HW2_linal20

Lina Lee

9/11/2020

Problem 3

In the lecture, there were two links to StackOverflow questions on why one should use version control. In your own words, summarize your thoughts (2-3 sentences) on version control in your future work. No penalties here if you say, useless!

I would say that version control is useful although we work alone. This is because we can save snapshot of the whole hard drive in version control system. Also, if you have an error, you can look at the previous version of code and debug. Also, these days, more and more people collaborate each other. By using version control system, other people can access your analysis.

Problem 4

Make sure you weave your code and text into a complete description of the process and end by creating a tidy dataset describing the variables, create a summary table of the data (summary, NOT full listing), note issues with the data, and include an informative plot.

a. Sensory data from five operators. – see video, I am doing this one

(1) with base R

Here, each item has 15 values. some elements are NA. I want to remove the NA's. However, I should not remove a row with NA, because I remove other elements in the row. so,I transformed the data into a vector to remove element with NA. After removing NA's, I transformed it back to matrix format. In the new matrix, each row represent each item. so each row for each item includes 15 values. From here, I separated data by row. separated row is a vector of 15 values. I transformed this into 3 by 5 matrix format. here, each columns represent each operator. Given total 5 rows, total 5 number of 3 by 5 matrix has been made. I add a columns of item number for each matrix. I combined those matrix by row. Still, columns represent operator; operator1, operator2, operator3, operator4, opertoar5. operator need to be one variable, so I transformed it from wide to long formate by using stack function.

```
library(RCurl)

# read data into R
mydat <- as.vector(read.delim("https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat",skip=

# transform data into vector
mydat<-as.vector(t(data.matrix(mydat)))

# remove NA elements
```

```r
mydat<-mydat[!is.na(mydat)]

# transform vector into the matrix form
# now each row is for a item
newdat<-as.data.frame(t(matrix(mydat,ncol = 10)))

# transform data so that each columns represent item or an operator
data_old<-matrix(NA,ncol=6)
for(i in 1:10){
data_new<-cbind(rep(i,3),t(matrix(newdat[i,][-1],5)))
data_old<-rbind(data_old,data_new)
}

# remove initialized empty row
updateDat<-as.data.frame(data_old[-1,])

#change the columns names
names(updateDat)<-c("item","1","2","3","4","5")


# change the columns types
updateDat$item<-as.factor(unlist(updateDat$item))
updateDat$"1"<-unlist(updateDat$"1")
updateDat$"2"<-unlist(updateDat$"2")
updateDat$"3"<-unlist(updateDat$"3")
updateDat$"4"<-unlist(updateDat$"4")
updateDat$"5"<-unlist(updateDat$"5")

# stack data so that we can have one columns for operator
stackDat<-stack(updateDat[,-1])

# combined data with the vector for item
finaldf<-cbind(unlist(rep(updateDat[,1],5)),stackDat)

#change the columns' positions
finaldf <- finaldf[, c(1, 3, 2)]

# change the columns' names
names(finaldf)<-c("item","operator","values")
head(finaldf)
```

```
##   item operator values
## 1    1        1    4.3
## 2    1        1    4.3
## 3    1        1    4.1
## 4    2        1    6.0
## 5    2        1    4.9
## 6    2        1    6.0
```

(2) with function in tidyverse

I used the data “updateDat” that was read above using R base here. I transformed the data from wide to long format using gather function in tidyverse.

