HW4_Lin_Zhengzhi

zhengzhi lin

2019.9.22

problem 4

The first one way is to code regularly, by creating a lot variables. This way of coding is easy at when doing the project, but it is hard to review. The second way is using pipe to avoid creating variables, this makes code clean and tidy, easy to read and review.

problem 5

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.1

##
## 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
```

```r
url <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
operator <- read.table(url, fill = TRUE)
op_dat <- as.matrix(operator[-c(1:2), ])
for (i in 1:10) {
  t <- op_dat[3 * i - 1, 1:5]
  t <- c(i, t)
  op_dat[3 * i - 1, ] <- t
  m <- op_dat[3 * i, 1 : 5]
  m <- c(i, m)
  op_dat[3 * i, ] <- m
}
op_dat <- op_dat %>% as.data.frame() %>% rename(item = V1,
         operator1 = V2, operator2 = V3, operator3 = V4, operator4 = V5,
         operator5 = V6) %>%
  mutate_if(is.factor, as.character) %>% mutate_if(is.character, as.numeric )
head(op_dat)
```

```
##   item operator1 operator2 operator3 operator4 operator5
## 1    1       4.3       4.9       3.3       5.3       4.4
## 2    1       4.3       4.5       4.0       5.5       3.3
```

```
## 3    1       4.1       5.3       3.4       5.7       4.7
## 4    2       6.0       5.3       4.5       5.9       4.7
## 5    2       4.9       6.3       4.2       5.5       4.9
## 6    2       6.0       5.9       4.7       6.3       4.6
```

```r
op_dat %>% str() %>% summary()
```

```
## 'data.frame':    30 obs. of  6 variables:
##  $ item     : num  1 1 1 2 2 2 3 3 3 4 ...
##  $ operator1: num  4.3 4.3 4.1 6 4.9 6 2.4 3.9 1.9 7.4 ...
##  $ operator2: num  4.9 4.5 5.3 5.3 6.3 5.9 2.5 3 3.9 8.2 ...
##  $ operator3: num  3.3 4 3.4 4.5 4.2 4.7 2.3 2.8 2.6 6.4 ...
##  $ operator4: num  5.3 5.5 5.7 5.9 5.5 6.3 3.1 2.7 4.6 6.8 ...
##  $ operator5: num  4.4 3.3 4.7 4.7 4.9 4.6 2.4 1.3 2.2 6 ...
```

```
## Length  Class   Mode
##      0   NULL   NULL
```

```r
url <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
olympic <- read.table(url, fill = TRUE)
olympic <- olympic[- 1, ]
o1 <- olympic[, 1 : 2] %>% rename(year = V1, "Long jump" = V2) %>%
  mutate_if(is.factor, as.character) %>% mutate_if(is.character, as.numeric)
o2 <- olympic[, 3 : 4] %>% rename(year = V3, "Long jump" = V4) %>%
  mutate_if(is.factor, as.character) %>% mutate_if(is.character, as.numeric)
o3 <- olympic[, 5 : 6] %>% rename(year = V5, "Long jump" = V6) %>%
  mutate_if(is.factor, as.character) %>%
  mutate_if(is.character, as.numeric)
o4 <- olympic[, 7 : 8] %>% rename(year = V7, "Long jump" = V8) %>%
  mutate_if(is.factor, as.character) %>%
  mutate_if(is.character, as.numeric)

oly_dat <- o1 %>%
  bind_rows(o2) %>% bind_rows(o3) %>% bind_rows(o4)
head(oly_dat)
```

```
##   year Long jump
## 1   -4    249.75
## 2    0    282.88
## 3    4    289.00
## 4    8    294.50
## 5   12    299.25
## 6   20    281.50
```

```r
oly_dat %>% str() %>% summary()
```

```
## 'data.frame':    24 obs. of  2 variables:
##  $ year     : num  -4 0 4 8 12 20 24 28 32 36 ...
##  $ Long jump: num  250 283 289 294 299 ...
```

