HW3_Wei_Yanran

Yanran Wei

September 18, 2017

```
knitr::opts_chunk$set(echo = TRUE)
```

Problem 4

The two links describe good programming style that makes R code easier to read, share and verify. It set up a kind of fundamental standard for R users to follow to create readable R code. In these two links, rules like naming, notation, syntax, functions and organizations are introduced. For me, the way to improve my coding style is to follow above rules and practice. For example, I will name variables in similar format, leave space around all binary operators and write comments if necessary. Then my code will be more readable to other R users.

Problem 5

```
library(lintr)
lint(filename = "./02_data_munging_summarizing_R_git/HW2_Wei_Yanran.Rmd")
```

There are hundreds of suggestions on my code. The most common suggestion include *Put spaces around all infix operators*, *Commas should always have a space after* and *Variable and function names should be all lowercase*. Other suggestions are *Trailing whitespace is superfluous*, *lines should not be more than 80 characters* and *Only use double-quotes*. These are all the bad programming syles I am gonna to improve in my homework.

Problem 6

a

Below is the summary of 13 observers with mean, standard deviation and correlation for each observer.

```
# Load data
library(data.table)
data_q6<-readRDS("C:/Users/Echo/Desktop/2017 Fall/Statistical Package/HW3/HW3_data.rds")
result_q6 <- matrix(nrow = 13,ncol = 6)
colnames(result_q6) <- c("Observers", "mean1", "mean2", "sd1", "sd2","correlation")
for(i in 1:13){
  middle_q6<-subset(data_q6,data_q6[, 1]==i)
  result_q6[i,1] = mean(middle_q6[, 1])
  result_q6[i,2] = mean(middle_q6[, 2])
  result_q6[i,3] = mean(middle_q6[, 3])
  result_q6[i,4] = sd(middle_q6[, 2])
  result_q6[i,5] = sd(middle_q6[, 3])
  result_q6[i,6] = cor(middle_q6[, 2], middle_q6[, 3])
}
```

```
knitr::kable(result_q6, caption="Summary of 13 Observers")
```

b

Below is the boxplot of means of 13 observers. The mean value of dv1 is larger than that of dv2.

```
par(mfcol=c(1,2))
boxplot(result_q6[, 2], data = result_q6, main = "Boxplot of Means from Dev1", xlab = "Dev1", ylab = "Me
boxplot(result_q6[, 3], data = result_q6, main = "Boxplot of Means from Dev2", xlab = "Dev2", ylab = "Me
```

c

Below is the violin plot of sd of 13 observers.

```
library(vioplot)
library(sm)
par(mfcol=c(1,2))
vioplot(result_q6[, 4], names=c("sd1"), col="gold")
title ("Violin Plot of Sd from Dev1")
vioplot(result_q6[, 5], names=c("sd2"), col="gold")
title ("Violin Plot of Sd from Dev2")
```

Problem 7

Submmary table after tidying is shown as below.

```
library(dplyr)
library(tidyr)
url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BloodPressure.dat"
    blood_pressure_raw <- read.table(url, header = F, skip = 1, fill = T, stringsAsFactors = F)
    blood_pressure_tidy <- blood_pressure_raw[-1,-5]
    colnames(blood_pressure_tidy) = c("Day", "Dev_1", "Dev_2", "Dev_3", "Doc_1", "Doc_2",   "Doc_3")
    blood_pressure_tidy <- blood_pressure_tidy %>%
      gather(reading_number, value, Dev_1:Doc_3) %>%
      separate(reading_number, into = c("methods", "replicate"), sep = "_") %>%
      mutate(value = as.numeric(value))
knitr::kable(summary(blood_pressure_tidy), caption="Blood Pressure Data Summary")
```

Problem 8

```
# function expression
f<-function(x){
  return(3^x - sin(x) + cos(5*x))
}
```

From the table above, we can observe that the value of the function is positive after 5 due to the large positive value of 3^x.

```
x <- seq(-50, 5, by = 0.1)
 plot(x,f(x),main = "f(x)", type = "l")
 abline(h = 0, col = "red")
```

So we shrink the dependent variable to the data range of (-50,0). Now the plot of the function is as below. Because the value of 3^x becomes small as value of x decreases, the function is mainly influenced by sin(x) and cos(5x) which lead to periodically fluctuation.

