HW2_jaeyounglee

Jaeyoung Lee

September 15, 2020

Problem 3

First of all, it is to handle a mistake. Also, one can handle various versions of a code and see the history of a code. Furthermore, using version control, it is easy to collaborate with others. This is because one can share a code and work on the cloud such as GitHub.

Problem 4

a. Sensory data from five operators. http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat

```r
######## Sensory data ########
# Getting "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
url_sensory <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
sensory_rawdata <- fread(url_sensory, fill = TRUE, skip = 2, data.table = FALSE)
saveRDS(sensory_rawdata, 'sensory_rawdata.RDS')
sensory_rawdata <- readRDS('sensory_rawdata.RDS')
```

There are missing values in the raw data and the categories "Items" are in the data like oberservations. We need to remove missing values and extract the 'Item' numbers from the data.

```r
# Using base R function only
# Convert data.frame to matrix and transpose the raw data
matrix_sensory <- t(as.matrix(sensory_rawdata))

# Find where the missing values are
na <- which(is.na(matrix_sensory==TRUE))

# The indexes where Item numbers are in the data
x <- 1
item <- x
for (i in 1:9){
  x <- x+18
  item <- c(item, x)
}

# Remove missing values and 'Item's from the data
sensory_data <- t(matrix(matrix_sensory[-c(na,item)], byrow = T, nrow = 10))
sensory_data <- data.table(sensory_data)
colnames(sensory_data) <- paste('Item', 1:10) # Assign column names
Opr <- rep(paste('Opr', 1:5), 3)              # Operator names
sensory_data <- cbind(Opr,sensory_data)       # Bind Operator names and the data
```

```r
# Re-order the rows by names of operators
sensory_data <- sensory_data[order(sensory_data$Opr)]
sensory_data_base <- sensory_data

# Final tidy data with base R functions
head(sensory_data_base)
```

```
##      Opr Item 1 Item 2 Item 3 Item 4 Item 5 Item 6 Item 7 Item 8 Item 9 Item 10
## 1: Opr 1    4.3    6.0    2.4    7.4    5.7    2.2    1.2    4.2    8.0     5.0
## 2: Opr 1    4.3    4.9    3.9    7.1    5.8    3.0    1.3    3.0    9.0     5.4
## 3: Opr 1    4.1    6.0    1.9    6.4    5.8    2.1    0.9    4.8    8.9     2.8
## 4: Opr 2    4.9    5.3    2.5    8.2    6.3    2.4    1.5    4.8    8.6     4.8
## 5: Opr 2    4.5    6.3    3.0    7.9    5.7    1.8    2.4    4.5    7.7     5.0
## 6: Opr 2    5.3    5.9    3.9    7.1    6.0    3.3    3.1    4.8    9.2     5.2
```

Above is the converted tidy data frames using the base R functions only. A summary of the data is as follows:

Opr	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9	Item 10
Length:15	Min. :3.300	Min. :4.200	Min. :1.300	Min. :5.90	Min. :4.90	Min. :1.100	Min. :0.700	Min. :3.000	Min. :6.700	Min. :2.80
Class :character	1st Qu.:4.050	1st Qu.:4.700	1st Qu.:2.350	1st Qu.:6.40	1st Qu.:5.70	1st Qu.:1.750	1st Qu.:1.000	1st Qu.:4.400	1st Qu.:7.950	1st Qu.:3.90
Mode :character	Median :4.400	Median :5.300	Median :2.600	Median :6.90	Median :5.90	Median :2.100	Median :1.200	Median :4.600	Median :8.800	Median :4.80
NA	Mean :4.467	Mean :5.313	Mean :2.773	Mean :6.88	Mean :5.92	Mean :2.393	Mean :1.407	Mean :4.427	Mean :8.467	Mean :4.52
NA	3rd Qu.:5.100	3rd Qu.:5.950	3rd Qu.:3.050	3rd Qu.:7.20	3rd Qu.:6.15	3rd Qu.:3.150	3rd Qu.:1.550	3rd Qu.:4.800	3rd Qu.:9.000	3rd Qu.:5.10
NA	Max. :5.700	Max. :6.300	Max. :4.600	Max. :8.20	Max. :7.00	Max. :4.000	Max. :3.100	Max. :4.900	Max. :9.400	Max. :5.50