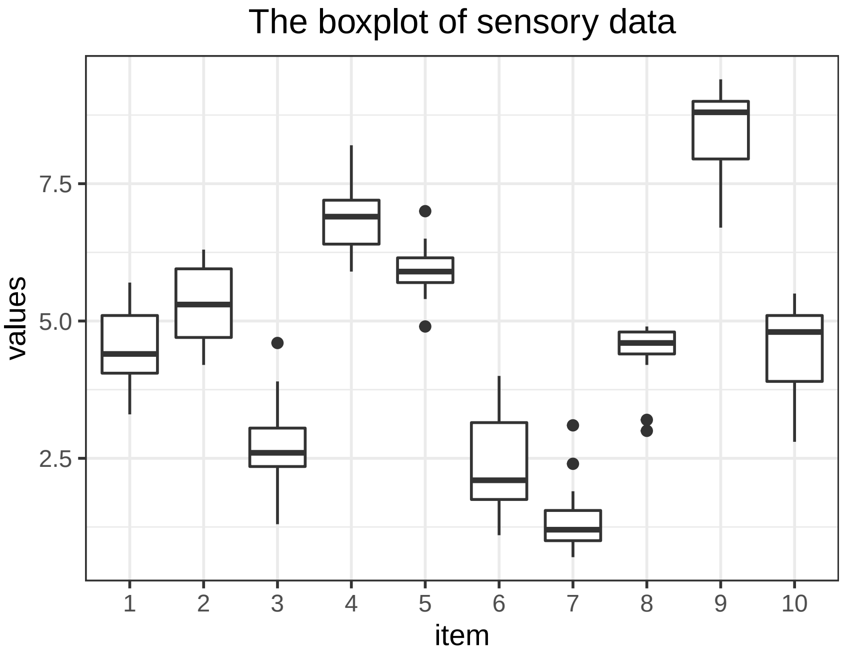
```
library(tidyverse)

# I used the data "updateDat" that was read above using R base here

# transform data from wide to long format
df2_tiver<-updateDat%>%gather(key="operator", value = "values",2:6)

# transform above data to the R dataframe
df2_tiver<-as.data.frame(df2_tiver)
```

(3) data summary table

item	operator	values
1 :15	Length:150	Min. :0.700
2 :15	Class :character	1st Qu.:3.025
3 :15	Mode :character	Median :4.700
4 :15	NA	Mean :4.657
5 :15	NA	3rd Qu.:6.000
6 :15	NA	Max. :9.400
(Other):60	NA	NA

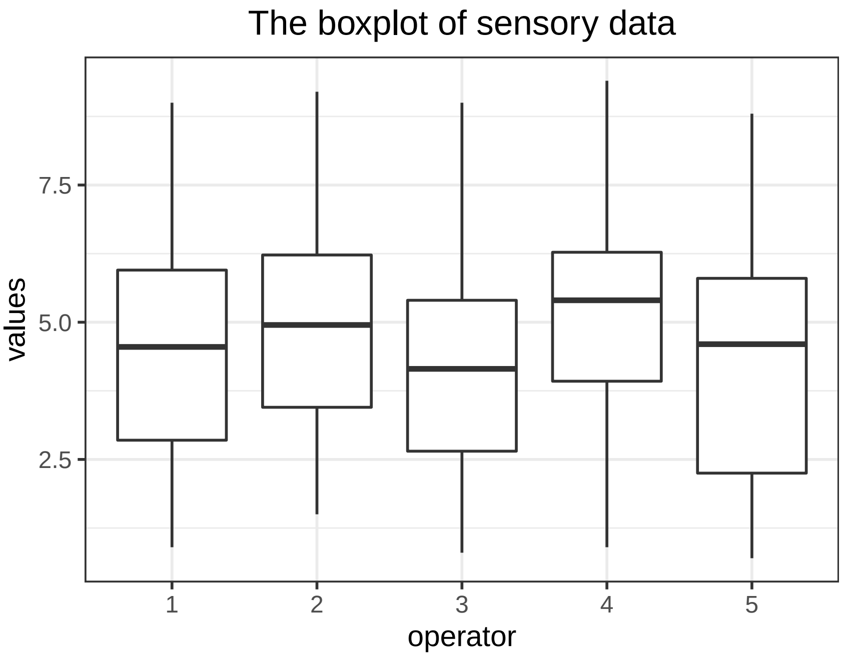
(4) Informative plot

```
ggplot()+
  geom_boxplot(aes(x=item,y=values),data=df2_tiver)+
  labs(title="The boxplot of sensory data")+
  theme_bw()+
   theme(plot.title = element_text(hjust = 0.5))
```



The distribution of values vary for item.

```
ggplot()+
  geom_boxplot(aes(x=operator,y=values),data=df2_tiver)+
  labs(title="The boxplot of sensory data")+
  theme_bw()+
  theme(plot.title = element_text(hjust = 0.5))
```



The median of values are vary by operator. The median of value for operator 2 and 4 are higher than the values of the other operators

b. Gold Medal performance for Olympic Men's Long Jump, year is coded as 1900=0.

