

## Stata survival commands for Moore/McCabe Assignments.

See 'findit' to search for and download user-written commands; 'encode x, gen(newx)' to destring a variable that should be quantitative or 'destring x, gen(newx)' for easier use of a string (i.e. text) variable; 'input' to create a small data set; 'gen id=\_n' to create an id variable; 'robmean' to obtain a trimmed mean; 'centile' to form deciles; gen z=uniform(), n(...) to create a data set of specified  $n$  with standard norm distribution; 'lincom' to obtain predicted values of  $y$  ( $\hat{y}$ ) after a regression equation; and 'egen std()' to create a standardized variable.

To save a work session's commands (without creating a command file or log file): right-click the Review window, then click either 'Save All' or 'Save Selected.'

To use the Command window's calculator: display 1+1 [enter]  
di (2+5)/(1+1) [enter]

### Chapter 1

```
use ex01_017.dta, clear
describe
su glucose, detail
stem glucose
stem glucose, prune
```

```
use ta01_007.dta, clear
desc
su pasadena reading, d [d: abbreviates 'detail']
hist pasadena, norm
gr bar pasadena, norm
hist reading, norm
gr bar reading, norm
tway line pasadena reading year
tway line pasadena reading year, ylabel(0(25)100) xlabel(1950(25)2000)
```

```
use ex01_029.dta, clear
d [d: abbreviates 'describe']
stem dtscore
dotplot dtscore
histogram dtscore, norm
kdensity dtscore, norm
gr box dtscore
qnorm dtscore
```

```
use ta01_009.dta, clear
d
tab gender
su concept, d
stem concept
bys gender: su concept
bys gender: stem concept
```

```
use ta01_005.dta, clear
d
su damage, d
hist damage, norm
gr box damage
```

```
use ex01_060.dta, clear
d
su trees, d
hist trees, norm
gr box trees
bys group: su trees, d
hist trees, by(group) norm
gr box trees, over(group)
```

```
u ta01_008.dta, clear
d
su days, d
hist days, norm
gr bar days
help centile
centile days, centile(20 40 60 80)
```

```
u ta01_008.dta, clear
d
su days, d
hist days, norm
gr bar days
help robmean
robmean days, trim(.1)
```

['robust mean']

```

ex01_035.dta, clear
d
su study, d
tab sex
stem study
stem study if study<300
bys sex: su study, d
bys sex: stem study
bys sex: stem study if study<300

```

```
di (25000+60000+255000)/3
```

```
display norm(1.85)
di invnorm(.96784)
```

[di: abbreviates 'display']

## Chapter 2

```

use ta01_010.dta, clear
d
tab type
tab type if type=="M"
destring type, gen(typeq)
"..."
tab1 type typeq, nolabel

```

['type' is coded as a string (text) variable]

[By destringing type, you don't have to use

[or: codebook type typeq]

Ex 2.29

```
input x
```

```
66
```

```
64
```

```
66
```

```
65
```

```
70
```

```
65
```

```
End
```

[Repeat using y data]

```
hist x, norm
```

```
hist y, norm
```

```
su x y, d
```

```
scatter y x || qfit y x
```

```
corr x y
```

*Note:* If the text-coded variable should be treated as quantitative, then instead of 'destring' use 'encode varname, gen(newx)'.

Ex 2.53

```
su rate mass, d
bys sex: su rate mass, d
hist rate, by(sex, total) norm
hist mass, by(sex, total) norm
gr box rate, over(sex, total)
gr box mass, over(sex, total)
scatter rate mass, ml(sex) || qfit rate mass
bys sex: scatter rate mass || qfit rate mass
corr rate mass
bys sex: corr rate mass
reg rate mass
bys sex: reg rate mass
```

Ex. 2.57

```
hist length if run==8305, norm
gr bar length if run==8305
hist time if run==8305, norm
gr bar time if run==8305
su length time if run==8305, d
scatter length time if run==8305 || qfit length time if run==8305
corr length time if run==8305
reg length time if run==8305
```

```
gen lengthc=length*2.54
su length lengthc, d
hist length, norm
gr bar length
hist lengthc, norm
gr bar lengthc
scatter length time || qfit length time
scatter lengthc time || qfit lengthc time
corr lengthc time
reg lengthc time
```

Ex. 2.8

```
use ta01_010.dta, clear
d
scatter hwy city if hwy~=66 & type=="T", mcolor(red) mlabel(type) msymbol(+) ||
scatter
hwy city if hwy~=66 & type=="M", mc(blue) ml(type) ms(o) || qfit hwy city if
hwy~=66
```

*Alternatively (and easier):*

```
encode type, gen(typeq) [To destring type]
tab1 type typeq, nolabel [or: codebook type typeq]
scatter hwy city if hwy~=66 & type==1, mc(red) ml(type) ms(+) || ..... & type==2,
mc(blue) ml(type) ms(o) || .....
```

### Stata calculator:

display 1+2  
di (5-2)/2  
di 64\*91  
di (47+4)/(8\*99)

### Getting help, finding/downloading user-written commands:

help tabulate  
findit placevar [Click blue text to download]  
findit renvars  
findit univar  
findit robmean

