [{y/(1-y)}^k - 1]/k with k=#; #=1 if not specified.
nbinomial: negative binomial.
reciprocal: reciprocal; 1/y.

**Correlation structures:**
exchangeable: exchangeable.
independent: independent.
unstructured: unstructured.
fixed matname: user-specified.
ar #: autoregressive of order #.
stationary #: stationary of order #.
nonstationary #: nonstationary of order #.

**Example 1:**

Similarly to the example for conditional logit regression above, this example is based on a hypothetical randomized trial in which thirty doctors were randomized to receive either an electronic decision-support system or a control condition (n=15 per group, and 20 patients per physician). The outcome variable is InsControl (a binary outcome variable set to 1 for “good insulin control” and 0 for inadequate insulin control). The DoctorId variable identifies the treating doctor; Treatment = 1 for decision-support system and 0 for the control; age is the patient’s age in years; and female is 1 for females and 0 for males.

The following code first sets the “panel” (or “grouping”) variable, and then runs the regression. Because the dependent variable is dichotomous, the family option is binomial. In this particular example, the link function is logit and the correlation structure is exchangeable.

/* Sets the “panel” or “grouping” variable

xtset doctorid

/* Runs the GEE regression

xtgee inscontrol treatment age female, family(binomial)
link(logit) corr(exchangeable)

**Example 2:**

This example is similar to the example above, but instead of using regular standard
errors, it uses robust standard errors and an independent correlation structure. Because the robust standard errors are sensitive to the correlation structure, the estimations are appropriate even if the correct correlation structure was not independent.

/* Sets the “panel” or “grouping” variable

xtset doctorid

*/ Runs the GEE with independent correlation and robust standard errors

xtgee inscontrol treatment age female, family(binomial) link(logit) corr(independent) vce(robust)