Homework 1

Yunting

1/20/2020

Instructions

There are two parts to this homework:
1. Reproduce this **entire document** in Markdown (including these instructions).
2. Insert code chunks as apporpriate and solve the problems in the second part.

Knit directly to PDF.

Submit both .Rmd and PDF to Blackboard in a single submission by the due date and time (before the next class).

STAT 412/612 Statistical Programming in R, Spring 2020 Secs 007/007 (Syllabus Extract)

Time: 5:30 - 8:00 pm ~~Fridays Don Myers Technology and Innovation (DMTI), Room 109.
Instructor: Dr. Richard Ressler
* Email: rressler@american.edu
* Office: DMTI 106-O
* Phone: 202-885-6472
* Office Hours: in Syllabus on Blackboard

Material

- Required Hardware and Software:
  - *Laptop computer* - **fully charged for each class**
  - Current version of R
  - Current version of R Studio

- Books: R for Data Science by Wickham and Grolemund (O'Reilly).

Acknowledgements

I have read the syllabus provided on the Blackboard system for this class and section.

I understand the course learning outcomes and tentative schedule.

If I have to miss class *on the rare occasion*, I am responsible for any assignments or papers given out during any missed class. I will obtain these materials from a colleague BEFORE the next class meeting.

I expect to have to do some research (using Google, stackoverflow, etc.) to do my assignments. I will cite sources from which I have used code and describe how I adjusted it.

I understand sharing, reviewing, or using solutions to any exam or homework in any way from previous or concurrent versions of this course is prohibited and a violation of the *Academic Integrity Code.*

Graded Work

Assignments: There will be approximately 9 formal homework assignments throughout the semester plus deliverables for a final project.

I may receive assistance from other students in the class and the professor, but my submissions must be composed of my own thoughts, coding and words. If I get ideas from online resources such as stackoverflow or github when I get stuck, I will cite my source and be specific about what I have added to it. I will be able to redo the code "cold" when I do this. Failure to do so is a violation of AU's Academic Integrity Code.

Late assignments will not be accepted.

Exams: We will have approximately three in-class exams. Any material covered in class, assigned readings,or on assignments is "fair game." No make-up exams will be given unless I have an extremely compelling excuse such as a previously-requested observance of a religious holiday or a documented medical emergency.

Project:

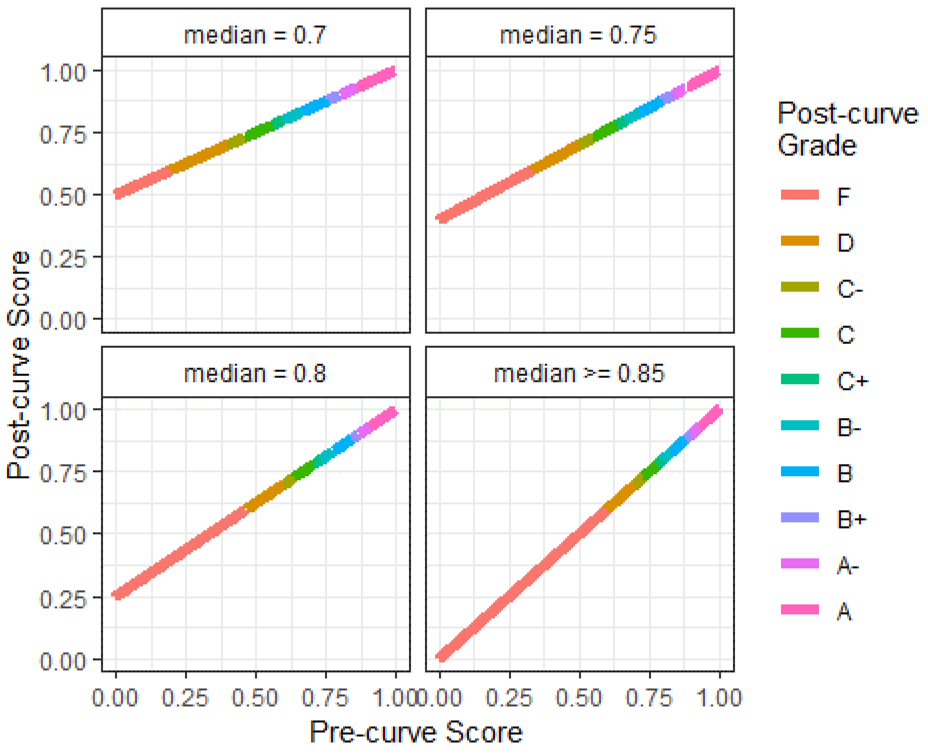
> My project should involve working with 2-4 classmates on a fairly large real-world dataset to answer some question of interest. It should be reproducible and include graphical representations of my data.

