

# Homework 2

Youngjin Cho

2020 9 13

## Problem 1

I took “Work with Data” and “Tidy Your Data” in Primers.

## Problem 2

I saved this file as HW2\_youngjin.

## Problem 3

Virsion Control is important in my future work. If I do not use Virsion Control, when I make mistake, I might not find when the mistake happened.

## Problem 4

### (a) Sensory Data

```
## Sensory Import
url_Sensory <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
Sensory_raw <- fread(url_Sensory, fill=TRUE, data.table = FALSE, skip=1)
knitr::kable(head(Sensory_raw))
```

Item	1	2	3	4	5
1.0	4.3	4.9	3.3	5.3	4.4
4.3	4.5	4.0	5.5	3.3	NA
4.1	5.3	3.4	5.7	4.7	NA
2.0	6.0	5.3	4.5	5.9	4.7
4.9	6.3	4.2	5.5	4.9	NA
6.0	5.9	4.7	6.3	4.6	NA

I imported data from the url. I see the data is kind of uncleaned. We cannot directly use the data because some rows have values in item but the other rows have no values in item, which makes the whole data mess, i.e., some data is located in one column left from its original location. We should fix this problem by cleaning the data.

## Using Basic R Functions

```
## Sensory data : old school : index data frame
Sensory_old_school <- Sensory_raw
Sorthelper <- data.frame(1:length(Sensory_old_school[,6]),as.numeric(is.na(Sensory_old_school[,6])))
colnames(Sorthelper) <- c("index", "NA")
head(Sorthelper)
```

```
##   index NA
## 1     1  0
## 2     2  1
## 3     3  1
## 4     4  0
## 5     5  1
## 6     6  1
```

First, we can use kind of old school way, which do not use tidyverse. We make index data frame that indicates the rows that have NA value in last column of the data. If ith row has NA value in its last column in the original data, the ith row and second column of Sorthelper will become 1, otherwise 0.

```
## Sensory data : old school : data cleaning with index
Sensory_old_school[Sorthelper[Sorthelper[,2]==1,][,1],2:6] <-
  Sensory_old_school[Sorthelper[Sorthelper[,2]==1,][,1],1:5]
Sensory_old_school <- Sensory_old_school[,-1]
colnames(Sensory_old_school) <- paste("Operator",1:5)
knitr::kable(head(Sensory_old_school))
```

Operator 1	Operator 2	Operator 3	Operator 4	Operator 5
4.3	4.9	3.3	5.3	4.4
4.3	4.5	4.0	5.5	3.3
4.1	5.3	3.4	5.7	4.7
6.0	5.3	4.5	5.9	4.7
4.9	6.3	4.2	5.5	4.9
6.0	5.9	4.7	6.3	4.6

By using Sorthelper as index, we can sort our data which has NA in last column and push these rows to the right by one column. Then we can get cleaned data!

## Using Tidyverse

```
## Sensory data : tidyverse : dividing data
Sensory_new <- cbind(1:length(Sensory_raw[,6]),Sensory_raw)
colnames(Sensory_new) <- c("number_key","trash",paste("Operator", 1:5))

Sensory_A <- Sensory_new %>%
  subset(is.na(Sensory_new[,7]))

Sensory_B <- Sensory_new %>%
  subset(Sensory_new[,7]>0)

Sensory_A <- Sensory_A[,-7]
Sensory_B <- Sensory_B[,-2]
```

I used subset function in tidyverse to clean data. First, I divided data into two parts, one is Sensory\_A, the rows have NA values in last column, and the other is Sensory\_B, the rows do not have NA values in last column. I cleaned these two data independently. Now we can combine them together.

```
## Sensory data : tidyverse : combining data
colnames(Sensory_A) <- colnames(Sensory_B)
Sensory_new <- rbind(Sensory_A,Sensory_B)
Sensory_new <- Sensory_new %>%
  arrange(number_key) %>%
  select(-1)
knitr::kable(head(Sensory_new))
```

Operator 1	Operator 2	Operator 3	Operator 4	Operator 5
4.3	4.9	3.3	5.3	4.4
4.3	4.5	4.0	5.5	3.3
4.1	5.3	3.4	5.7	4.7
6.0	5.3	4.5	5.9	4.7
4.9	6.3	4.2	5.5	4.9
6.0	5.9	4.7	6.3	4.6