```
## Length  Class   Mode
##      0   NULL   NULL
```

```r
url <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
weight <- read.table(url, fill = TRUE)
weight <- weight[- 1, 1 : 6]
w1 <- weight[, 1 : 2] %>%  rename("Body Wt" = V1, "Brain Wt" = V2) %>%
  mutate_if(is.factor, as.character) %>% mutate_if(is.character, as.numeric)
w2 <- weight[, 3 : 4] %>%  rename("Body Wt" = V3, "Brain Wt" = V4) %>%
  mutate_if(is.factor, as.character) %>% mutate_if(is.character, as.numeric)
w3 <- weight[, 5 : 6] %>%  rename("Body Wt" = V5, "Brain Wt" = V6) %>%
  mutate_if(is.factor, as.character) %>%
  mutate_if(is.character, as.numeric)
weight_dat <- w1 %>%
  bind_rows(w2) %>% bind_rows(w3)
head(weight_dat)
```

```
##   Body Wt Brain Wt
## 1   3.385     44.5
## 2   0.480     15.5
## 3   1.350      8.1
## 4 465.000    423.0
## 5  36.330    119.5
## 6  27.660    115.0
```

```r
weight_dat %>% str() %>% summary()
```

```
## 'data.frame':    63 obs. of  2 variables:
##  $ Body Wt : num  3.38 0.48 1.35 465 36.33 ...
##  $ Brain Wt: num  44.5 15.5 8.1 423 119.5 ...
```

```
## Length  Class   Mode
##      0   NULL   NULL
```

```r
url <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"
tomato <- read.csv(url, sep = "\t")
tomato < - tomato %>% mutate_if(is.factor, as.character)
```

```
## Warning in Ops.factor(left): '-' not meaningful for factors
```

```
## Warning in Ops.factor(left, right): '<' not meaningful for factors
```

```
##      X.this.needs.reformatting.to.read.into.Splus
## [1,]                                           NA
## [2,]                                           NA
## [3,]                                           NA
```

```r
size <- paste(tomato[1, ])
size <- strsplit(size, " ")
size <- size[[1]][size[[1]] != ""]
size <- as.numeric(size[1 : 3])
size <- as.vector(size)
```

```r
ife <- paste(tomato[2, ])
ife <- strsplit(ife, " ")
ife <- ife[[1]][ife[[1]] != ""]
ife[2 : 4] <- strsplit(ife[2: 4], ",")
name <- as.matrix(rep(ife[[1]], 3))
ife <- rbind(ife[[2]], ife[[3]], ife[[4]])
ife <- ife %>%
  cbind(name) %>%
  cbind(c(1000, 2000, 3000))

pusa <- paste(tomato[3, ])
pusa <- strsplit(pusa, " ")
pusa <- pusa[[1]][pusa[[1]] != ""]
pusa[2 : 4] <- strsplit(pusa[2:4], ",")
name <- as.matrix(rep(pusa[[1]], 3))
pusa <- rbind(pusa[[2]], pusa[[3]], pusa[[4]])
pusa <- pusa %>%
  cbind(name) %>%
  cbind(c(1000, 2000, 3000))
tomato_dat <- rbind(pusa, ife)
tomato_dat <- tomato_dat %>% as.data.frame()
colnames(tomato_dat) <- c("1st", "2nd", "3rd", "name", "plant density")
tomato_dat <- tomato_dat[, c("name", "plant density", "1st", "2nd", "3rd")]

tomato_dat[, 2 : 5] <- tomato_dat[, 2 : 5] %>%
  mutate_if(is.factor, as.character) %>%
  mutate_if(is.character, as.numeric)

head(tomato_dat)
```

```
##             name plant density  1st  2nd  3rd
## 1 PusaEarlyDwarf          1000  8.1  8.6 10.1
## 2 PusaEarlyDwarf          2000 12.7 13.7 11.5
## 3 PusaEarlyDwarf          3000 14.4 15.4 13.7
## 4         Ife\\#1          1000 16.1 15.3 17.5
## 5         Ife\\#1          2000 16.6 19.2 18.5
## 6         Ife\\#1          3000 20.8 18.0 21.0
```