To simplify the question, we only observe one 'period' which correspond to x between the range of (-5,0).

```
x <- seq(-5, 0, by = 0.1)
 plot(x,f(x),main = "f(x)", type = "l")
 abline(h = 0, col = "red")
```

```
# Newton's method with a first order approximation
newton <- function(f, tol = 0.001, x0, N = 500){
  # N = total number of iterations
  # x0 = initial guess
  # tol = abs(x(n+1) - x(n))
  # f = function to be evaluated for a root

  h <- 0.001
  i <- 1
  x1 <- x0
  p <- numeric(N)
  while (i <= N){
    df_dx <- (f(x0 +h) - f(x0)) / h
    x1 <- (x0 - (f(x0) / df_dx))
    p[i] <- x1
    i <- i+ 1
    if (abs(x1 - x0) < tol){
      break
    }
    x0 <- x1
  }
  return(p[1:(i-1)])
}
```

```
newton(f, x0 = -3)
newton(f, x0 = -3.5)
newton(f, x0 = -4)
newton(f, x0 = -5)
```

In the range of x between -5 and 0, ther are four solutions, -2.8870, -3.5287, -3.9301 and -4.9715.

Problem 9

a. Load databasets

Load table using function fread.

```
library(data.table)
#this had the defect code and description
Car_Gebreken_select <- fread(input = "C:/Users/Echo/Downloads/Open_Data_RDW__Gebreken.csv", header = T,
colnames(Car_Gebreken_select) <- c("defect", "description")

#this has the license plate, inspection date and defect code
Car_Geconstat_select <- fread(input = "C:/Users/Echo/Desktop/Open_Data_RDW__Geconstateerde_Gebreken.txt
colnames(Car_Geconstat_select) <- c("license_plate", "inspection_date", "defect")
```

```
#this has the license plate, make and model of vehicle
Car_Person_select  <- fread(input = "C:/Users/Echo/Desktop/Personenauto_basisdata.txt", verbose = TRUE,
colnames(Car_Person_select) <- c("license_plate", "make", "model")
```

b. Merge three tables

Merge tables using merge function.

```
merge2<-merge(Car_Geconstat_select,Car_Person_select,by="license_plate",all=TRUE)
merge3<-merge(Car_Gebreken_select,merge2,by="defect",all=TRUE)
```

c. Clean the data and remove NA

```
# Remove NA
mergec<-merge3[apply(merge3,1,function(x)!any(is.na(x))),,drop=F]
```

d. How many DIFFERENT makes and models of cars

```
merged <- mergec[grep("2017",mergec$inspection_date),]
makes <- n_distinct(merged$make)
models <- n_distinct(merged$model)
makes
models
```

e. 5 most frequent defects and make/models

Select columns we need utilizaing sqldf funcion.

```
library(sqldf)
freq5<-sqldf("select description,make, count (*) as count
              from merged
             Group by description
             ORDER BY count DESC
             LIMIT 5",row.names=TRUE)
knitr::kable(freq5, caption="5 most frequetn defects")
```

f. Relationship between number of defects and make

```
qf <- sqldf("select make, count (*) as count
            from merged
            group by description
            ORDER BY count DESC",row.names=TRUE)
lm1 <- lm(count ~ make, data = qf)
knitr::kable(summary(lm1)$coefficients)
aov1 <- aov(formula = count ~ make, data = qf)
summary(aov1)
```

g. Relationship between number of defects and model

```r
qg <- sqldf("select model, count (*) as count
            from merged group
            by description
            ORDER BY count DESC",row.names=TRUE)
lm2 <- lm(count~model, data = qg)
knitr::kable(summary(lm2)$coefficients)
aov2 <- aov(formula = count ~ model, data = qg)
summary(aov2)
```