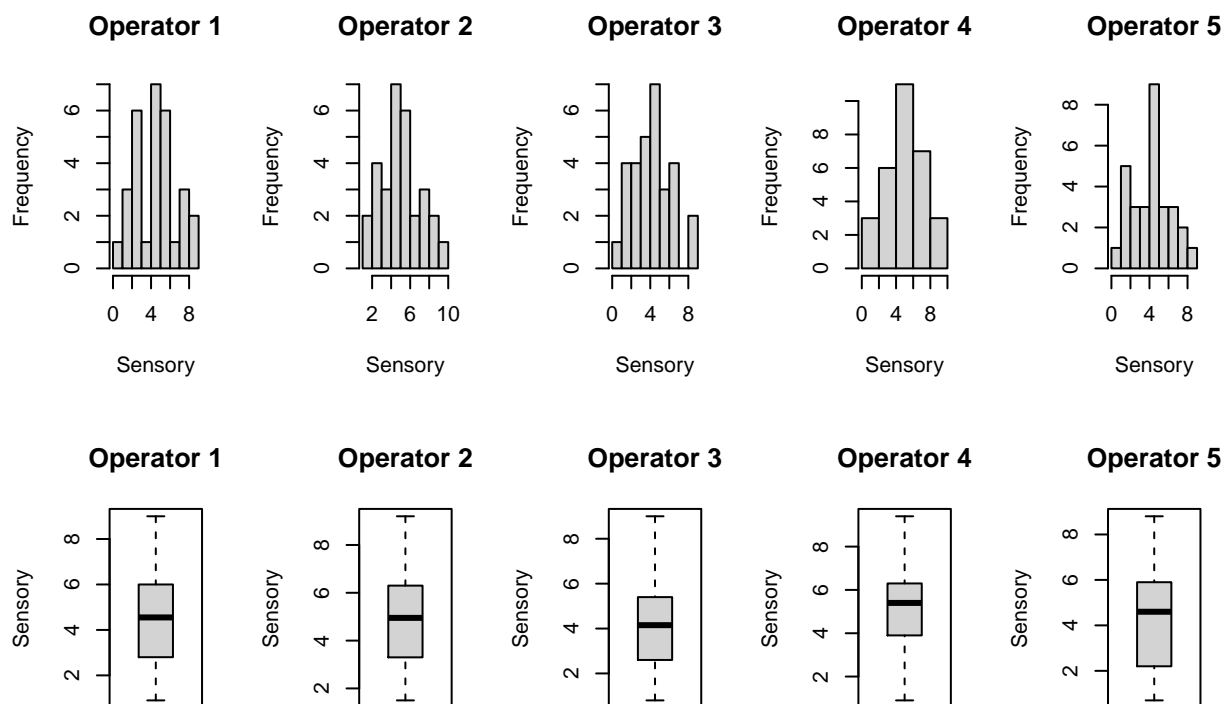
I combined two data and arranged in original order by using number\_key column. It is the same as the result using old school way!

## Data Description

```
## Sensory data : summary
knitr::kable(summary(Sensory_new))
```

Operator 1	Operator 2	Operator 3	Operator 4	Operator 5
Min. :0.900	Min. :1.500	Min. :0.800	Min. :0.900	Min. :0.700
1st Qu.:2.850	1st Qu.:3.450	1st Qu.:2.650	1st Qu.:3.925	1st Qu.:2.250
Median :4.550	Median :4.950	Median :4.150	Median :5.400	Median :4.600
Mean :4.593	Mean :5.063	Mean :4.167	Mean :5.193	Mean :4.267
3rd Qu.:5.950	3rd Qu.:6.225	3rd Qu.:5.400	3rd Qu.:6.275	3rd Qu.:5.800
Max. :9.000	Max. :9.200	Max. :9.000	Max. :9.400	Max. :8.800

```
## Sensory data : histogram and boxplot
par(mfrow=c(2,5))
hist(Sensory_new[,1],main="Operator 1", xlab="Sensory")
hist(Sensory_new[,2],main="Operator 2", xlab="Sensory")
hist(Sensory_new[,3],main="Operator 3", xlab="Sensory")
hist(Sensory_new[,4],main="Operator 4", xlab="Sensory")
hist(Sensory_new[,5],main="Operator 5", xlab="Sensory")
boxplot(Sensory_new[,1],main="Operator 1", ylab="Sensory")
boxplot(Sensory_new[,2],main="Operator 2", ylab="Sensory")
boxplot(Sensory_new[,3],main="Operator 3", ylab="Sensory")
boxplot(Sensory_new[,4],main="Operator 4", ylab="Sensory")
boxplot(Sensory_new[,5],main="Operator 5", ylab="Sensory")
```



I summarized the Sensory data and plotted histogram and boxplot. No special issue was found.

## (b) Long Jump Data

```
## Long Jump import
url_Longjump <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
Longjump_raw <- fread(url_Longjump)

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

knitr::kable(head(Longjump_raw))
```

Year	Long	Jump	Year	Long	Jump	Year	Long	Jump	Year	Long	Jump
-4	249.75	24	293.13	56	308.25	80	336.25	NA	NA	NA	NA
0	282.88	28	304.75	60	319.75	84	336.25	NA	NA	NA	NA
4	289.00	32	300.75	64	317.75	88	343.25	NA	NA	NA	NA
8	294.50	36	317.31	68	350.50	92	342.50	NA	NA	NA	NA
12	299.25	48	308.00	72	324.50	NA	NA	NA	NA	NA	NA
20	281.50	52	298.00	76	328.50	NA	NA	NA	NA	NA	NA

I imported data from the url. I see the data is kind of uncleaned. We cannot directly use the data because columns with same characteristics are repeated, i.e., year and record columns are repeated. Also, in later columns, the value for year and record are NA values. We should fix this problem by cleaning the data.

