Problem Set 1:Getting Started Key

Claire Duquennois

6/5/2020

Erica Schultz - Answer Key

1. Preliminary: Set up your GitHub and Git.

Git and GitHub are often used in conjunction but are really two distinct things. Git is basically a version control software that is based on your local computer. GitHub is a webservice where you have an account that allows you to do version control via the website, which becomes particularly interesting when you are working on collaborative projects and want to avoid the issue of how to synchronize and manage changes to a document when multiple people are working on the same document simultaneously. In theory you could use only one or the other but many people use them in conjunction.

To get a big picture overview of exactly what you can do with Git and Git hub, I would recommend taking the time to watch the Git and GitHub for Poets series of youtube videos https://www.youtube.com/playlist?list=PLRqwX-V7Uu6ZF9C0YMKuns9sLDzK6zoiV.

I am assuming you already have a GitHub account, if not make sure you start one.

Getting Started with Git

Git is often used in conjunction with GitHub as a way to work on GitHub projects locally offline and then reintegrate your work back into a GitHub project. While there is a visual interface you can use to run Git, most programmers seem to prefer to run it directly by typing in Git commands through a terminal prompt.

On MACs: You can type Git commands directly into your Terminal (you can find it in Application/Utilities).

On Windows: You will want to install Git Bash. Go to git-scm.com and download the appropriate .exe file for your operating system. Run the .exe file and install Git. In the installation, the only default you may want to change is to select the option to "Use Git and optional Unix tools from the Windows Command Prompt". Once installed, launch Git Bash and you will see a terminal window in which you can execute Git commands. See the following link for a video of the installation and some examples of commands https://www.youtube.com/watch?v=J_Clau1bYco.

You can use the terminal window to navigate through the folders in your computer's directory and then, using the proper commands, you will be able to use this terminal to push and pull materials from your computer onto github and vice versa. This will allow you to work on a github project on your local machine without having to be constantly online.

Project Setup

On GitHub, generate a project repository for your group for this assignment by cloning the MQE_PS1 repository from my GitHub account and naming it `PS1_name1_name2_name3` where the usernames are the first names of your groupmembers. This repository includes this R-Markdown file you will use to complete the assignment.

So that you can work on your individual machines without having to be online, each group member should:

1) Make an empty folder named, for example, PS1_GS where you will put the files you are going to pull off GitHub (preferably in Dropbox or someplace where it will be backed up).
2) Open your command terminal (or Git Bash on Windows).
3) Navigate via your command terminal to the new folder by typing

```
$ cd /c/Users/ *******/Dropbox/PS1_GS
```

followed by enter.

Note: the directory path will be different for you! To easily get your directory path, you can simply drag and drop the folder into your terminal window.

4) Check that you are in the right folder by typing

```
$ pwd
```

and you should see the directory path to the folder you want to be using.

5) One group member who will 'host' the assignment should clone the MQE_PS1 folder from my github account into their github site and rename it as PS1_name1_name2_name3. The host and the other group members can then clone you homework project file onto your local machines as follows: In the terminal type `git clone` and copy the url it gives you

```
$ git clone github.com/hostuserid/PS1_name1_name2_name3.git
```

Notice, the url is just the web address with .git added to the end.

You should now find that the folder you created has a copy of the group's R-Markdown file.

Open the file in RStudio and click on the `Knit` button. You should see the incomplete assignment displayed in an html window. You can now make edits to this document.

TO DO: Start by adding your name as group member at the top of the document.

Once you have made changes to the assignment file and saved them, and viewed them by clicking `Knit` (which will generate the updated html file), you may want to share these changes with your group mates.

1) Go to the terminal window and navigate to your cloned directory by typing `$ cd /c/Users/*******/Dropbox/MQE_CI/PS1_name1_name2_name3`
2) Check the status of your edits. Type `$git status`

You should see that you have modified files that have not yet been commited.

---

3) **WARNING:**

At this point you want to check if GitHub knows who you are. Type

```
$ git config --list
```

and it will spit out a bunch on information. Look and see if your GitHub username and the email you used to sign on to GitHub with are listed. If they are all is well. If they are not listed type

```
$ git config --global user.name "janedoe"
```

```
$ git config --global user.email janedoe@pitt.edu
```

Make sure you set these to your github user name and the email you used to sign up with GitHub.

---

4) Now that GitHub knows who to attribute edits to, you can commit the modified files. Type

```
$ git commit -a -m "Jane Doe's changes"
```

where the -a argument tells git to commit all the modifications and -m adds a message to you commit (a good practice when working on jointly edited files). Now if you type

```
$ git status
```

you will see that the files you changed are ready to be pushed to github so that they are reflected there.

5) Now you can type

```
$ git push origin master
```

which basically tells git to push your changes from your remote "origin" computer to the "master" branch on github.

6) You can now look at your repository on github and check that the edits you made appear in .Rmd file and the .html file. Because others in your group may have made changes as well, GitHub may prompt you to merge changes and fix any conflicting lines.