#### *Open data set:*

use dataname, clear

#### *Arrange order of variables:*

move varname1 varname2 [Move varname1 to varname2's position]  
aorder [alphabetical order]  
placevar varnames, f [Move to top of list -first]  
placevar varnames, l [Move to end of list – last]  
placevar varname(s), b(varname )  
placevar varname(s), a(varname)

#### *Changing case of variable names:*

renvars varnames, up [Stata varnames are case sensitive]  
renvars varnames, lo

#### *Format statistical output to one or two decimal places:*

tabstat varnames, s(mean sd min max) format(%9.2f) ['format': option for many procedures]

#### *Describing/summarizing data:*

desc  
d varname(s)  
su  
su varname(s)  
su varname(s), detail [5-number summary]  
sort varname [lowest to highest values]  
list  
l varname(s)  
l in 1/10  
l in -10/l ['l': last]  
l varname(s) in 25/50  
codebook varname(s)  
inspect varname(s)

*Checking spreadsheet:*

```
browse  
br in -20/1  
br varname(s)
```

```
edit  
edit varname(s)
```

[But don't use 'edit' in real life!]

*Input a little data:*

```
input x  
23  
64  
end
```

```
su x  
l x  
replace x=21 if x==23  
l x
```

[l: list]

[or: replace x=21 in 1 (i.e. in obs 1)]

*How to destring a string (text) variable:*

```
encode sex, gen(gender)  
tab gender
```

*Log & command log:*

```
log using hw1, replace [Or 'append']  
log off [Suspend log]  
log on [Restart log]  
log close [End log]
```

File>View>(select log file)

[To display. Edit in MS-Word, but save as plain text.]

```
cmdlog using hw1, replace [Or 'append']  
cmdlog off  
cmdlog on  
cmdlog close
```

[Use MS-Word to obtain/edit cmdlog. Save edits as plain text.]

*Do-file commands:*

```
version 10.0  
capture log close  
set more off  
log using hw1, replace [or 'append']  
use data_name, clear
```

```
su  
log close  
exit
```

[and other commands]

*Label dataset:*

label data "Homework #1, September 2"

*Create notes:*

[Right-click in the variables window to create notes attached to the data set in general or attached to a specific variable.]

note: TS I need help.  
note

[TS: time-date stamp]  
[Type in command window to display notes]

*Creating/transforming/labeling/renaming/'noting' variables:*

generate id=\_n  
list id

[Create an ID variable called id]

[Right-click in the variables window to create labels for variables]

compress [Type in command window for more efficient data storage]

generate total=test1+test2+test3

su total, d

kdensity total, norm

gr box total

[Right-click in variables window to rename 'total' to 'totscore ']

[Right-click in variables window to label totscore]

[Right-click in variables window to make note] totscore: TS gen

totscore=test1+test2+test3.

compress

note

[Type in command window to display all notes]

note totscore

[Display this variable's notes only]

note totscore drop in 1

[Drop totscore note #1]

*Log transformation:*

gen logx=ln(x)

su x logx

for varlist x logx: kdensity X, norm \ more

la var logx "log of x"

note logx: TS gen logx=ln(x)

*Square the value of a quantitative variable:*

gen xsq=x^2

su x xsq

for varlist x xsq: kdensity X, norm \ more

for varlist x xsq: gr bar X \ more

la var xsq "X squared"

note xsq: TS gen xsq=x^2

compress

*Create a categorical variable from a quantitative variable:*

```

gen catvar=varname>=50 & varname<.
tab catvar
la def lh 0 "<50" 1 ">=50 & <."
la value catvar lh
tab catvar
note catvar: TS gen catvar=varname>=50 & varname<.
compress

```

*Trimmed mean:*

```
robmean math, trim(.1) [set 'trim' option]
```

*Tabulation:*

```

tabulate catvar
tab catvar, nolabel [nol: displays numerical coding]
tab catvar, plot
tab1 catvar1 catvar2
tab catvar1 catvar2
tab catvar1 catvar2, col chi2
tab catvar1 catvar2, row chi2
tab catvar1 catvar2, cell chi2
tabi 12 17 \ 23 11 \ 26 41 [immediate' option]
tabi 12 17 \ 23 11 \ 26 41, col chi2

```

*Graphs:*

```

stem varname [Automatically rounds non-integer data]
dotplot varname
graph box varname
histogram varname, norm
kdensity varname , norm
qnorm varname
gr box varname, ylabel(0(10)100) xlabel(0(10)100) [Set y & x graph scales]
bys catvar: stem varname [By a categorical variable]
dotplot varname, over(catvar)
hist varname, norm by(catvar, total)

```

*Bar & pie graphs:*

```

hist catvar, discrete freq gap(10) [A bar graph, not a histogram]
hist catvar, disc frac gap(10)
gr bar varnames, percent

```

```

gr bar varnames, perc legend(order(4 3 2 1)
label(1 "Var4") label(2 "Var3") label(3 "Var2")
label(4 "Var1")) [How to reorder bar display &
and make legend]
gr bar varname, over(catvar)