## Using Basic R Functions

```
## Long Jump data : old school : Vectorization -> Matrix form -> Data Frame
Longjump_old_school <- as.matrix(Longjump_raw)
Longjump_old_school <- as.vector(t(Longjump_old_school))
Longjump_old_school <- matrix(Longjump_old_school,2,length(Longjump_old_school)/2)
Longjump_old_school <- as.data.frame(t(Longjump_old_school))
head(Longjump_old_school)
```

```
##   V1      V2
## 1 -4 249.75
## 2 24 293.13
## 3 56 308.25
## 4 80 336.25
## 5 NA      NA
## 6 NA      NA
```

First, I vectorized the data by as.vector function and made 2 by 22 matrix with it and transformed it to data frame.

```
## Long Jump data : old school : data cleaning with index
Sorthelper_2 <- data.frame(1:length(Longjump_old_school[,2]),1-as.numeric(is.na(Longjump_old_school[,2])),
colnames(Sorthelper_2) <- c("index", "non NA")

Longjump_old_school <- Longjump_old_school[Sorthelper_2[Sorthelper_2[,2]==1,][,1],]
Longjump_old_school[,1] <- Longjump_old_school[,1]+1900
Longjump_old_school=Longjump_old_school[order(Longjump_old_school[,1]),]

rownames(Longjump_old_school) <- 1:dim(Longjump_old_school)[1]
colnames(Longjump_old_school) <- c("Year", "Record")

knitr::kable(head(Longjump_old_school))
```

Year	Record
1896	249.75
1900	282.88
1904	289.00
1908	294.50
1912	299.25
1920	281.50

Second, similar to Sensory data cleaning, I made index matrix (Sorthelper\_2) to make indices of rows with no NA values. I filtered the data with this index and added 1900 to the year column to show year properly. Then we can get cleaned data!

## Using Tidyverse

```
## Long Jump data : tidyverse : Vectorization -> Matrix form -> Data Frame
Longjump_new <- as.matrix(Longjump_raw)
Longjump_new <- as.vector(t(Longjump_new))
Longjump_new <- matrix(Longjump_new,2,length(Longjump_new)/2)
```

```
Longjump_new <- as.data.frame(t(Longjump_new))
head(Longjump_new)
```

```
##   V1      V2
## 1 -4 249.75
## 2 24 293.13
## 3 56 308.25
## 4 80 336.25
## 5 NA      NA
## 6 NA      NA
```

Now we use tidyverse. The first step is same as old school, which I vectorized the data by as.vector function and made 2 by 22 matrix with it and transformed it to data frame.

```
## Long Jump data : tidyverse : data cleaning with subset
Longjump_new <- Longjump_new %>%
  subset(Longjump_new[,2]>0)

Longjump_new$V1 <- Longjump_new$V1+1900

rownames(Longjump_new) <- 1:dim(Longjump_new)[1]
colnames(Longjump_new) <- c("Year", "Record")

Longjump_new <- Longjump_new %>%
  arrange(Year)
knitr::kable(head(Longjump_new))
```

Year	Record
1896	249.75
1900	282.88
1904	289.00
1908	294.50
1912	299.25
1920	281.50

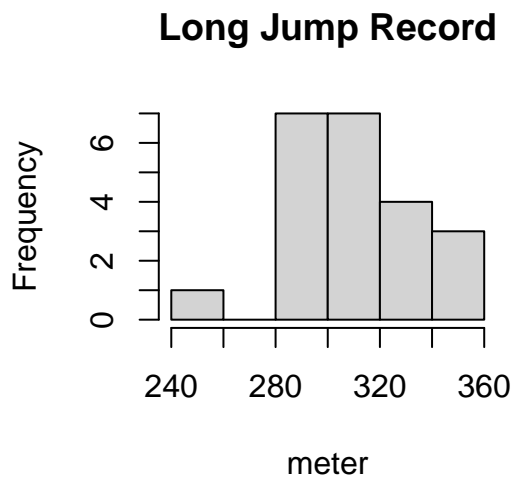
I used subset function in tidyverse to get the rows without NA values. I added 1900 to the year column to show year properly. Then we can get cleaned data! It is same as the result in old school way.

## Data Description

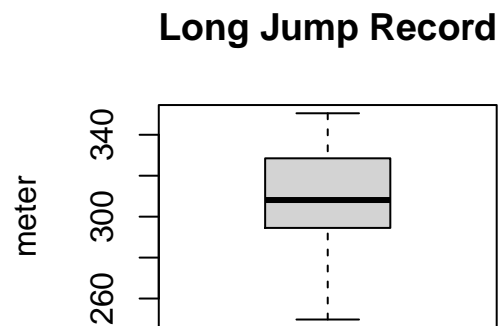
```
## Long Jump data : summary
knitr::kable(summary(Longjump_new))
```

Year	Record
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

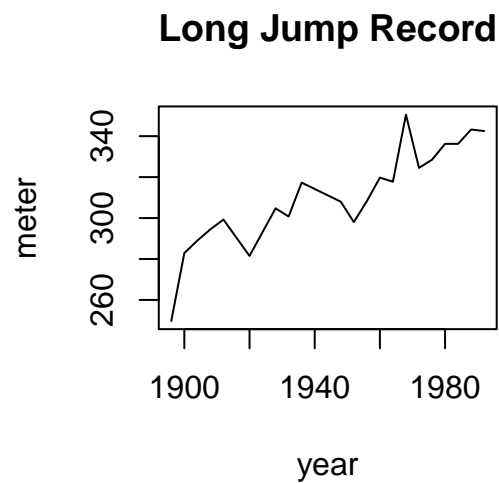
```
## Long Jump data : histogram and boxplot
par(mfrow=c(1,1))
hist(Longjump_new[,2],main="Long Jump Record", xlab="meter")
```



```
boxplot(Longjump_new[,2],main="Long Jump Record", ylab="meter")
```



```
## Long Jump data : year vs record
par(mfrow=c(1,1))
plot(Longjump_new[,1],Longjump_new[,2],main="Long Jump Record", xlab="year", ylab="meter", type="l")
```



I summarized the Long Jump data and plotted histogram, boxplot, and year versus record plot. The record improves as the year passes. There were no special issues for the data.