(1) with base R

After reading data, I sliced data by two columns and combined them by row. I also removed empty rows and changed the values of the variable "year" into real year format by adding 1900.

```
library(data.table)
url<-"https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
JumpData<-fread(url)
```

```
## Warning in fread(url): Detected 12 column names but the data has 8 columns.
## Filling rows automatically. Set fill=TRUE explicitly to avoid this warning.
```

```
# read data
JumpData<-read.delim(url, sep=' ',header=FALSE,skip=1)

# change columns' names
names(JumpData)<-c("year","LongJump","year","LongJump","year","LongJump","year","LongJump")

# slice data by two columns and combined them by row
newJumpDat<-do.call(rbind,list(JumpData[,1:2],JumpData[,3:4],JumpData[,5:6],JumpData[,7:8]))

# remove empty rows
newJumpDat<-newJumpDat[-c(23,24),]

# change values of the variable "year" into real year format
newJumpDat$year<-newJumpDat$year+1900
head(newJumpDat)
```

```
##   year LongJump
## 1 1896   249.75
## 2 1900   282.88
## 3 1904   289.00
## 4 1908   294.50
## 5 1912   299.25
## 6 1920   281.50
```

(2) with function in tidyverse

I selected consecutive two columns (columns1 and 2, 3 and 4, 5 and 6, 7 and 8) from jumpData and save it into a list. so, the list consists of 4 elements, and each element has a dataset of two columns. The list has total 4 elements. Then, I combined the dataset of two columns by rows that had been saved in list. I combined the first element and the second element of the list by row. Like this I combined every element of list together.

```
# I used "jumpData" I read above using base R

x<-list()

# select consecutive two columns (columns1 and 2, 3 and 4, 5 and 6, 7 and 8) and save it into a list
for(i in c(1,3,5,7)){
x[[(i-1)/2+1]]<-JumpData%>%select(c(i,i+1))
}
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(i)` instead of `i` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```r
# combined the columns that are save in list by row
JumpDat_tidv<-rbind(x[[1]],x[[2]],x[[3]],x[[4]])
head(JumpDat_tidv)
```

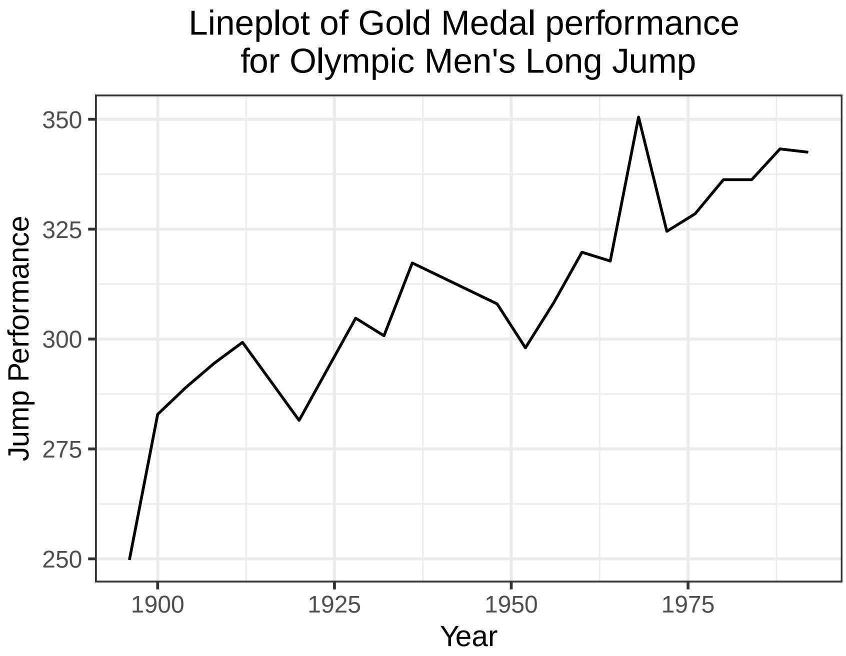
```
##   year LongJump
## 1   -4   249.75
## 2    0   282.88
## 3    4   289.00
## 4    8   294.50
## 5   12   299.25
## 6   20   281.50
```