7) If you want to view the rendered html online, open the .html file and add https://htmlpreview.github.io/? at the begining of the web address in your browsed bar (example: https://htmlpreview.github.io/?https://github.com/claireduq/MQEPS/blob/master/PS1/PS1_OVB.html.)

Getting the most up to date version of your group assignment

When there are changes on in the GitHub repository that you want reflected on your computer, you will need to "pull" these changes. For instance, once your group mates have added their names, and pushed their changes to GitHub, you can pull these new changes by typing

```
$git pull
```

Now you are set up to work cooperatively on an assignment using GitHub. The remainder of this assignment consists of a simple coding and plotting exercise to practice collaborating with Github.

Omitted Variable Bias Simulation (Group excercise)

For this problem we are going to generate simulated data, with known characteristics, so that we can explore how the omission of a key variable biases our estimates.

To do this in R markdown we must first tell R to interpret the text we write as code. We do this by embedding a code *chunk* in the document. The following is a code *chunk* that only consists of a comment. Notice the chunk starts and ends with three ‘ marks and is given in a name in brackets.

```
#This is a comment, alone in a chunk.
```

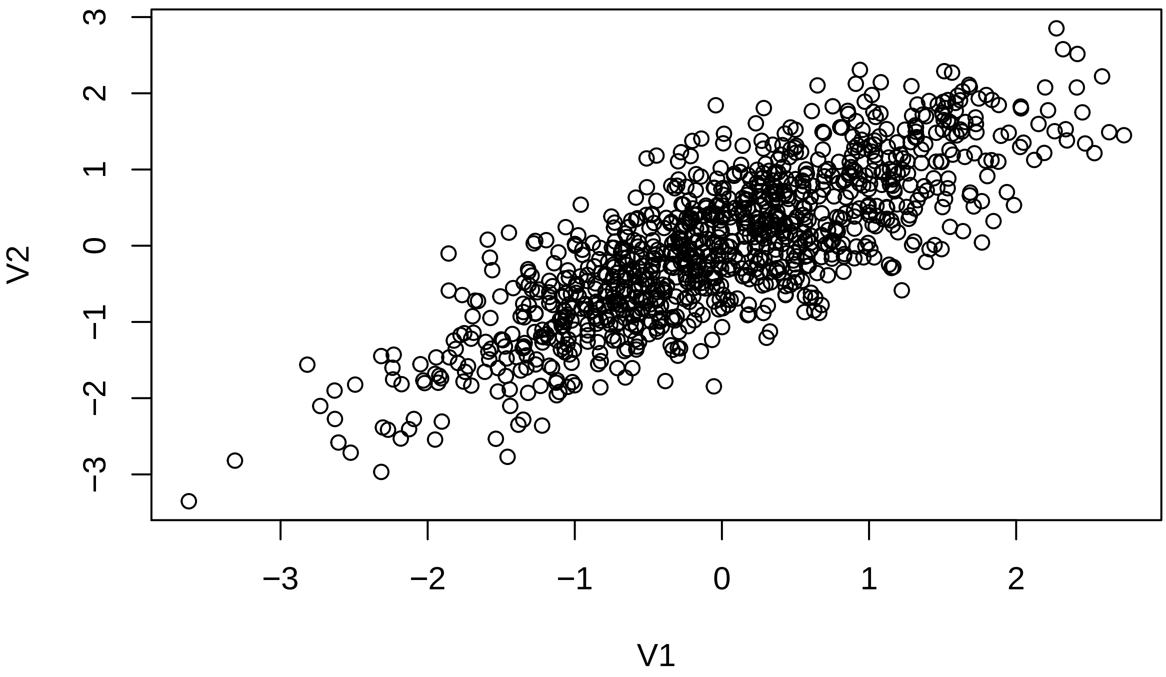
In the following chunk I generate simulated data consisting of two correlated variables

```
library(MASS)
library(ggplot2)

out <- as.data.frame(mvrnorm(1000, mu = c(0,0),
                      Sigma = matrix(c(1,0.8,0.8,1), ncol = 2),
                      empirical = TRUE))
cor(out)
```

```
##     V1  V2
## V1 1.0 0.8
## V2 0.8 1.0
```

```
plot(out)
```



Next I generate a randomly distributed error term and I calculate the outcome variable which depends on both V1 and V2 and some noise:

$$Y = \beta_1 V_1 + \beta_2 V_2 + \epsilon$$

```
out$error<-rnorm(1000, mean=0, sd=1)

#The data generating process
B1<-3
B2<-6

out$Y<-out$V1*B1+out$V2*B2+out$error
```

TO DO: For the questions below write the needed code and a written response to the question.

Question 1: Write a chunk in which you regress Y on V_1 and V_2. Are your estimates of β_1 and β_2 biased?

```
lm(Y~V1+V2, data = out)
```

```
##
## Call:
## lm(formula = Y ~ V1 + V2, data = out)
##
## Coefficients:
## (Intercept)           V1           V2
##     0.01009      2.86683      6.13132
```

The coefficients are not biased because our coefficients are centered around their true values.