### (c) Brain Body Data

```
## Brain Body import
url_Brain <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
Brain_raw <- fread(url_Brain)

## Warning in fread(url_Brain): Detected 12 column names but the data has 6
## columns. Filling rows automatically. Set fill=TRUE explicitly to avoid this
## warning.

knitr::kable(head(Brain_raw))
```

Body	Wt	Brain	Wt	Body	Wt	Brain	Wt	Body	Wt	Brain	Wt
3.385	44.5	521.000	655.0	2.50	12.1	NA	NA	NA	NA	NA	NA
0.480	15.5	0.785	3.5	55.50	175.0	NA	NA	NA	NA	NA	NA
1.350	8.1	10.000	115.0	100.00	157.0	NA	NA	NA	NA	NA	NA
465.000	423.0	3.300	25.6	52.16	440.0	NA	NA	NA	NA	NA	NA
36.330	119.5	0.200	5.0	10.55	179.5	NA	NA	NA	NA	NA	NA
27.660	115.0	1.410	17.5	0.55	2.4	NA	NA	NA	NA	NA	NA

I imported data from the url. It is actually the same problem as cleaning the Long Jump Record data. We cannot directly use the data because columns with same characteristics are repeated, i.e., Brain weight and Body weight columns are repeated. Also, in later columns, the value for year and record are NA values. We should fix this problem by cleaning the data.

#### Using Basic R Functions

```
## Brain Body data : old school : Vectorization -> Matrix form -> Data Frame
Brain_old_school <- as.matrix(Brain_raw)
Brain_old_school <- as.vector(t(Brain_old_school))
Brain_old_school <- matrix(Brain_old_school,2,length(Brain_old_school)/2)
Brain_old_school <- as.data.frame(t(Brain_old_school))
head(Brain_old_school)
```

```
##      V1      V2
## 1  3.385  44.5
## 2 521.000 655.0
## 3   2.500  12.1
## 4      NA     NA
## 5      NA     NA
## 6      NA     NA
```

First, I vectorized the data by as.vector function and made 2 by 62 matrix with it and transformed it to data frame.

```
## Brain Body data : old school : data cleaning with index
Sorthelper_3 <- data.frame(1:length(Brain_old_school[,2]),1-as.numeric(is.na(Brain_old_school[,2])))
Brain_old_school <- Brain_old_school[Sorthelper_3[Sorthelper_3[,2]==1,],[1,]]

rownames(Brain_old_school) <- 1:dim(Brain_old_school)[1]
colnames(Brain_old_school) <- c("Brain weight (g)", "Body weight (kg)")

knitr::kable(head(Brain_old_school))
```



Brain weight (g)	Body weight (kg)
3.385	44.5
521.000	655.0
2.500	12.1
0.480	15.5
0.785	3.5
55.500	175.0

Second, similar to Sensory data cleaning, I made index matrix (Sorthelper\_3) to make indices of rows with no NA values. I filtered the data with this index and we can get cleaned data!

## Using Tidyverse

```
## Brain Body data : tidyverse : Vectorization -> Matrix form -> Data Frame
Brain_new <- as.matrix(Brain_raw)
Brain_new <- as.vector(t(Brain_new))
Brain_new <- matrix(Brain_new,2,length(Brain_new)/2)
Brain_new <- as.data.frame(t(Brain_new))
head(Brain_new)
```

```
##      V1    V2
## 1  3.385 44.5
## 2 521.000 655.0
## 3   2.500 12.1
## 4      NA   NA
## 5      NA   NA
## 6      NA   NA
```

Now we use tidyverse. The first step is same as old school, which I vectorized the data by as.vector function and made 2 by 62 matrix with it and transformed it to data frame.

```
## Brain Body data : tidyverse : data cleaning with subset
Brain_new <- Brain_new %>%
  subset(Brain_new[,2]>0)

colnames(Brain_new) <- c("Brain weight (g)", "Body weight (kg)")
rownames(Brain_new) <- 1:dim(Brain_new)[1]

knitr::kable(head(Brain_new))
```

Brain weight (g)	Body weight (kg)
3.385	44.5
521.000	655.0
2.500	12.1
0.480	15.5
0.785	3.5
55.500	175.0