Now, handle the same data with `tidyverse` package.

```r
# Sensory data with tidyverse package
# Making matrix which is the same with base R function but using pipes.
matrix_sensory <- sensory_rawdata %>% as.matrix() %>% t()
na <- which(is.na(matrix_sensory==TRUE))      # Find missing values

# The indexes where Item numbers are in the data
x <- 1
item <- x
for (i in 1:9){
  x <- x+18
  item <- c(item, x)
}

# Remove missing values and Item numbers from the data
sensory_data <- matrix_sensory[-c(na,item)] %>% matrix(byrow = T, nrow = 10) %>% t()
sensory_data <- data.table(sensory_data)
Opr <- rep(paste('Opr', 1:5), 3)
sensory_data <- bind_cols(Opr,sensory_data) # bind operators and data
```

```
## New names:
## * NA -> ...1
```

```r
colnames(sensory_data) <- c('Opr',paste('Item', 1:10))

# Re-order the rows by names of operators
sensory_data <- sensory_data[order(sensory_data$Opr)]
sensory_data_tidyverse <- sensory_data

# Final tidy data with tidyverse
sensory_data_tidyverse %>% head()
```
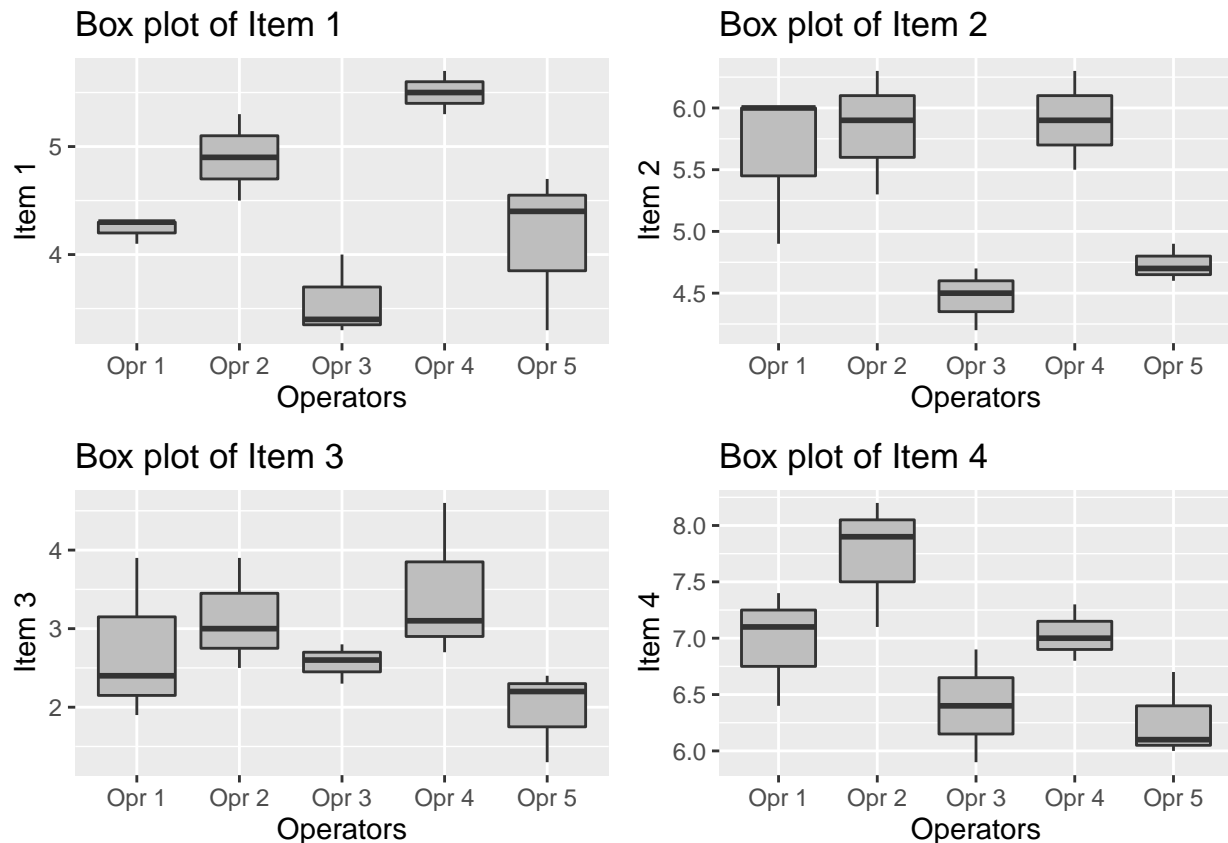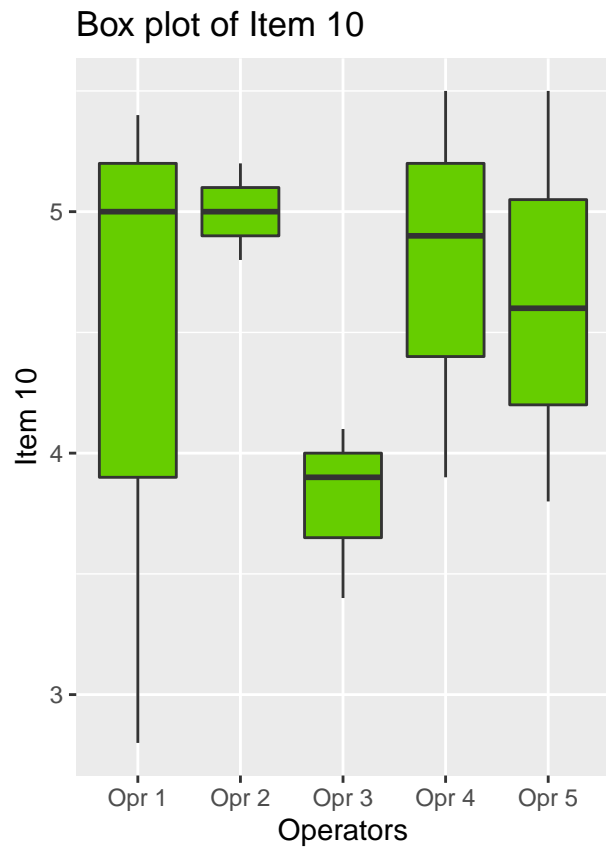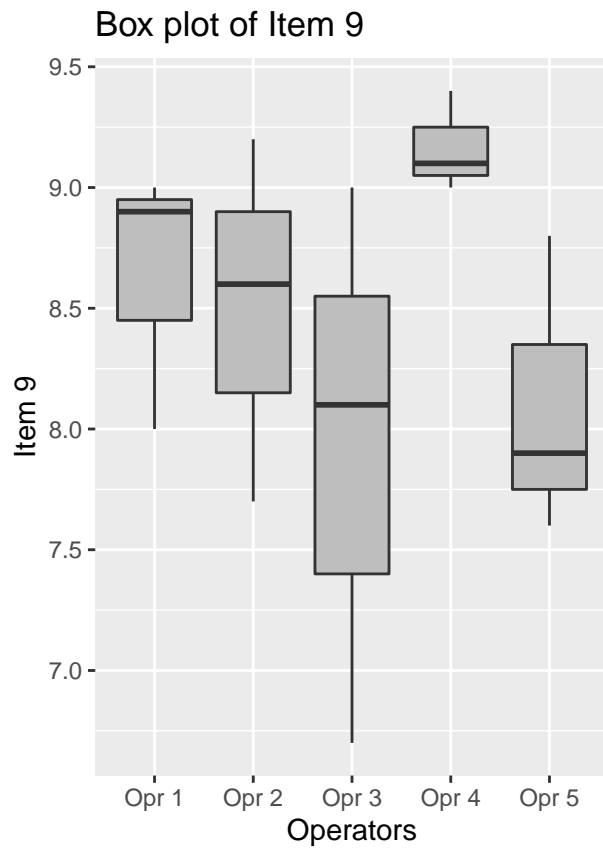
```
##        Opr Item 1 Item 2 Item 3 Item 4 Item 5 Item 6 Item 7 Item 8 Item 9 Item 10
## 1: Opr 1    4.3    6.0    2.4    7.4    5.7    2.2    1.2    4.2    8.0    5.0
## 2: Opr 1    4.3    4.9    3.9    7.1    5.8    3.0    1.3    3.0    9.0    5.4
## 3: Opr 1    4.1    6.0    1.9    6.4    5.8    2.1    0.9    4.8    8.9    2.8
## 4: Opr 2    4.9    5.3    2.5    8.2    6.3    2.4    1.5    4.8    8.6    4.8
## 5: Opr 2    4.5    6.3    3.0    7.9    5.7    1.8    2.4    4.5    7.7    5.0
## 6: Opr 2    5.3    5.9    3.9    7.1    6.0    3.3    3.1    4.8    9.2    5.2
```
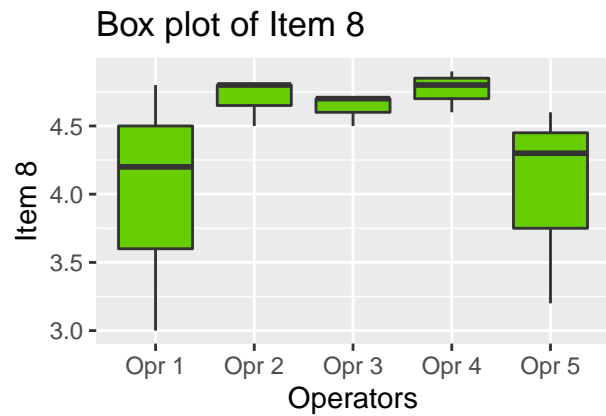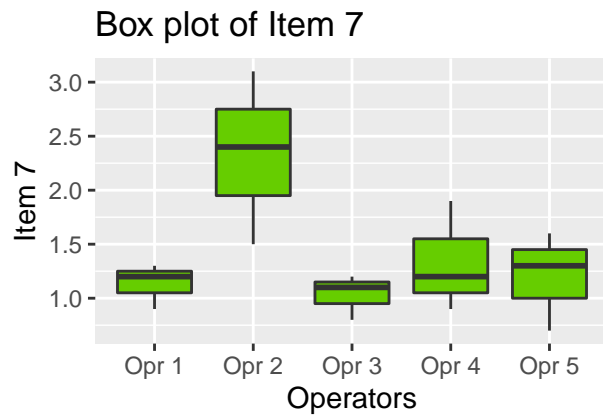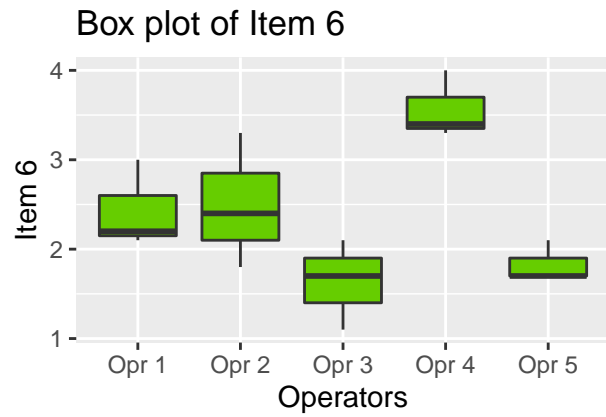
The result by tidyverse is the same with the base R function. The summary of the data converted by tidyverse is as follows.

