**EPIDEMIOLOGY 340.600: STATA PROGRAMMING**

**Lab 4**

This lab is optional; you are NOT required to complete these questions. Please use this lab as an opportunity to review the course material and prepare yourself for the homework questions. Sample responses to the lab questions will be provided separately.

1. In lectures 3 & 4, we discussed how you can define your own “program”. It’s an awesome tool that allows us to automate a specific task. If you think a specific part of your code will be used multiple times, you might as well put that into a program. In this lab, we will practice customizing our programs.
2. Start Stata, open your do-file editor, write the header, and load transplants.dta.
3. Write a program called mymean. This program will take varlist as a user input, and calculate the mean value of each variable, and display the values.

capture program drop mymean

program define mymean

syntax varlist

foreach var in `varlist' {

quietly sum `var'

display r(mean)

}

end

1. Modify your program mymean so that when an if argument is supplied, mymean would only include the observations that meet the condition specified by the if argument. In other words, if the user types mymean height if age>65, the program mymean will calculate the mean only among patients older than 65.

capture program drop mymean

program define mymean

syntax varlist [if]

foreach var in `varlist' {

quietly sum `var' `if'

display r(mean)

}

end

1. Further modify your program mymean to include the option sd. When the option sd is supplied, mymean will display the standard deviation along with the mean. This version of mymean should still be able to accommodate the if argument.

capture program drop mymean

program define mymean

syntax varlist [if], [sd]

foreach var in `varlist' {

quietly sum `var' `if'

display r(mean)

if "`sd'" != "" {

display r(sd)

}

}

end

// The answer above is in the simplest possible form for clarity. In practice, I will arrange the outputs a little bit better.

capture program drop mymean

program define mymean

syntax varlist [if], [sd]

foreach var in `varlist' {

quietly sum `var' `if'

if "`sd'" != "" {

display "`var': " r(mean) " (" r(sd) ")"

}

else {

display "`var': " r(mean)

}

}

end

1. Further modify your program mymean to include the option digits(), with a number in the parenthesis. When the option digits() is supplied, mymean will round up the mean (and the standard deviation, if applicable) in units of digits(). If digits() is NOT supplied, round in units of 0.001. (Hint: use the Stata function round())

capture program drop mymean

program define mymean

syntax varlist [if], [sd] [digits(real 0.001)]

foreach var in `varlist' {

quietly sum `var' `if'

display round(r(mean), `digits')

if "`sd'" != "" {

display round(r(sd), `digits')

}

}

end

1. Did you make if, sd, and digits() optional arguments? That is, your program should run whether or not these arguments are supplied. To do so, simply surround each argument with brackets. For example, [sd]
2. I’d like to draw your attention to the merge command. It’s hard to write a question around merge, but it’s a really important command in practice.

merge 1:1 fake\_id using donors\_recipients

This is the code from the lecture. We are merging transplants.dta with donors\_recipients.dta. We are merging observations with the same fake\_id, and expect that there will be only one observation per fake\_id in both datasets.

1. We want to study if death (died==1) is associated with several predictor variables: bmi, prev\_ki, age, peak\_pra, or gender. Run logistic regression between died and each of the predictor variables using foreach loop. At each run, save the name and the regression coefficient of the predictor variable into an external Stata dataset file named output.dta.

postfile output str30 name coef using output

// you may add ", replace" to allow overwriting output.dta

foreach var in bmi prev\_ki age peak\_pra gender {

quietly logistic died `var'

post output ("`var'") (\_b[`var'])

}

postclose output

1. You have all your commands in your do file, right? Run your do file from the beginning and make sure your do file does exactly the same thing.