(3) data summary table

year	LongJump
Min. :1896	Min. :249.8
1st Qu.:1921	1st Qu.:295.4
Median :1950	Median :308.1
Mean :1945	Mean :310.3
3rd Qu.:1971	3rd Qu.:327.5
Max. :1992	Max. :350.5

(4) Informative plot

```r
ggplot()+
  geom_line(aes(x=year,y=LongJump),
            data=newJumpDat)+
  labs(title="Lineplot of Gold Medal performance \nfor Olympic Men's Long Jump",
       y="Jump Performance",
       x="Year")+
  theme_bw()+
  theme(plot.title = element_text(hjust = 0.5))
```


Lineplot of Gold Medal performance
for Olympic Men's Long Jump
Jump Performance
350
325
300
275
250
1900
1925
1950
1975
Year

c. Brain weight (g) and body weight (kg) for 62 species.

###(1) with base R similar with the b., I sliced dataset by two columns and combined them by row.

```
# read data into R
wtData<-read.delim("https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat",sep=

# change columns names
names(wtData)<-c("bodywt","brainwt","bodywt","brainwt","bodywt","brainwt")

# slice the data by two consecutive columns and combined them by row.
nwtDat<-do.call(rbind,list(wtData[,1:2],wtData[,3:4],wtData[,5:6]))

# remove the empty row
nwtDat<-nwtDat[-63,]
head(nwtDat)
```

```
##    bodywt brainwt
## 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
```

(2) with function in tidyverse

Again, this has been done like b. I selected consecutive two columns (columns1 and 2, 3 and 4, 5 and 6) from wtData using "select" function in tidyverse and save it into a list. so, the list consists of 3 elements, and each element has a dataset of two columns. Then, I combined the dataset of two columns by rows that had been saved in list. I combined the first element and the second element of the list by row. Like this I combined every element of list together.

```
newWt_tidv<-list()

# I used wtData that I read above using base R functions

# slice the data by two consecutive columns and combined them by row.
for(i in c(1,3,5)){
newWt_tidv[[(i-1)/2+1]]<-wtData%>%select(c(i,i+1))
}

# combined them by row
newWt_tidv<-rbind(newWt_tidv[[1]],newWt_tidv[[2]],newWt_tidv[[3]])
head(newWt_tidv)
```