Opr	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9	Item 10
Length:15	Min. :3.300	Min. :4.200	Min. :1.300	Min. :5.90	Min. :4.90	Min. :1.100	Min. :0.700	Min. :3.000	Min. :6.700	Min. :2.80
Class :character	1st Qu.:4.050	1st Qu.:4.700	1st Qu.:2.350	1st Qu.:6.40	1st Qu.:5.70	1st Qu.:1.750	1st Qu.:1.000	1st Qu.:4.400	1st Qu.:7.950	1st Qu.:3.90
Mode :character	Median :4.400	Median :5.300	Median :2.600	Median :6.90	Median :5.90	Median :2.100	Median :1.200	Median :4.600	Median :8.800	Median :4.80
NA	Mean :4.467	Mean :5.313	Mean :2.773	Mean :6.88	Mean :5.92	Mean :2.393	Mean :1.407	Mean :4.427	Mean :8.467	Mean :4.52
NA	3rd Qu.:5.100	3rd Qu.:5.950	3rd Qu.:3.050	3rd Qu.:7.20	3rd Qu.:6.15	3rd Qu.:3.150	3rd Qu.:1.550	3rd Qu.:4.800	3rd Qu.:9.000	3rd Qu.:5.10
NA	Max. :5.700	Max. :6.300	Max. :4.600	Max. :8.20	Max. :7.00	Max. :4.000	Max. :3.100	Max. :4.900	Max. :9.400	Max. :5.50