```r
tomato_dat %>% str() %>% summary()
```

```
## 'data.frame':    6 obs. of  5 variables:
##  $ name         : Factor w/ 2 levels "Ife\\#1","PusaEarlyDwarf": 2 2 2 1 1 1
##  $ plant density: num  1000 2000 3000 1000 2000 3000
##  $ 1st          : num  8.1 12.7 14.4 16.1 16.6 20.8
##  $ 2nd          : num  8.6 13.7 15.4 15.3 19.2 18
##  $ 3rd          : num  10.1 11.5 13.7 17.5 18.5 21
```

```
## Length  Class   Mode
##      0   NULL   NULL
```

Problem 6

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.1
```

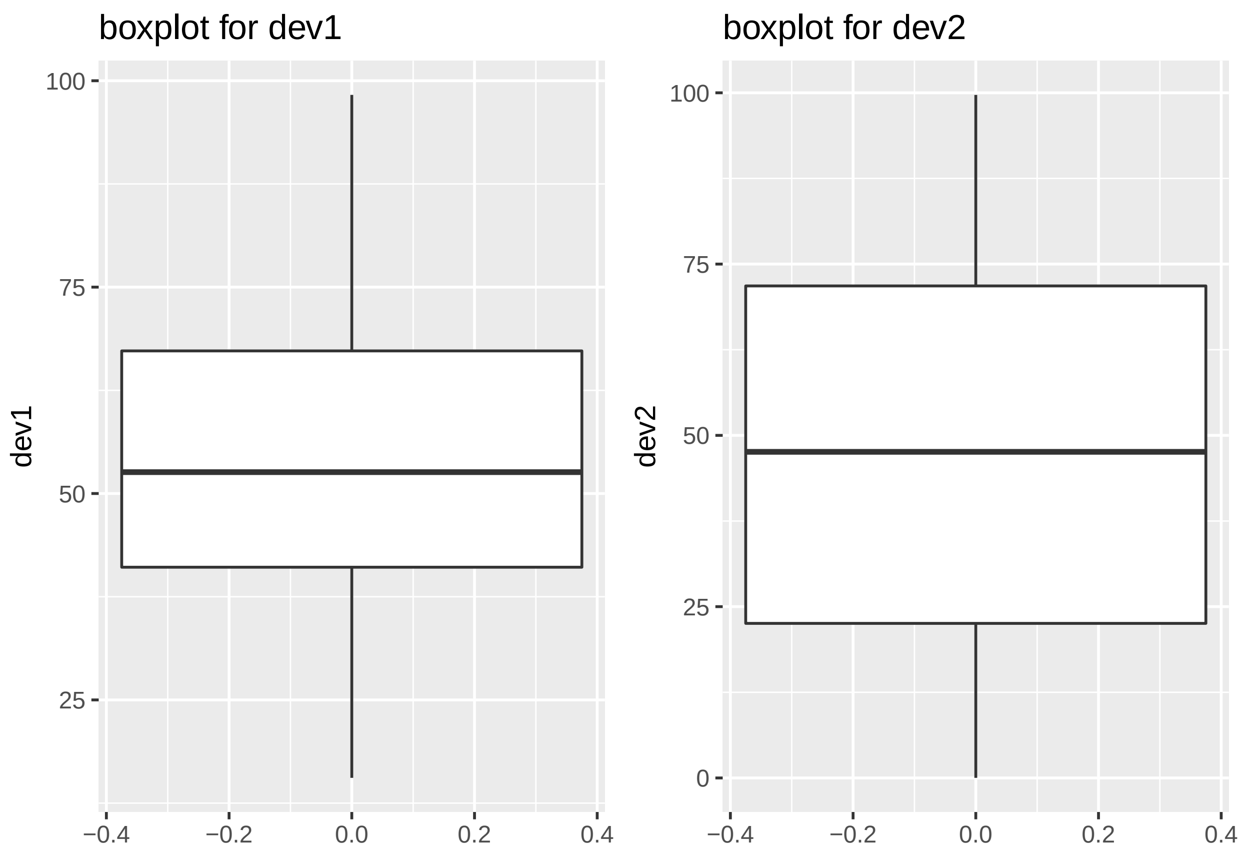
```r
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.5.3

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
dat <- readRDS("HW4_data.rds", refhook = NULL)
#dat <- dat_ori[1:13,]

#my function

func_stat <- function(){
  tab <- matrix(0, ncol = 5, nrow = 13)
for (i in 1:13) {
  mean <- mean(dat$dev1[which(dat$Observer==i)])
  mean2 <- mean(dat$dev2[which(dat$Observer==i)])
  sd <- sqrt(var(dat$dev1[which(dat$Observer==i)]))
  sd2 <- sqrt(var(dat$dev2[which(dat$Observer==i)]))
  corelation <- cor(dat$dev2[which(dat$Observer==i)], dat$dev1[which(dat$Observer==i)])
  tab[i,] <- cbind(mean, mean2, sd, sd2, corelation)
}
  tab <- as.data.frame(tab)
  names(tab) <- c("dev1 mean", "dev2 mean", "dev1 sd", "dev2 sd", "correlation")
  return(tab)
}
func_stat()
```

```