Grading I should be able to explain my work on assignments, exams, and project and my rationale. Based on my explanation (or lack thereof), the professor may modify my grade. My final grade will be determined by:

Undergraduate students-412:	**Graduate Students-612:**
Assignments (40%) Exams (30% composed of: Exam 1 = 10%,Exam 2 = 10%, Final Exam = 10%) Final Project (20%) Attendance and Participation (10%)	Assignments (40%) Exams (30% composed of: Exam 1 = 10%, Exam 2 = 10%, Final Exam = 10%) Final Project (30%) Attendance and Participation (May lower Grade%)

Final Grades

The final grades will be based on a curve if the median is below 85. A visual representation of possible curves follows:



Other Notes

We will occasionally need to type equations like $A = \pi * r^2$. More often we will evaluate something like: the number of cars in the mtcars built-in dataset is 50.

Exercises using Base R

Complete the following exercises using base R.
Useful functions seq(), sum(), mean(), sd,[], c(), length(), log(), data.frame()

1. Create a vector that contains all integers divisible by 5 from 65 to 250. Assign this vector to a variable. Add up the elements of this vector. What is the mean of the vector?

```
x <- c(65:250)
y <- x [x%%5 ==0]
sum(y)
```

```
## [1] 5985
```

```
mean(y)
```

```
## [1] 157.5
```

2. Create a vector of numerics of length 100 that starts at 65 and ends at 250 and assign to a variable. The difference between any two consecutive elements should be the same. Add up the elements of this vector. What is its standard deviation?

```r
x <- seq(from <- 65,to <- 250,length.out = 100)
x
```

```
##   [1]  65.00000  66.86869  68.73737  70.60606  72.47475  74.34343  76.21212
##   [8]  78.08081  79.94949  81.81818  83.68687  85.55556  87.42424  89.29293
##  [15]  91.16162  93.03030  94.89899  96.76768  98.63636 100.50505 102.37374
##  [22] 104.24242 106.11111 107.97980 109.84848 111.71717 113.58586 115.45455
##  [29] 117.32323 119.19192 121.06061 122.92929 124.79798 126.66667 128.53535
##  [36] 130.40404 132.27273 134.14141 136.01010 137.87879 139.74747 141.61616
##  [43] 143.48485 145.35354 147.22222 149.09091 150.95960 152.82828 154.69697
##  [50] 156.56566 158.43434 160.30303 162.17172 164.04040 165.90909 167.77778
##  [57] 169.64646 171.51515 173.38384 175.25253 177.12121 178.98990 180.85859
##  [64] 182.72727 184.59596 186.46465 188.33333 190.20202 192.07071 193.93939
##  [71] 195.80808 197.67677 199.54545 201.41414 203.28283 205.15152 207.02020
##  [78] 208.88889 210.75758 212.62626 214.49495 216.36364 218.23232 220.10101
##  [85] 221.96970 223.83838 225.70707 227.57576 229.44444 231.31313 233.18182
##  [92] 235.05051 236.91919 238.78788 240.65657 242.52525 244.39394 246.26263
##  [99] 248.13131 250.00000
```

```r
sum(x)
```

```
## [1] 15750
```

```r
sd(x,na.rm = TRUE)
```

```
## [1] 54.21339
```

3. Extract the 11th element from the vector you created in part 1.

```r
y[11]
```

```
## [1] 115
```

4. Extract the 11th to 25th elements from the vector you created in part 1.

```r
y[11:25]
```

```
##  [1] 115 120 125 130 135 140 145 150 155 160 165 170 175 180 185
```

