Guide to Computing
Minitab commands used in labs (mtbcode.out)

A full listing of Minitab commands can be found by invoking the HELP command while running Minitab. A reference card, with listing of available commands, can be purchased in the University Bookstore.

Use the HELP command (or reference card) much as you would a dictionary or thesaurus, to find out how a command works or find a command to accomplish the calculation at hand.

The commands listed here are for common routines used in this course:

Reading Data into Minitab
Writing Data and Output Files from Minitab
Summarizing data
Re-organizing Data for Analysis by the General Linear Model
Executing the General Linear Model
Calculating Residuals
Using Residuals to Check Assumptions
Randomization Tests with Minitab

These routines are built up out of commands, just as sentences are built out of words. Most of the assignments in this course can be accomplished by modifying one of the following routines, depending on the situation or goals of computation.

Many routines have been assigned a name, which is used in other parts of the lab manual. The name of the routine occurs in ***boldface italics*** to the right of the routine box.

Minitab commands

Reading Data into Minitab.

Data are read from a file (in quotes) into columns. For the data files in this course, you will need to state the number of lines of data, because information about the data is listed below the data in the same file.

```
MTB > read 'srbx9_5.dat' c1 c2;
SUBC> nobs = 7.
MTB > name c1 'age(I)' c2 'age(II)'
```

Define Data from file

Data are also typed directly from the keyboard.

```
MTB > set into c1
DATA> 2.68 2.60 2.43 2.90 2.94 2.70 2.68 2.98 2.85
DATA> end
MTB > set c2
DATA> 2.36 2.41 2.39 2.85 2.82 2.73 2.58 2.89 2.78
DATA> end
MTB > name c1 'N(B)' c2 'N(13)'
```

Define Data from keyboard

Writing Data and Output Files from Minitab

Data in columns can be saved by writing to a named system file, as in the box above.

```
MTB > write [to] 'srbx1311.dat' c1 c2
```

Define Data to file

This command prints output directly to the screen.

```
MTB > print c1 c2
```

Output to screen

Most commands send output to the screen automatically.

```
MTB > describe c1 c2
MTB > histogram c1 c2
```

Results of computations need to be printed to the screen

```
MTB > let k2 = std(c1)
MTB > let k2 = k2*k2
MTB > print k2
```

Compute a variance
Output to screen

Output that appears on the screen can be written to a named system file, this file can then be printed out, or moved to another computer.

```
MTB > outfile 'srbx9_5.out'
MTB > print c1 c2
MTB > describe c1 c2
MTB > nooutfile
```

Outfile open

Outfile closed

Summarizing Data

The following 4 commands calculate and display descriptive statistics on data in column 1

```
MTB > describe c1
MTB > histogram c1
MTB > dotplot c1
MTB > mean c1
```

This next routine calculates a cumulative relative relative frequency
CRF(C1 < 2) = 0.09
from the following data:

```
1  2  3  3  4  4  4  5  5  6  7
```

CRF(C1 < 2)

```
MTB > histogram c1;
MTB > start 2.
MTB > let k1 = 1                     <--from screen display
MTB > let k2 = 11                    <--from screen display
MTB > let k3 = k1/k2
MTB > print k3
```

Re-organizing Data for Analysis by the General Linear Model

The following commands reorganize data from tabular format (7 columns) to model format (2 columns, response and explanatory variable).

Reorganize

```
MTB > read 'srtab8_1.dat' c1-c7;
SUBC> nobs = 5.
MTB > stack c1-c7 c8
MTB > name c8 'wlength'
MTB > set c9
DATA> (1 2 3 4 5 6 7)5
DATA> end
MTB > name c9 'groups'
MTB > print c8 c9
```

Minitab commands

Executing the General Linear Model, Calculating Residuals

ANOVA designs

```
MTB > anova 'wlength' = 'groups';
SUBC> fits c10;
SUBC> residuals c11.
MTB > name c10 'fits' c11 'res'
```

Run GLM ANOVA

```
MTB > glm 'wlength' = 'groups';
SUBC> fits c10;
SUBC> residuals c11.
MTB > name c10 'fits' c11 'res'
```

Run GLM ANOVA

Regression designs

```
MTB > read 'ryder.dat' c1 c2;
SUBC> nobs = 23.
MTB > name c1 'area'  c2 'yield'
MTB > regress 'yield' 1 predictor 'area';
SUBC> residuals c10.
MTB > name c10 'res'
MTB > let k1 = 837382                      <---from screen display
MTB > let k2 = 1.45                        <---from screen display
MTB > let c11 = k1 + k2*'area'
MTB > name c11 'fits'
```

Run GLM Regression

```
MTB > read 'ryder.dat' c1 c2;
SUBC> nobs = 23.
MTB > name c1 'area'  c2 'yield'
MTB > glm 'yield' = 'area';
SUBC> covariate 'area';
SUBC> fits c9;
SUBC> residuals c10.
MTB > name c9 'fits' c10 'res'
```

Run GLM Regression

Using Residuals to Check Assumptions

A. linear relation of response to explanatory ? (bowls and arches)

```
MTB > plot 'res' vs 'fits'
```

GLM linear?