##    dev1 mean dev2 mean  dev1 sd  dev2 sd correlation
## 1   54.26610  47.83472 16.76982 26.93974 -0.06412835
## 2   54.26873  47.83082 16.76924 26.93573 -0.06858639
## 3   54.26732  47.83772 16.76001 26.93004 -0.06834336
## 4   54.26327  47.83225 16.76514 26.93540 -0.06447185
## 5   54.26030  47.83983 16.76774 26.93019 -0.06034144
## 6   54.26144  47.83025 16.76590 26.93988 -0.06171484
## 7   54.26881  47.83545 16.76670 26.94000 -0.06850422
## 8   54.26785  47.83590 16.76676 26.93610 -0.06897974
## 9   54.26588  47.83150 16.76885 26.93861 -0.06860921
## 10  54.26734  47.83955 16.76896 26.93027 -0.06296110
## 11  54.26993  47.83699 16.76996 26.93768 -0.06944557
## 12  54.26692  47.83160 16.77000 26.93790 -0.06657523
## 13  54.26015  47.83972 16.76996 26.93000 -0.06558334
```

```r
p <- ggplot(data = dat)
p1 <- p + geom_boxplot(aes(y = dev1)) + ggtitle("boxplot for dev1")
p2 <- p + geom_boxplot(aes(y = dev2)) + ggtitle("boxplot for dev2")
p3 <- p + geom_violin(aes(x = "",y = dev1)) + ggtitle("violin plot for dev2")
p4 <- p + geom_violin(aes(x = "",y = dev2)) + ggtitle("violin plot for dev2")

grid.arrange(p1, p2, ncol = 2)
```




```r
grid.arrange(p3, p4, ncol = 2)
```

Problem 7

Reimann sum is .8556252

```
func.1 <- function(x){
  y <- exp(- x ^ 2 / 2)
  return(y)
}
x <- seq(0, 1, by = 1e-6)
sum(func.1(x) * (1e-6))
```