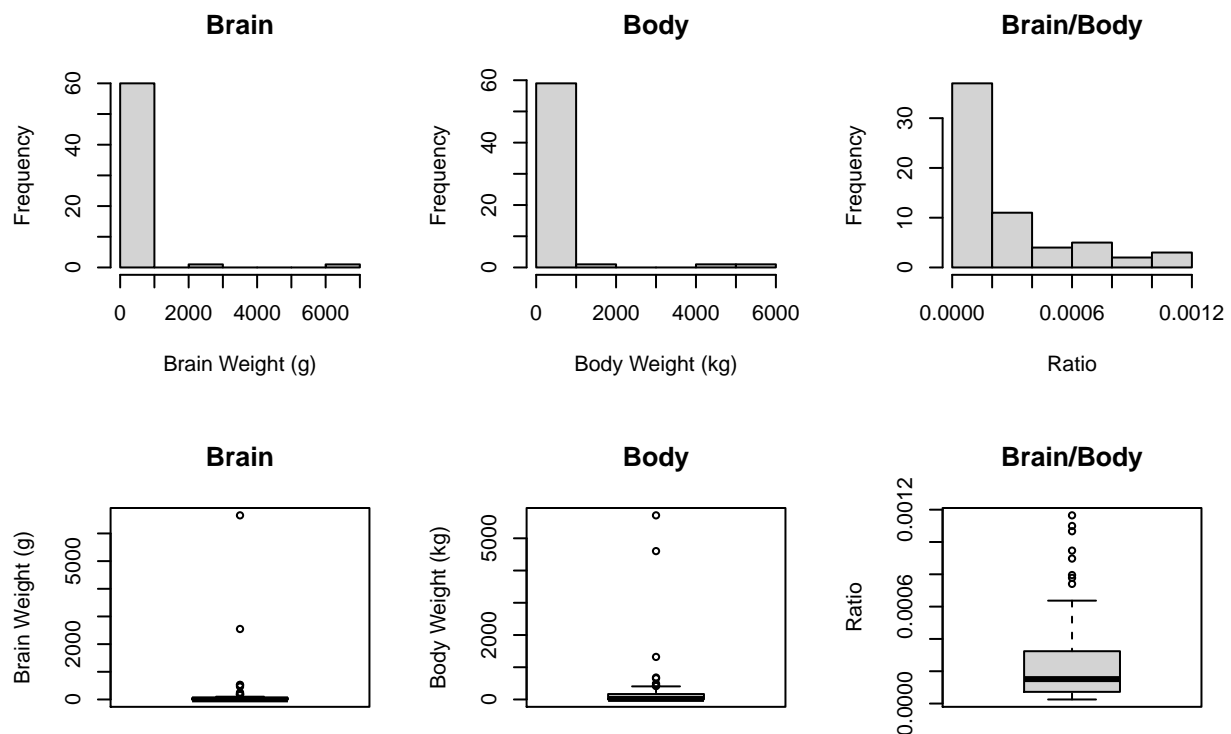
I used subset function in tidyverse to get the rows without NA value and we can get cleaned data! It is same as the result in old school way.

## Data Description

```
## Brain Body data : summary
knitr::kable(summary(Brain_new))
```

Brain weight (g)	Body weight (kg)
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

```
## Brain Body data : histogram and boxplot
par(mfrow=c(2,3))
hist(Brain_new[,1],main="Brain", xlab="Brain Weight (g)")
hist(Brain_new[,2],main="Body", xlab="Body Weight (kg)")
hist(0.001*Brain_new[,1]/Brain_new[,2],main="Brain/Body", xlab="Ratio")
boxplot(Brain_new[,1],main="Brain", ylab="Brain Weight (g)")
boxplot(Brain_new[,2],main="Body", ylab="Body Weight (kg)")
boxplot(0.001*Brain_new[,1]/Brain_new[,2],main="Brain/Body", ylab="Ratio")
```



I made summary, histogram, and boxplot for the data. Looks like there is an outlier.

```
## Outlier elimination
Brain_new_outlier_eliminated <- Brain_new %>%
  subset(Brain_new[,1]<300 & Brain_new[,2]<600)
```

Eliminated observations that have brains weight more or equal to 300 g or bodies weight more or equal to 600 kg.

```
## Brain Body data : histogram and boxplot after outlier elimination
```

```
par(mfrow=c(2,3))
```

```
hist(Brain_new_outlier_eliminated[,1],main="Brain", xlab="Brain Weight (g)")
```