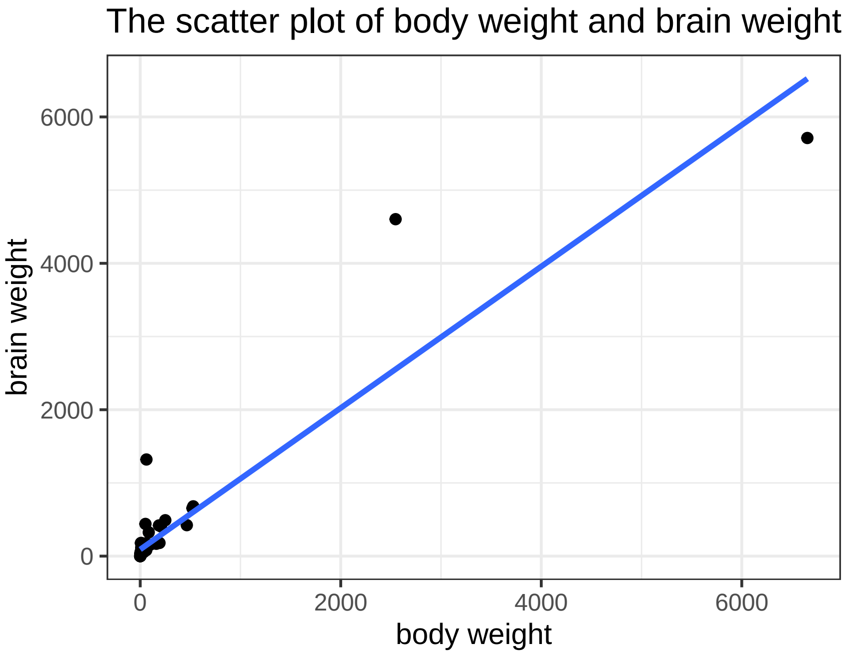
```
##    bodywt brainwt
## 1   3.385    44.5
## 2   0.480    15.5
## 3   1.350     8.1
## 4 465.000   423.0
## 5  36.330   119.5
## 6  27.660   115.0
```

(3) data summary table

bodywt	brainwt
Min. : 0.005	Min. : 0.10
1st Qu.: 0.600	1st Qu.: 4.25
Median : 3.342	Median : 17.25
Mean : 198.790	Mean : 283.13
3rd Qu.: 48.202	3rd Qu.: 166.00
Max. :6654.000	Max. :5712.00
NA's :1	NA's :1

(4) Informative plot

```
ggplot(aes(x=bodywt,y=brainwt),data=nwtDat)+
  geom_point()+
  geom_smooth(method='lm', formula= y~x,se = FALSE)+
  theme_bw()+
  labs(title="The scatter plot of body weight and brain weight", x="body weight",y="brain weight")+
  theme(plot.title = element_text(hjust = 0.5))
```



d. Triplicate measurements of tomato yield for two varieties of tomatos at three planting densities.

(1) with base R

First, I read data as a vectors of characters into R. I removed the elements with " “. The characters includes variable name or values, so I selected elements with values.After that, I made a loop. each loop has been operate with measurement values for each combination of item & density. each element is a character consisting of several values seperated by”,“. Therefore, I splited each value by”," in the loop. and then I removed whitespace in characters and add the columns of density of tomato into the data made above. the dataset made in each loop has been saved in an element of a list. Each element of the list includes dataset made in each loop iteration. Finally, every element of the list has been combined to make a final dataset.

```
# read data as a vectors of characters into R
split<-unlist(strsplit(readLines("https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"),

# remove the elements with " "
split<-split[split != ""]

#select elements with values
value<-split[c(6,7,8,10,11,12)]


valList<-list()

# operate this loop with measurement values for each combination of item & density ( m represent a comb
for(m in 1:6){


char<-value[m]
# change list into vector
# split each value by "," so that each value in new list has one value
char<-unlist(strsplit(char,",", fixed =TRUE))

# remove white space in character and transform vector type from character into numeric
valList[[m]]<-as.numeric(as.vector(str_replace_all(string=char, pattern=" ", repl="")))
}

# add the columns of density of tomato into the data made above
bindmat<-as.data.frame(cbind(matrix(unlist(valList),ncol=2),c(rep("10000",3),rep("20000",3),rep("30000"

names(bindmat)<-c("type1","type2","density")
```