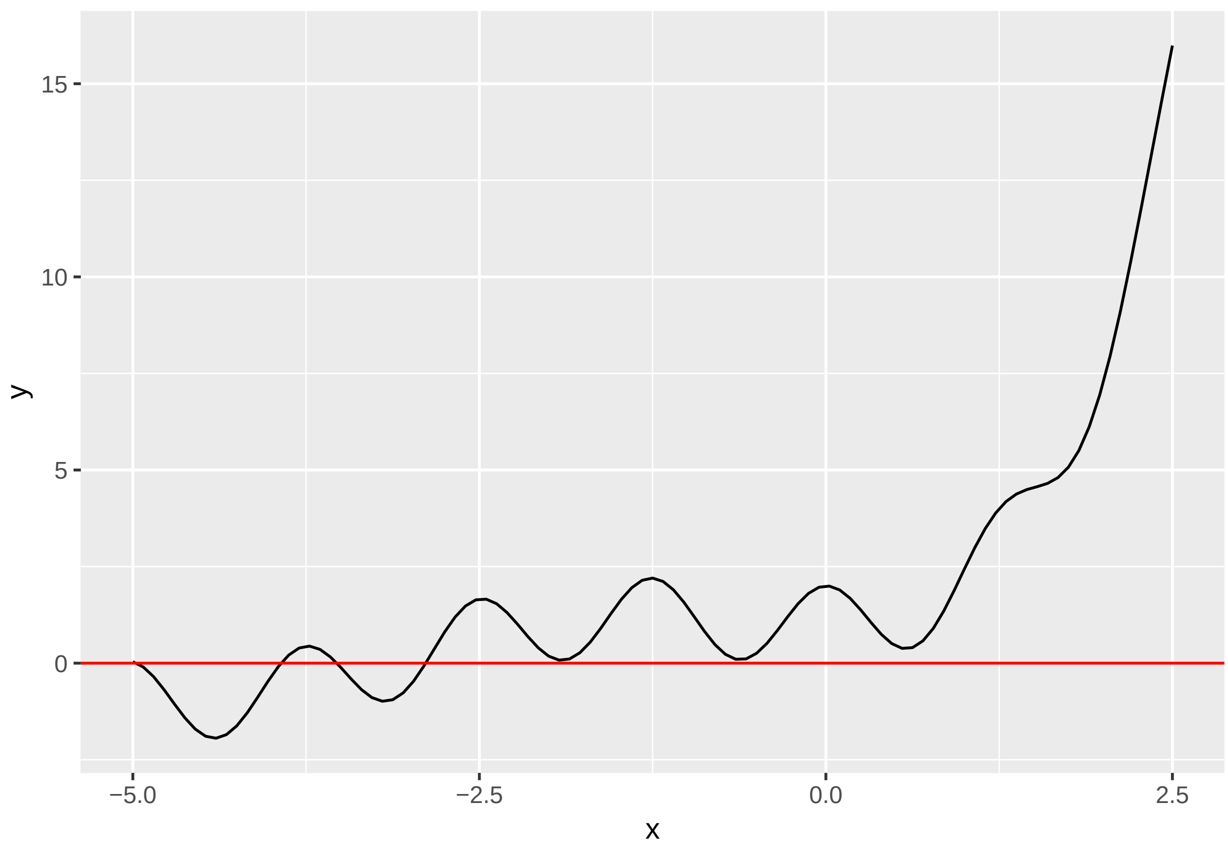
```
## [1] 0.8556252
```

Problem 8

One of solutions is x = - 3.930114

```
library(ggplot2)
func <- function(x){
  y <- 3 ^ x - sin(x) + cos(5 * x)
  return(y)
}
```

```r
d_func <- function(x){
  y <- 3 ^ x * log(3) - cos(x) - 5 * sin(5 * x)
  return(y)
}

ggplot(data = data.frame(x = 0,y = 0), mapping = aes(x = x)) +
  stat_function(fun = func) +
  xlim(-5, 2.5) +
  geom_abline(intercept = 0, slope = 0, colour = "red")
```




```r
x_0 <- -2.5
eps <- 1e-6
x <- x_0
while (abs(func(x)-0) > eps) {
  x <- x - func(x)/d_func(x)
}
x
```

```
## [1] -3.930114
```