Question 2: Write a chunk in which you regress Y on V_1 only. Is your estimate of β_1 biased?

```
lm(Y~V1, data = out)
```

```
##
## Call:
## lm(formula = Y ~ V1, data = out)
##
## Coefficients:
## (Intercept)           V1
##     0.01009      7.77188
```

This is biased because there is an omitted variable in this regression.

Question 3: Generate a new variable Y_{adj} such that $Y_{adj} = Y - \beta_2 * V_2$. Then regress Y_{adj} on V_1. Is your estimate of β_1 biased? Can you explain why/why not?

```
out$Y_adj <- out$Y - B2*out$V2

lm(Y_adj~V1, data = out)
```

```
##
## Call:
## lm(formula = Y_adj ~ V1, data = out)
##
## Coefficients:
## (Intercept)           V1
##     0.01009      2.97188
```

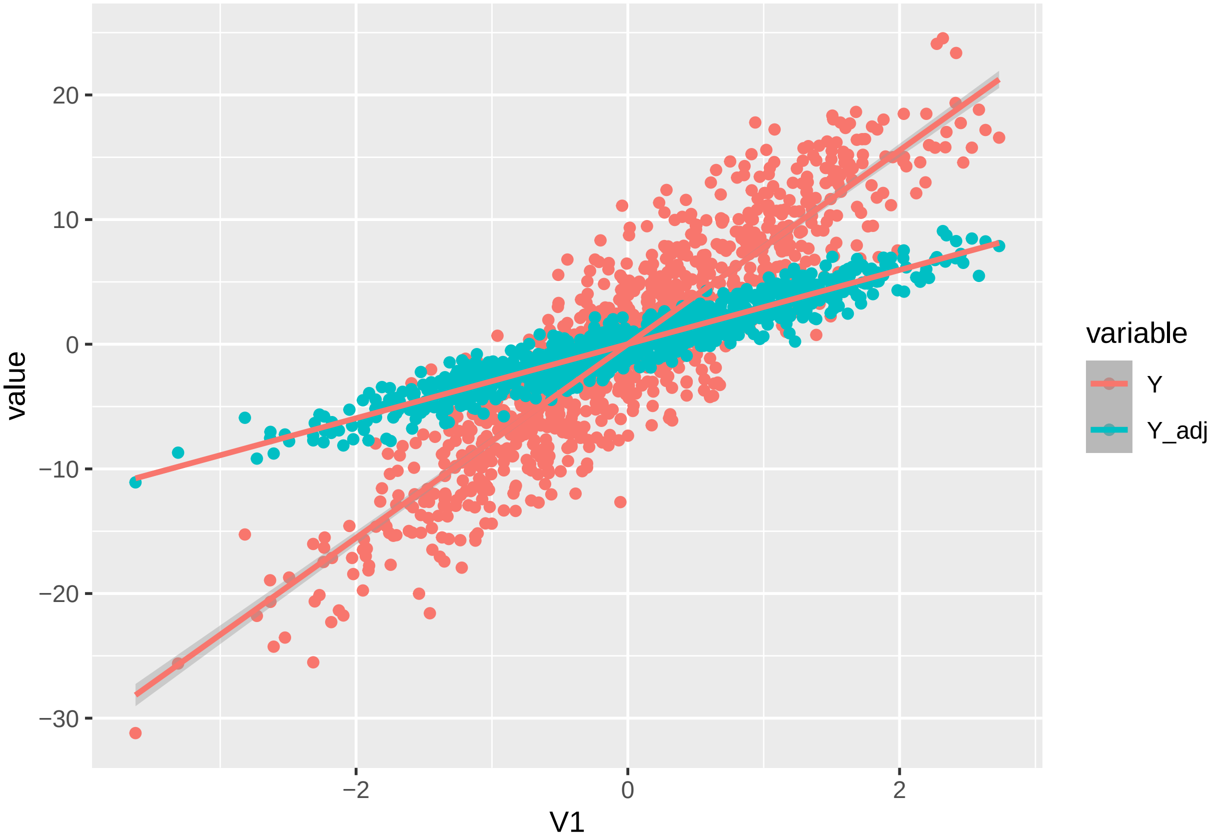
Now it is not biased because our outcome variable has taken away the effect of V2.

Question 4: The code below generates a scatter plot and regression line for the relationship between V_1 and Y as well as V_1 and Y_{adj}. Submit an improved visualization of this data. Hint: you will need to delete the # to get the code to run

```
plotted<-ggplot(out, aes(V1, y = value, color = variable)) +
    geom_point(aes(y = Y, col = "Y")) +
    geom_point(aes(y = Y_adj, col = "Y_adj"))+
    geom_smooth(method='lm', aes(y = Y, col = "Y"))+
    geom_smooth(method='lm', aes(y = Y_adj, col = "Y"))

plotted
```

```
## 'geom_smooth()' using formula 'y ~ x'
## 'geom_smooth()' using formula 'y ~ x'
```



Submission instructions:

1) Make sure the final version of your assignment is uploaded on GitHub in both html and Rmarkdown format.

2) Knit your final version as a Word or Pdf document and submit this to Gradescope by the due date.