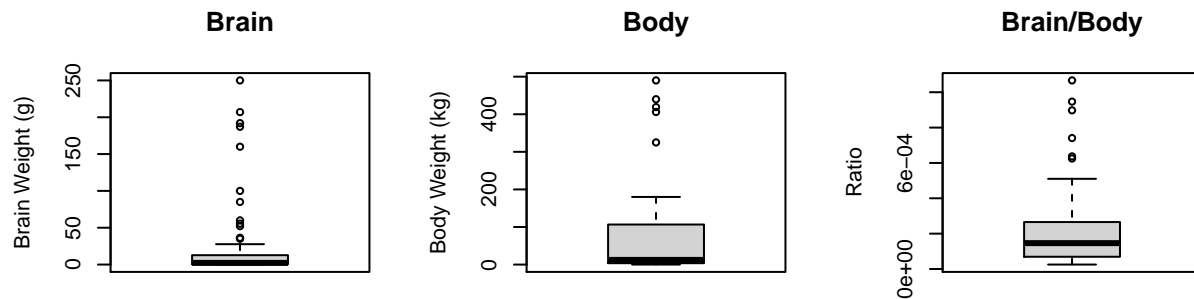
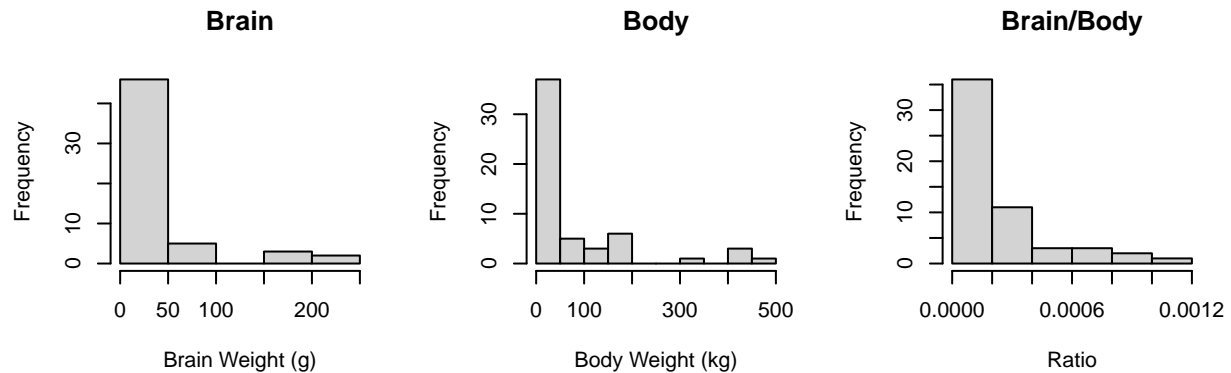
```
hist(Brain_new_outlier_eliminated[,2],main="Body", xlab="Body Weight (kg)")
```

```
hist(0.001*Brain_new_outlier_eliminated[,1]/Brain_new_outlier_eliminated[,2],main="Brain/Body", xlab="R
```

```
boxplot(Brain_new_outlier_eliminated[,1],main="Brain", ylab="Brain Weight (g)")
```

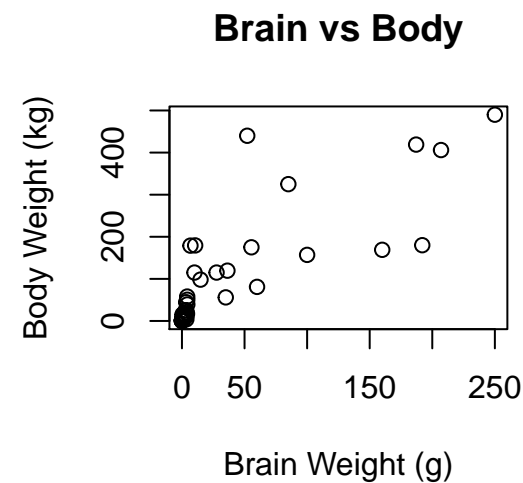
```
boxplot(Brain_new_outlier_eliminated[,2],main="Body", ylab="Body Weight (kg)")
```

```
boxplot(0.001*Brain_new_outlier_eliminated[,1]/Brain_new_outlier_eliminated[,2],main="Brain/Body", ylab="R
```



```
## Brain Body data : brain vs body plot after outlier elimination
```

```
plot(Brain_new_outlier_eliminated[,1],Brain_new_outlier_eliminated[,2],main="Brain vs Body", xlab="Brain
```



I made histogram, boxplot, and brain vs body plot after outlier elimination. The histogram and boxplot look better than before and looks like there is a positive correlation between brain weight and body weight. Special issue was not detected in the data.

## (d) Tomato Data

```
## Tomato import
url_Tomato <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"
Tomato_raw <- fread(url_Tomato)

## Warning in fread(url_Tomato): Detected 3 column names but the data has 4 columns
## (i.e. invalid file). Added 1 extra default column name for the first column
## which is guessed to be row names or an index. Use setnames() afterwards if this
## guess is not correct, or fix the file write command that created the file to
## create a valid file.

knitr::kable(Tomato_raw)
```

V1	10000	20000	30000
Ife#1	16.1,15.3,17.5	16.6,19.2,18.5	20.8,18.0,21.0
PusaEarlyDwarf	8.1,8.6,10.1,	12.7,13.7,11.5	14.4,15.4,13.7

I imported data from the url. It looks like data is mess. It looks similar to kind of contingency table, i.e., yield results are in combination of density (10000, 20000, 30000) and Tomato kind (Ife#1, PusaEarlyDwarf). We should transform this data into data with 3 columns (Tomato Kind, Density, Yield) and 18 observation.

### Using Basic R Functions

```
## Tomato data : old school : Transpose the data
Tomato_old_school <- Tomato_raw

Tomato_old_school <- as.data.frame(t(Tomato_old_school))

colnames(Tomato_old_school) <- Tomato_old_school[1,]
Tomato_old_school <- Tomato_old_school[-1,]
Tomato_old_school[,3] <- rownames(Tomato_old_school)
rownames(Tomato_old_school) <- 1:3

head(Tomato_old_school)
```

```
##           Ife\\#1 PusaEarlyDwarf      V3
## 1 16.1,15.3,17.5  8.1,8.6,10.1, 10000
## 2 16.6,19.2,18.5 12.7,13.7,11.5 20000
## 3 20.8,18.0,21.0 14.4,15.4,13.7 30000
```