(2) with function in tidyverse

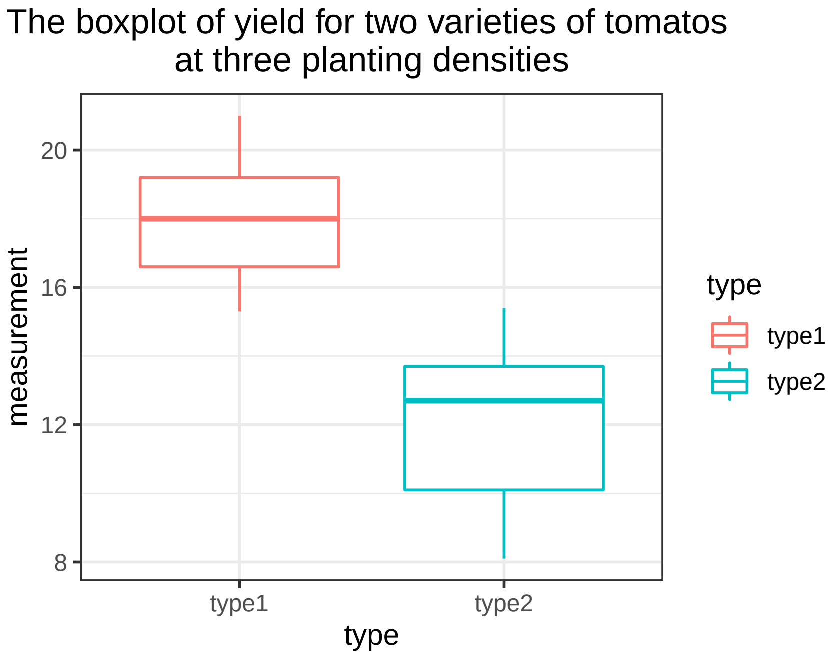
```
tomato_tidv<-bindmat%>%gather(key="type",value="measurement",1:2)%>%
  mutate(density=as.factor(density),type=as.factor(type),measurement=as.numeric(measurement))
```

(3) data summary table

density	type	measurement
10000:6	type1:9	Min. : 8.10
20000:6	type2:9	1st Qu.:12.95
30000:6	NA	Median :15.35
NA	NA	Mean :15.07
NA	NA	3rd Qu.:17.88
NA	NA	Max. :21.00

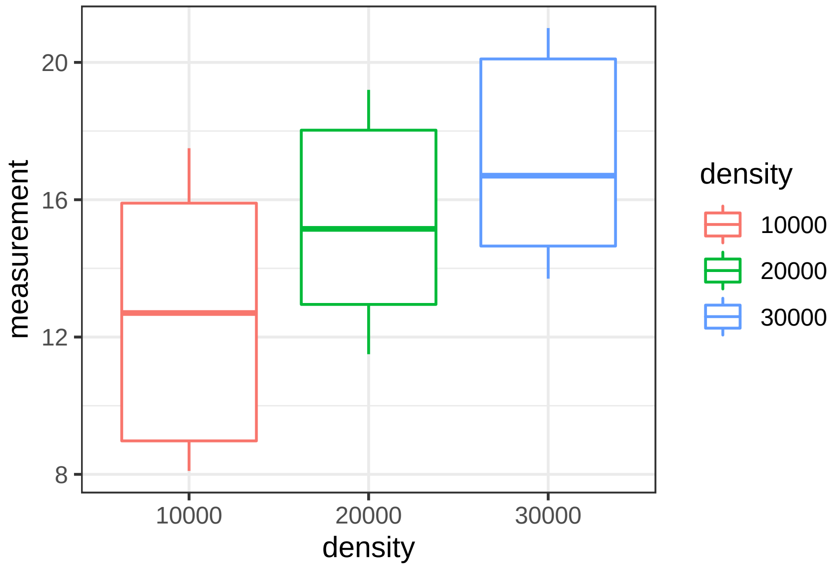
(4) Informative plot

```
ggplot()+
  geom_boxplot(aes(x=type,y=measurement,color=type),
               data=tomato_tidv)+
  labs(title="The boxplot of yield for two varieties of tomatos \nat three planting densities")+
  theme_bw()+
  theme(plot.title = element_text(hjust = 0.5))
```



```
ggplot()+
  geom_boxplot(aes(x=density,y=measurement,color=density),data=tomato_tidv)+
  labs(title="The boxplot of  yield for two varieties of tomatos \nat three planting densities")+
  theme_bw()+
  theme(plot.title = element_text(hjust = 0.5))
```

The boxplot of yield for two varieties of tomatos
at three planting densities


measurement
density
density