```

```

tab catvar, gen(catvars) [Creates a categ var for each category of catvar]
desc catvars*
gr pie catvar1-catvar4

```

*Line graph:*

```
use ta01_007, clear
tway line pasadena year
tway line pasadena year, ylab(0(25)100) xlab(1950(25)2000)
```

*Scatterplots:*

```
scatter varname1 varname2
scatter varname1 varname2, mlabel(id)
scatter varname1 varname2 || lfit varname1 varname2
scatter varname1 varname2, mlabel(id) || lfit varname1 varname2
scatter varname1 varname2 || qfit varname1 varname2
scatter varname1 varname2, by(catvar, total)
scatter varname1 varname2, by(catvar, total) || qfit varname1 varname2

scatter varname1 varname2 if varnamen2<97
scatter varname1 varname2 if catvar==1
```

*Summarize a quantitative variable by a categorical variable:*

```
tab catvar, su(varname)
tabstat varnames, by(catvar) stats(mean sd median min max n)
tabstat varnames, by(catvar) stats(mean sd median min max n) format(%9.2f)
tabstat varnames, by(catvar) stats(mean sd median min max n) format(%9.2f)
    col(stats)
tabstat varnames, by(catvar) stats(q min max n) format(%9.1f) col(stats)
```

*Draw a random sample:*

```
sample 10                                [% sample]
bys catvar: sample 10
sample 10, count                          [count sample]
bys catvar: sample 10, count
sample 10 if catvar==0
sample 10 if catvar==0, count

bsample    [as above: sampling with replacement; use, for example, to draw
            repeated samples from the same data set]
```

*Create a variable from scratch with a standard normal distribution, and create an id variable:*

```
clear
set obs 500
set seed 123                                [To permit replication]
gen z=uniform(), n(2000)
kdensity z, norm
su z, d
gen id = _n                                [Create id variable]
list id
```

*Standardize a variable:*

```
egen zvarname=std(varname) [egen: 'extended' generate]
su varname zvarname
for varlist varname zvarname: kdensity X, norm \ more
```

*Using Stata calculator for standard normal distribution:*

```
di norm(2.5) [% of observations in standard normal distribution < standardized 2.5]
di invnorm(.99379) [standardized value in standard normal distribution > .99379 of
the obs]
```

*Correlation, regression:*

```
graph matrix varnames [Put dependent variable last.]
gr matrix varnames, half
```

```
correlate varnames
bys catvar: corr varnames
corr varname varname if varname~=98 [or: != ]
corr varname varname if varname<98
corr varname varname if catvar==0
```

*regress dvarname ivarname*

```
predict e [e': residual]
list e dvarname ivarname
rvfplot, yline(0)
lvr2plot, ml(id)
```

```
reg dvarname ivarname
lincom ivarname*45
lincom invarname*45 + invarname*56
lincom _cons + invarname*45
lincom _con + invarname*45 + invarname*56
regress dvarname ivarname if ivarname~=22
reg dvarname ivarname if ivarname<22
reg dvarname ivarname if ivarname>22
reg dvar ivar ivar*ivar [interaction]
reg dvar ivar ivarsq [quadratic]
```

*Obtain a confidence interval:*

```
ci varname [.95 default]
ci varname, level(90)
ci bivar, bi ['binomial exact']
ci bivar, bi w ['wilson' option]
ci bivar, bi w l(99)
ci bivar, bi a ['agresti' option]
```

*One-sample tests:*

```
ttest varname=63
ttest varname=63, l(99)
```

```
signtest varname=63
ttesti obs mean sd val
ttesti 1000 63 17.4 51, l(90)
ttest postvar=prevar
ttest postvar=prevar, l(90)
signtest postvar=prevar
ttesti obs1 mean1 sd1 obs2 mean2 sd2
ttesti 1000 20 5 943 15 11
ttesti 1000 20 5 943 15 11, l(99)
```

[nonparametric]  
[‘immediate’ form]

```
bitest bivar=.5, detail
bitest bivar, d l(90)
bitesti N #successes p, detail
bitesti 1000 47 .23, d
prtest postbivar=prebivar
prtest postbivar=prebivar, l(99)
prtest1 N p1 p2
prtesti 1000 .23 .29
```

[See also prtest]

*Two-sample tests:*

```
ttest varname, by(bivar)
ttest varname, by(bivar) l(99)
ttest varname, by(bivar) unequal
ranksum varname, by(bivar)
ttesti N1 sd1 mean1 N2 sd2 mean2
ttesti 1000 14 71 1120 11 64
ttesti 1000 14 71 1120 11 64, l(99)
prtest bivar, by(bivar)
prtest bivar, by(bivar) l(90)
prtesti N1 p1 N2 p2
prtesti 1000 29 777 31
prtesti 1000 29 777 31, l(99)
```

[nonparametric]

[see ‘count’ option]

*ANOVA:*

```
oneway varname catvar, tab scheffe
kwallist varname, by(catvar)
```

[nonparametric]

```
anova varname catv1 catv2 catv1*catv2
anova varname catv1 catv2
```