First, for convenience, I transposed the data.

```
## Tomato data : old school : Use strsplit and unlist function to separate yield values
Tomato_old_school <- data.frame(
  c(rep(colnames(Tomato_old_school)[1],9),rep(colnames(Tomato_old_school)[2],9)),
  rep(c(rep(10000,3),rep(20000,3),rep(30000,3)),2),
  as.numeric(c(unlist(strsplit(Tomato_old_school[1,1],","), unlist(strsplit(Tomato_old_school[2,1],","),
    unlist(strsplit(Tomato_old_school[1,2],","), unlist(strsplit(Tomato_old_school[2,2],","), unlist(
  )

colnames(Tomato_old_school) <- c("kind", "density", "yield")
knitr::kable(Tomato_old_school)
```

kind	density	yield
Ife#1	10000	16.1
Ife#1	10000	15.3
Ife#1	10000	17.5
Ife#1	20000	16.6
Ife#1	20000	19.2
Ife#1	20000	18.5
Ife#1	30000	20.8
Ife#1	30000	18.0
Ife#1	30000	21.0
PusaEarlyDwarf	10000	8.1
PusaEarlyDwarf	10000	8.6
PusaEarlyDwarf	10000	10.1
PusaEarlyDwarf	20000	12.7
PusaEarlyDwarf	20000	13.7
PusaEarlyDwarf	20000	11.5
PusaEarlyDwarf	30000	14.4
PusaEarlyDwarf	30000	15.4
PusaEarlyDwarf	30000	13.7

I used strsplit and unlist function to split yield values in the same data point, and made data frame with these yield values, Tomato kinds, and density values.

## Using Tidyverse

```
## Tomato data : tidyverse : Transpose the data
Tomato_new <- Tomato_raw

Tomato_new <- as.data.frame(t(Tomato_new))

colnames(Tomato_new) <- Tomato_new[1,]
Tomato_new <- Tomato_new[-1,]
Tomato_new[,3] <- rownames(Tomato_new)
rownames(Tomato_new) <- 1:3

head(Tomato_new)
```

```
##           Ife\\#1 PusaEarlyDwarf    V3
## 1 16.1,15.3,17.5  8.1,8.6,10.1, 10000
## 2 16.6,19.2,18.5 12.7,13.7,11.5 20000
## 3 20.8,18.0,21.0 14.4,15.4,13.7 30000
```

First, for convenience, I transposed the data.

```
## Tomato data : tidyverse : use gather and separate_rows function
Tomato_new <- gather(data=Tomato_new, key="kind", value="yield", `Ife\\#1`, `PusaEarlyDwarf`)
Tomato_new <- Tomato_new %>%
  separate_rows(yield, sep=",")
Tomato_new <- Tomato_new[-13,]
Tomato_new <- Tomato_new %>%
  arrange(kind,V3)

Tomato_new <- Tomato_new[,c(2,1,3)]
```

```
colnames(Tomato_new)[2] <- "density"

Tomato_new$yield=as.numeric(Tomato_new$yield)
knitr::kable(Tomato_new)
```

kind	density	yield
Ife#1	10000	16.1
Ife#1	10000	15.3
Ife#1	10000	17.5
Ife#1	20000	16.6
Ife#1	20000	19.2
Ife#1	20000	18.5
Ife#1	30000	20.8
Ife#1	30000	18.0
Ife#1	30000	21.0
PusaEarlyDwarf	10000	8.1
PusaEarlyDwarf	10000	8.6
PusaEarlyDwarf	10000	10.1
PusaEarlyDwarf	20000	12.7
PusaEarlyDwarf	20000	13.7
PusaEarlyDwarf	20000	11.5
PusaEarlyDwarf	30000	14.4
PusaEarlyDwarf	30000	15.4
PusaEarlyDwarf	30000	13.7

I used gather function in tidyverse to make kind column. I used separate\_rows function in tidyverse to split yield values in the same density, and arranged the data by kind and density. The result is as same as the previous result.

## Data Description

```
## Tomato data : Summary for yield in Ife\\#1 kind with density 10000
knitr::kable(summary(Tomato_new %>%
  subset(kind == "Ife\\#1"& density == 10000) %>%
  select(yield))
)
```

yield
Min. :15.3
1st Qu.:15.7
Median :16.1
Mean :16.3
3rd Qu.:16.8
Max. :17.5

```
## Tomato data : Summary for yield in Ife\\#1 kind with density 20000
knitr::kable(summary(Tomato_new %>%
  subset(kind == "Ife\\#1"& density == 20000) %>%
  select(yield))
)
```

)

yield
Min. :16.60
1st Qu.:17.55
Median :18.50
Mean :18.10
3rd Qu.:18.85
Max. :19.20

```
## Tomato data : Summary for yield in Ife\\#1 kind with density 30000
knitr::kable(summary(Tomato_new %>%
  subset(kind == "Ife\\#1"& density == 30000) %>%
  select(yield))
)
```

yield
Min. :18.00
1st Qu.:19.40
Median :20.80
Mean :19.93
3rd Qu.:20.90
Max. :21.00

```
## Tomato data : Summary for yield in PusaEarlyDwarf kind with density 10000
knitr::kable(summary(Tomato_new %>%
  subset(kind == "PusaEarlyDwarf"& density == 10000) %>%
  select(yield))
)
```

