Homework 3 STAT 5014

Shane Bookhultz

September 20, 2017

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: sm

## Package 'sm', version 2.2-5.4: type help(sm) for summary information

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

##
## Please cite as:

##  Hlavac, Marek (2015). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2. http://CRAN.R-project.org/package=stargazer
```

Problem 4

My takeaway from the programming style guides is that to make things generally readable, and explain whatever you are doing explicitly if it is not obvious. My coding style implements most of these ideas anyway, but I'm going to try to comment more, utilize spaces more, and name functions/variables better.

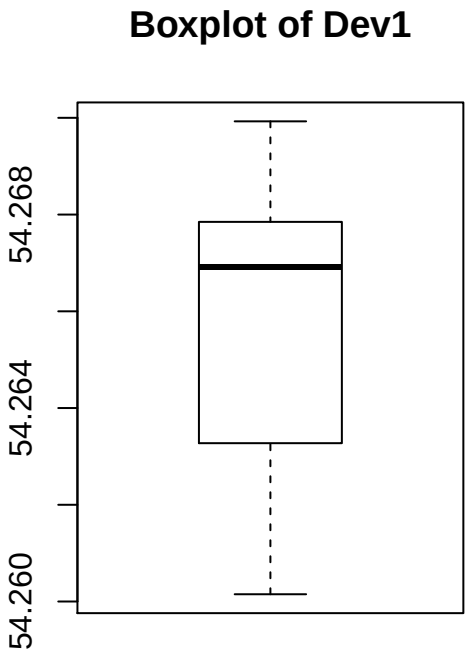
Problem 5

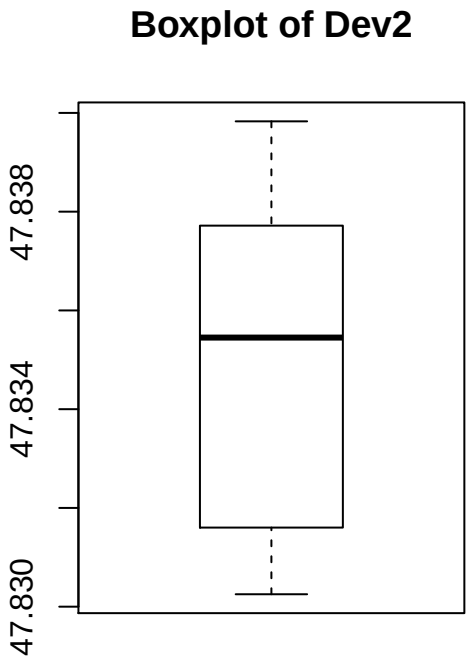
From the messages in my HW2 code, lintr noticed around 80 errors. Most of them were spacing (with commas and parentheses), words in a variable/function should be seperated with an underscore instead of a dot, removing commented code, and only using double quotes. Honestly, I don't see why it matters using double or single quotes since they both indicate a string, and I'm not sure I should remove commented code if I may need to use it in the future.

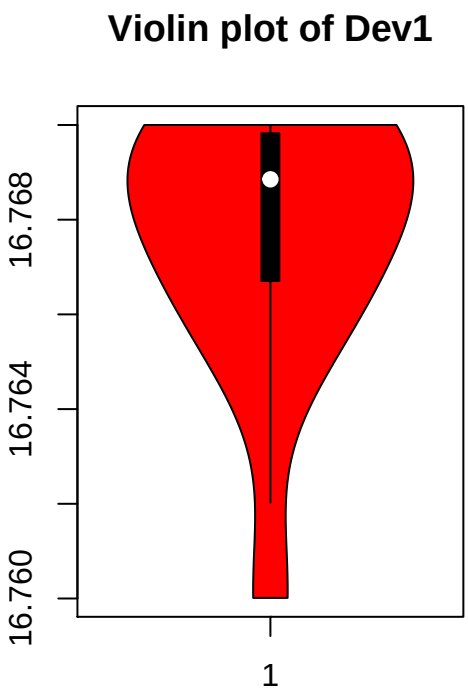
Problem 6

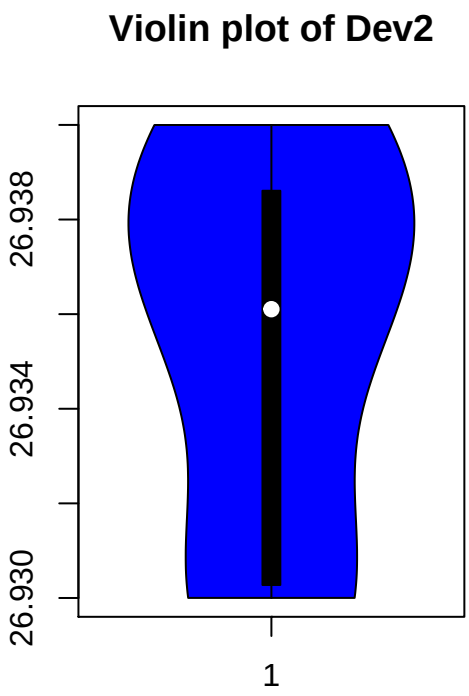
Observer	Mean of Dev1	Mean of Dev2	Sd of Dev1	Sd of Dev2	Correlation
1	54.26610	47.83472	16.76983	26.93974	-0.0641284
2	54.26873	47.83082	16.76924	26.93573	-0.0685864

Observer	Mean of Dev1	Mean of Dev2	Sd of Dev1	Sd of Dev2	Correlation
3	54.26732	47.83772	16.76001	26.93004	-0.0683434
4	54.26327	47.83225	16.76514	26.93540	-0.0644719
5	54.26030	47.83983	16.76774	26.93019	-0.0603414
6	54.26144	47.83025	16.76590	26.93988	-0.0617148
7	54.26881	47.83545	16.76670	26.94000	-0.0685042
8	54.26785	47.83590	16.76676	26.93610	-0.0689797
9	54.26588	47.83150	16.76885	26.93861	-0.0686092
10	54.26734	47.83955	16.76896	26.93027	-0.0629611
11	54.26993	47.83699	16.76996	26.93768	-0.0694456
12	54.26692	47.83160	16.77000	26.93790	-0.0665752
13	54.26015	47.83972	16.76996	26.93000	-0.0655833


Boxplot of Dev1
54.268
54.264
54.260


Boxplot of Dev2
47.838
47.834
47.830


Violin plot of Dev1
16.768
16.764
16.760
1


Violin plot of Dev2
26.938
26.934
26.930
1

Problem 7

In this BloodPressure dataset, I read in the data as a csv file seperated by spaces. From there I split up the data into a Day vector, a BloodPressure vector, and a vector that had values from Dev1 to 3 and Doc1 to 3. Then, I replicated the DevDoc vector so it would be the same length as the BP vector and Day vector (90 length). Then I added them all to a data frame and renamed the columns.

Problem 8

Number of iterations	x(iterations)
1	0.5219224
2	-4.3186393
3	-4.3251931
4	-4.3255781
5	-4.3255823
6	-4.3255823
7	-4.3255823
8	-4.3255823
9	-4.3255823

Problem 9