5. Combine the vectors from parts 1 and 2 and assign this combined vector to a new variable.

```r
a <- seq(from <- 65,to <- 250,length.out = 100)
total <- c(y,a)
total
```

```
##   [1]  65.00000  70.00000  75.00000  80.00000  85.00000  90.00000  95.00000
##   [8] 100.00000 105.00000 110.00000 115.00000 120.00000 125.00000 130.00000
##  [15] 135.00000 140.00000 145.00000 150.00000 155.00000 160.00000 165.00000
##  [22] 170.00000 175.00000 180.00000 185.00000 190.00000 195.00000 200.00000
##  [29] 205.00000 210.00000 215.00000 220.00000 225.00000 230.00000 235.00000
##  [36] 240.00000 245.00000 250.00000  65.00000  66.86869  68.73737  70.60606
##  [43]  72.47475  74.34343  76.21212  78.08081  79.94949  81.81818  83.68687
##  [50]  85.55556  87.42424  89.29293  91.16162  93.03030  94.89899  96.76768
##  [57]  98.63636 100.50505 102.37374 104.24242 106.11111 107.97980 109.84848
##  [64] 111.71717 113.58586 115.45455 117.32323 119.19192 121.06061 122.92929
##  [71] 124.79798 126.66667 128.53535 130.40404 132.27273 134.14141 136.01010
##  [78] 137.87879 139.74747 141.61616 143.48485 145.35354 147.22222 149.09091
##  [85] 150.95960 152.82828 154.69697 156.56566 158.43434 160.30303 162.17172
##  [92] 164.04040 165.90909 167.77778 169.64646 171.51515 173.38384 175.25253
##  [99] 177.12121 178.98990 180.85859 182.72727 184.59596 186.46465 188.33333
## [106] 190.20202 192.07071 193.93939 195.80808 197.67677 199.54545 201.41414
## [113] 203.28283 205.15152 207.02020 208.88889 210.75758 212.62626 214.49495
## [120] 216.36364 218.23232 220.10101 221.96970 223.83838 225.70707 227.57576
## [127] 229.44444 231.31313 233.18182 235.05051 236.91919 238.78788 240.65657
## [134] 242.52525 244.39394 246.26263 248.13131 250.00000
```

6. Use a function to determine the length of the vector in part 5.

```
length_of_total <- length(total)
length_of_total
```

```
## [1] 138
```

7. What is the sum of the log of every element in the vector in part 5?

```
num <- c(y,a)
log(num)
```

```
##   [1] 4.174387 4.248495 4.317488 4.382027 4.442651 4.499810 4.553877 4.605170
##   [9] 4.653960 4.700480 4.744932 4.787492 4.828314 4.867534 4.905275 4.941642
##  [17] 4.976734 5.010635 5.043425 5.075174 5.105945 5.135798 5.164786 5.192957
##  [25] 5.220356 5.247024 5.273000 5.298317 5.323010 5.347108 5.370638 5.393628
##  [33] 5.416100 5.438079 5.459586 5.480639 5.501258 5.521461 4.174387 4.202731
##  [41] 4.230293 4.257116 4.283238 4.308695 4.333521 4.357744 4.381395 4.404499
##  [49] 4.427082 4.449166 4.470773 4.491922 4.512634 4.532925 4.552813 4.572313
##  [57] 4.591440 4.610208 4.628630 4.646719 4.664487 4.681944 4.699102 4.715970
##  [65] 4.732559 4.748877 4.764933 4.780735 4.796291 4.811609 4.826696 4.841559
##  [73] 4.856204 4.870638 4.884866 4.898895 4.912729 4.926375 4.939837 4.953120
##  [81] 4.966229 4.979169 4.991943 5.004556 5.017012 5.029315 5.041468 5.053475
##  [89] 5.065340 5.077066 5.088656 5.100113 5.111440 5.122640 5.133717 5.144672
##  [97] 5.155508 5.166228 5.176834 5.187329 5.197715 5.207995 5.218169 5.228242
## [105] 5.238213 5.248087 5.257864 5.267546 5.277135 5.286633 5.296042 5.305363
## [113] 5.314598 5.323749 5.332816 5.341802 5.350709 5.359536 5.368286 5.376960
## [121] 5.385560 5.394087 5.402541 5.410924 5.419238 5.427483 5.435661 5.443772
## [129] 5.451818 5.459800 5.467719 5.475576 5.483371 5.491106 5.498781 5.506399
## [137] 5.513958 5.521461
```

```
sum(log(num))
```

```
## [1] 688.9763
```

8. Create two vectors of length 3, one with numbers and one with characters. Create a dataframe with the vectors. Sum the numbers in the first column.

```
apple <- 1:3
bird <- c("cat", "dog","egg")
df <- data.frame(apple,bird)
df
```

```
##   apple bird
## 1     1  cat
## 2     2  dog
## 3     3  egg
```

```
sum(df$apple)
```

```
## [1] 6
```