**EPIDEMIOLOGY 340.600: STATA PROGRAMMING AND DATA MANAGEMENT**

**Assignment 2**

**Due date: 11:59 p.m., Wednesday, May 5, 2021 via CoursePlus dropbox**

**Direction**

Write a .do file that performs the tasks described below. Your .do file must be called assignment2\_*yourname*.do (e.g.: assignment2\_allanmassie.do). Your .do file should follow conventions for .do file structure described in class. Make sure your script will run on our machines, even if we are using a different version of Stata. Do not submit your log files as part of the assignment.

**Evaluation**

For Question 1, use the dataset hw2\_pra\_hist.dta and hw2\_hosp.dta (download from CoursePlus) to perform the required tasks. Your .do file will be run on a different dataset with more visits.

For the other questions, simply define your program. You do not need to run the programs you write in your .do file. The graders will run your programs using a dataset that will not be released to you.

A successful answer would look like this:

|  |
| --- |
| [insert your header code]  \*\*\* Question 1  [insert your code]  \*\*\* Question 2  [insert your code, if necessary]  program define unilogit  [insert your code]  end  [insert your Question 3 response]  [insert your closer code] |

Another script called assignment2\_test.do can be downloaded from CoursePlus. We will use a script that looks like this to grade your responses. Use this script to test your code yourself.

Partial credit will be awarded if the output is wrong, so have your script do *something* for every question. Make sure the output includes the question number as indicated. Make sure to follow the coding guidelines from class; for example, your script should include comments.

**Question 1**

**Context**: You are conducting a study that examines the regional variation in the distribution of panel-reactive antibody (PRA). You recruited 73 patients (px\_id = 1, …., 73) from 10 hospitals (hosp\_id=1, …,10) in 3 regions (region=A, B, C, … ), and measured PRA 3 times: visit 1, visit 2, and visit 3. You hear that the organization that funds your research plans to extend the funding for several more visits (visit 4, visit 5, …, visit N). Since you do not know how many more visits there will be, you decide to write a .do file that can work regardless of how many visits the dataset has.

**Codebook**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Description** | **Values/Range** |
| **hw2\_pra\_hist.dta** | | |
| hosp\_id | Hospital ID | Integers: 1 – 10 |
| px\_id | Patient ID | Integers: 1 – 94 |
| visit\_id | Visit ID | Integers: 1 – N  1 indicates the first visit. K indicates the K-th visit. |
| pra | PRA value at the visit | Integers: 0 – 100 |
| **hw2\_hosp.dta** | | |
| hosp\_id | Hospital ID | Integers: 1 – 10 |
| Region | Region | Alphabets |

1. Load hw2\_pra\_hist.dta. Print a table as shown below, which displays the number of patients with a non-missing PRA value at each visit. N and XX should be replaced with the correct values from the dataset. (Hint: how do you write a forvalue loop for all values of visit\_id?)

Question 1.i)

Visit Count

1 XX

2 XX

⋮

[omitted, but your .do file should display all variables]

⋮

N XX

1. Create a new variable peak\_pra, which contains the maximum value of PRA within each participant. Print the median (IQR) of peak\_pra across the patients as shown below. XX.X should be replaced with the correct values from the dataset and formatted with one digit after the decimal point (e.g., 12.0). (Hint: each patient must be counted only once when calculating the median and IQR.)

Question 1.ii): The median (IQR) of peak\_pra is xx.x (xx.x-xx.x).

1. Another dataset provided to you, hw2\_hosp.dta, has information on which region each hospital is located in. Merge the current dataset in memory with hw2\_hosp.dta without altering the number of observations in memory. Use the command list to list the ID of the patient with the highest peak\_pra value for each region as shown below.

XX should be replaced with the correct values from the dataset. If there are ties (i.e., multiple patients with the highest value), print all tied patients. If region C has ties (while A and B does not), the table will look like below.

+---------------------------+

| region px\_id peak\_pra |

|---------------------------|

| A XX XX |

|---------------------------|

| B XX XX |

|---------------------------|

| C XX XX |

| C XX XX |

+---------------------------+

Hint: your list command should look like this

list region px\_id peak\_pra [insert your code here], sepby(region) noobs

**Question 2**

Define a program called unilogit. This program runs a series of univariable (simple) logistic regressions between each of the predictors supplied by the user and a binary outcome variable also supplied by the user.

For example, if the user runs

unilogit var1 var2 var3 var4, outcome(var5)

this program will quietly run four univariable logistic regressions on var5,

logistic var5 var1

logistic var5 var2

logistic var5 var3

logistic var5 var4

and return the following output, assuming that var2 and var4 were significantly (p<0.05) associated with var5.

Significantly associated with var5:

var2 (p=x.xxx)

var4 (p=x.xxx)

P-values should be rounded to the nearest thousandth (i.e., three digits after the decimal point).

This program should also support if subsetting. For example, if the user runs

unilogit var1 var2 var3 if var4==1, outcome(var5)

this program should run only on the observations with var4==1.

This program should not alter the dataset in the memory: i.e., if you need to alter the dataset, preserve before making changes and restore to the original status after completing your procedures.

Hint: The program model in lecture 4 has some similarities with this question. The p-value after logistic can be obtained using the following code:

(1-normal(abs(\_b[var1]/\_se[var1])))\*2

**Question 3**

Print the following text: "Question 3: I estimate that it took me xxxx hours to complete this assignment."

For example, if it took you six hours of active work, your .do file will contain the line. Give an honest answer; this is just for our data collection purposes. However, this question is worth some points, so don't skip it!

**EXTRA CREDIT CHALLENGE**

Note: You can earn full score on Assignment 2 WITHOUT answering the following question.

**Question 4**

A [prime number](https://en.wikipedia.org/wiki/Prime_number) is a natural number greater than 1 that cannot be formed by multiplying two smaller natural numbers. Write prime, a program that takes any real number as an option n and determines whether the number is a prime number or not. The program will also display an error message when the user enters any number that is not a natural number greater than 1.

For example:

If the user types prime, n(100), your program will display

“100 is NOT a prime number.”

If the user types prime, n(109), your program will display

“109 is a prime number.”

If the user types prime, n(1), your program will display

“Invalid input: enter a natural number greater than 1.”

If the user types prime, n(3.14), your program will display

“Invalid input: enter a natural number greater than 1.”

\*\*\*extra special bonus points for the fastest prime program\*\*\*