```
## [1] 0
```

Number	Most Frequent Defects	Occurances
1	Tire(s) present with a profile depth of 1.6 to 2.5 mm	388904
2	Operation/Condition Required Light/ Retroreflector 5*.55	271096
3	Excessive oil leakage	224143
4	Tire insufficient profile	171131
5	Mechanical parts of the braking system show wear	139996

Make	Frequent Defects
VOLKSWAGEN	Tire(s) present with a profile depth of 1.6 to 2.5 mm
PEUGEOT	Operation/Condition Required Light/ Retroreflector 5*.55
OPEL	Excessive oil leakage
VOLKSWAGEN	Tire insufficient profile
VOLKSWAGEN	Mechanical parts of the braking system show wear

```
##
## ================================================
##                             Dependent variable:
##                         ---------------------------
##                                      nn
## ------------------------------------------------
## MakeBUERSTNER                     1,366.000
##
##
## MakeCHEVROLET-SOUTHWIND             2.000
```

```
##
##
## Constant                            1.000
##
##
## ------------------------------------------------
## Observations                          3
## R2                                  1.000
## ================================================
## Note:                  *p<0.1; **p<0.05; ***p<0.01
```

```
## [1] "                   Df  Sum Sq Mean Sq"
## [2] "SmallDefects$Make  2 1242152  621076"
```

```
##
## ================================================================
##                                      Dependent variable:
##                                  ---------------------------
##                                             nn
## ----------------------------------------------------------------
## Model*                                   -2.000
##
##
## Model========; ADRIATIK S 590 DS           5.000
##
##
## Constant                                  3.000
##
##
## ----------------------------------------------------------------
## Observations                                3
## R2                                        1.000
## ================================================================
## Note:                               *p<0.1; **p<0.05; ***p<0.01
```

```
## [1] "                    Df Sum Sq Mean Sq"
## [2] "SmallDefects2$Model  2     26      13"
```

h. Overall for this workflow I made a bunch of datasets and created many allocations in memory dedicated to these datasets. Initally I just read in the data and only selected the cars that were brought in in 2017. From there, I inner_joined the 3 datasets togethe based on license plate and defect code. I called this the FullCarData, and from there I checked for NA's and there were only NA's in the Make/Model column (58/~400000). I converted the column names to english and renamed the column names. Then I created a table of the top 5 occuring defect, translated them to english and put them in a table from 1 to 5. Next, I found the car model that each has the occuring defect most, and put that in a table. In addition, I did the same procedure but with model. Lastly, I created a categorical regression model with the make of the car on the x against number of times the defect came up on the y. I did the same with model. Overall, I could be definitely more computationally efficient. I could create more for loops, stop the intermediate data creation steps, and create more concise names. This problem took up a large amount of data as I read in gigabyte large files, and compiled them together into another created data file, probably taking up at least 3 gigabytes of memory.