Here are some box plots of the sensory data. Gray plots are based on base R functions and green ones are from tidyverse.

Box plot of Item 1

Box plot of Item 2

Box plot of Item 3

Box plot of Item 4

Box plot of Item 5

Box plot of Item 6

Box plot of Item 7

Box plot of Item 8

Box plot of Item 9

Box plot of Item 10

b. Gold Medal performance for Olympic Men's Long Jump, year is coded as 1900=0.
http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat

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

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

```
saveRDS(medal_rawdata, 'medal_rawdata.RDS')
medal_rawdata <- readRDS('medal_rawdata.RDS')
```

The raw data has missing values and wide type data. It is better to reshape the data. Also, we need two
vectors : 'Year' and 'Long Jump'.

```
# Using base R function only
# Year is coded as 1900 = 0
# Extract Year and Long Jump vectors
year <- c(medal_rawdata[[1]], medal_rawdata[[3]],
          medal_rawdata[[5]], medal_rawdata[[7]]) + 1900
longjump <- c(medal_rawdata[[2]], medal_rawdata[[4]],
              medal_rawdata[[6]], medal_rawdata[[8]])

# Bind the vectors as a data table and rename the categories
medal_data <- data.table(year[1:(length(year)-2)],
                         longjump[1:(length(longjump)-2)])
colnames(medal_data) <- c('Year', 'LongJump')
medal_data_base <- medal_data

# Final tidy data with base R functions
head(medal_data_base)
```

```
##      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
```

Above is the converted tidy data frames using the base R functions. A summary of the data is as follows:

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

Now, handle the same data with `tidyverse` package.

```r
# Using tidyverse package
# Year is coded as 1900 = 0
medal_data <- medal_rawdata[,1:8] # remove missing values only columns

# Extracting 'Year' columns and 'Long Jump' columns and remove missing values
colnames(medal_data) <- paste(rep(c('Year', 'Jump'),4), rep(1:4,each = 2))
year <- medal_data[,c(1,3,5,7)] %>%
  gather(key = 'name1', value = 'Year', 1,2,3,4) %>% filter(Year != na)
```

```
## Warning in Year != na: longer object length is not a multiple of shorter object
## length
```

```r
year[,2] <- year[,2] + 1900
jump <- medal_data[,c(2,4,6,8)] %>%
  gather(key = 'name2', value = 'LongJump', 1,2,3,4) %>% filter(LongJump != na)
```

```
## Warning in LongJump != na: longer object length is not a multiple of shorter
## object length
```

```r
# Bind the vectors as a data table and rename the categories
medal_data <- bind_cols(year[,2], jump[,2])
```

```
## New names:
## * NA -> ...1
## * NA -> ...2
```

```r
colnames(medal_data) <- c('Year', 'LongJump')
medal_data_tidyverse <- medal_data

# Final tidy data with tidyverse
medal_data_tidyverse %>% head()
```

```
## # A tibble: 6 x 2
##     Year LongJump
##    <dbl>    <dbl>
## 1   1896     250.
## 2   1900     283.
## 3   1904     289
## 4   1908     294.
## 5   1912     299.
## 6   1920     282.
```
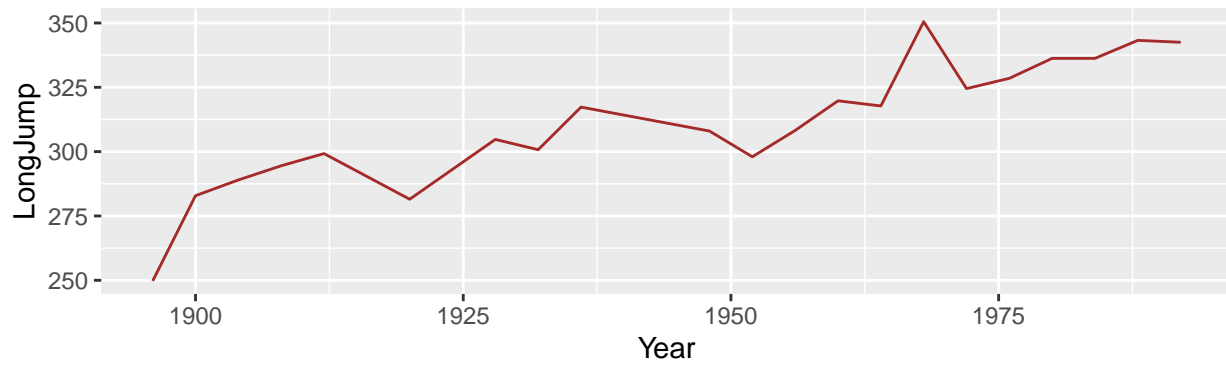
The result by tidyverse is the same with the base R function. The summary of the data converted by tidyverse is as follows.
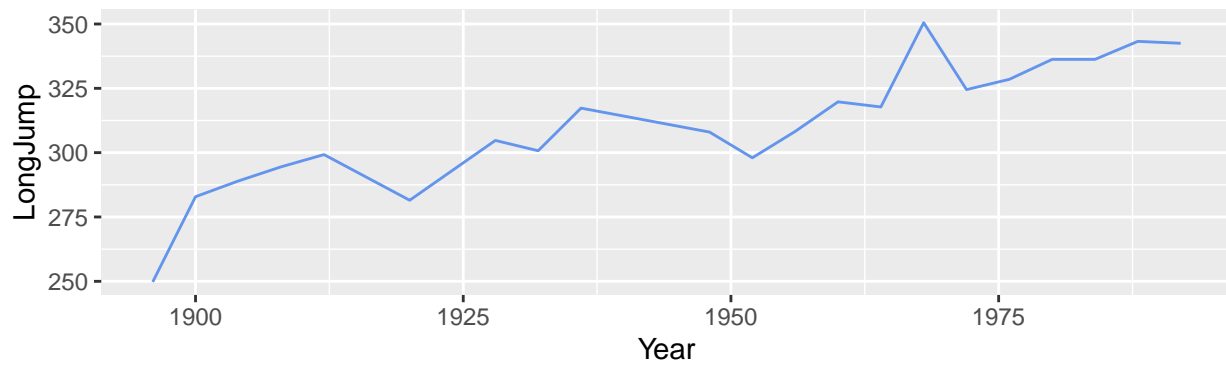
Year	LongJump
Min. :1896	Min. :249.8
1st Qu.:1921	1st Qu.:295.4
Median :1950	Median :308.1
Mean :1945	Mean :310.3
3rd Qu.:1971	3rd Qu.:327.5
Max. :1992	Max. :350.5