yield
Min. : 8.100
1st Qu.: 8.350
Median : 8.600
Mean : 8.933
3rd Qu.: 9.350
Max. :10.100

```
## Tomato data : Summary for yield in PusaEarlyDwarf kind with density 20000
knitr::kable(summary(Tomato_new %>%
  subset(kind == "PusaEarlyDwarf"& density == 20000) %>%
  select(yield))
)
```

yield
Min. :11.50
1st Qu.:12.10
Median :12.70
Mean :12.63

yield
3rd Qu.:13.20
Max. :13.70

```
## Tomato data : Summary for yield in PusaEarlyDwarf kind with density 30000
knitr::kable(summary(Tomato_new %>%
  subset(kind == "PusaEarlyDwarf"& density == 30000) %>%
  select(yield))
)
```

yield
Min. :13.70
1st Qu.:14.05
Median :14.40
Mean :14.50
3rd Qu.:14.90
Max. :15.40

```
## Tomato data : boxplot
par(mfrow=c(2,3))
boxplot(Tomato_new %>%
  subset(kind == "Ife\\#1"& density == 10000) %>%
  select(yield), ylab="yield", main=c("yield for Ife\\#1", "in density 10000")
)

boxplot(Tomato_new %>%
  subset(kind == "Ife\\#1"& density == 20000) %>%
  select(yield), ylab="yield", main=c("yield for Ife\\#1", "in density 20000")
)

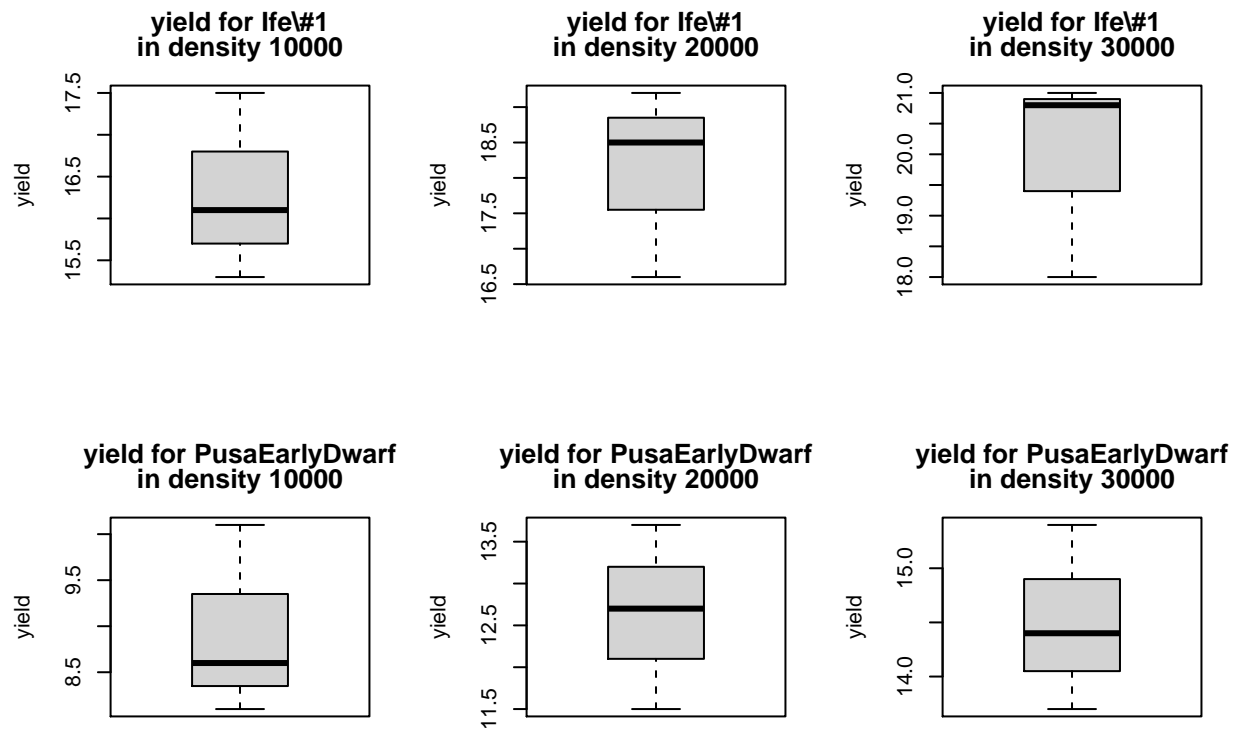
boxplot(Tomato_new %>%
  subset(kind == "Ife\\#1"& density == 30000) %>%
  select(yield), ylab="yield", main=c("yield for Ife\\#1", "in density 30000")
)

boxplot(Tomato_new %>%
  subset(kind == "PusaEarlyDwarf"& density == 10000) %>%
  select(yield), ylab="yield", main=c("yield for PusaEarlyDwarf", "in density 10000")
)

boxplot(Tomato_new %>%
  subset(kind == "PusaEarlyDwarf"& density == 20000) %>%
  select(yield), ylab="yield", main=c("yield for PusaEarlyDwarf", "in density 20000")
)

boxplot(Tomato_new %>%
  subset(kind == "PusaEarlyDwarf"& density == 30000) %>%
  select(yield), ylab="yield", main=c("yield for PusaEarlyDwarf", "in density 30000")
)
```





I summarized the Tomato data and plotted histogram and boxplot. Special issue was not detected in the data.