B1. Do errors sum to zero ?

Note that this assumption is listed for completeness. There is no need to check it if fitted values are calculated via least squares, as in statistical packages such as Minitab.

```
MTB > let k1 = sum('res')
MTB > print k1
```

B2. Are errors independent ?

```
MTB > let c20 = lag('res')
MTB > plot c20 'res'                          <--graphical analysis

MTB > runs 'res'                              <--runs test (optional)

MTB > corr c20 'res'            <--autocorrelation, lag 1(optional)

MTB > acf 'res'             <--autocorrelation, many lags (optional)
```

Errors independent?

B3. Are the errors homogeneous ?
(no cones facing left or right)

```
MTB > plot 'res' vs 'fits'
```

Errors homogeneous?

B4 Are the errors normally distributed ?

```
MTB > hist 'res'                              <---graphical analysis
MTB > nscores 'res' c30

MTB > plot c30 'res'                      <--- 2nd graphical analysis

MTB > rootogram 'res'                   <---3rd graphical analysis,
                                           shows confidence limits
```

Errors normal ?

Randomization Tests with Minitab

Statistic is mean (compared to zero)

```
MTB > let k1 = mean(c1)
```

Calculate statistic

```
MTB > nooutfile
MTB > set into c2
DATA> -1 1
DATA> end
MTB > let k2 = 0
```

Set up

```
MTB > store 'ran.ctl'
STOR> let k2 = k2 + 1
STOR> sample 12 c2 c3;
STOR> replace.
STOR> let c4 = c1*c3
STOR> let k3 = mean(c4) - k1
STOR> let c5(k2) = k3
STOR> end
```

Create control file

```
MTB > execute 'ran.ctl'
MTB > execute 'ran.ctl'  1000
```

Check file

Run file

```
MTB > outfile
MTB > histogram c5;
SUBC> start k1.
```

CRF(c5 > k1)

p-value is % of distribution greater than statistic in k1

Randomization Tests with Minitab

Statistic is difference between two means.

```
MTB > let k10 = mean(c6) - mean(c7)
MTB > print k10
MTB > let c25(1) = k10
MTB > name c25 'F(st)'
```

Calculate statistic

```
MTB > nooutfile
```

set up

```
MTB > store into 'srbx9_5.ctl'
STOR> stack c1 c2 into c3;
STOR> subscripts c4.
STOR> sample 14 times from c3 into c5
STOR> unstack c5 c6 c7;
STOR> subscripts c4.
STOR> let k3 = mean(c6) - mean(c7)
STOR> stack c25 k3 into c25
STOR> end
```

Create control file

```
MTB > execute 'srbx9_5.ctl'
MTB > print 'F(st)'
MTB > execute 'srbx9_5.ctl' 10 times
MTB > print 'F(st)'
MTB > execute 'srbx9_5.ctl' 500
```

Check file

Run file

```
MTB > outfile
MTB > hist 'F(st)'
MTB > hist 'F(st)';
SUBC> start k10.
```

CRF(c25 > k10)

p-value is cumulative relative frequency (%) of outcomes in c25 that are **larger** than value in k10   CRF(c25 > k10) = %

Randomization Tests with Minitab

Statistic is mean squared error (MSE)

```
MTB > regress 'wloss' on 1 predictor 'humidity';
SUBC> residuals c5.
MTB > name c5 'res'
MTB > let k10 = std('res')**2
MTB > print k10
```

Calculate statistic

```
MTB > let c25(1) = k10
MTB > name c25 'F(st)'
MTB > nooutfile
```

Set up

```
MTB > store into 'srbx14_1.ctl'
STOR> sample 9 times from 'wloss' into c8
STOR> regress c8 1 'humidity';
SUBC> residuals c5.
STOR> let k3 = std('res')**2
STOR> stack c25 k3 into c25
STOR> end
```

Create control file

```
MTB > execute 'srbx14_1.ctl'
MTB > print 'F(st)'
MTB > execute 'srbx14_1.ctl' 10 times
MTB > print 'F(st)'
MTB > execute 'srbx14_1.ctl' 500
```

Check file

Run file

```
MTB > outfile
MTB > hist 'F(stat)'
MTB > hist 'F(stat)';
SUBC> start k10.
```

CRF(c25 < k10)

p-value is cumulative relative frequency (%) of outcomes in c25 that are **smaller** than value in k10. CRF(c25 < k10) = %