As informative plots, line plots are used.

Plot of Long Jump by Years with Base R

Plot of Long Jump by Years with tidyverse

c. Brain weight (g) and body weight (kg) for 62 species.
http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat

```
######## Brain weight data ########
# Getting "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
url_brain <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
brain_rawdata <- 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.
```

```
saveRDS(brain_rawdata, 'brain_rawdata.RDS')
brain_rawdata <- readRDS('brain_rawdata.RDS')
```

The data needs two columns which are 'Body Wt' and 'Brain Wt'.

```
# Using base R function only
# The method is the same with the data from part (b)
# Extract Body Wt and Brain Wt vectors
bodywt <- c(brain_rawdata[[1]], brain_rawdata[[3]], brain_rawdata[[5]])
brainwt <- c(brain_rawdata[[2]], brain_rawdata[[4]], brain_rawdata[[6]])

# Remove missing values
brain_data <- data.table(bodywt[-length(bodywt)], brainwt[-length(brainwt)])
colnames(brain_data) <- c('Body_Wt', 'Brain_Wt')
brain_data_base <- brain_data

# Final tidy data with base R functions
head(brain_data_base)
```

```
##     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
```

Above is the converted tidy data frames using the base R functions. A summary of the data is as follows:

Body_Wt	Brain_Wt
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

Now, handle the same data with `tidyverse` package.

```r
# Tidy data with tidyverse
# Remove vectors which have only missing values
brain_data <- brain_rawdata[,1:6]

# Extracting 'Year' columns and 'Long Jump' columns and remove missing values
colnames(brain_data) <- paste(rep(c('bw', 'brw'),3), rep(1:3,each = 2))
bw <- brain_data[,c(1,3,5)] %>% gather(key = 'name1', value = 'BW', 1,2,3) %>%
  filter(BW != na)
```

```
## Warning in BW != na: longer object length is not a multiple of shorter object
## length
```

```r
brw <- brain_data[,c(2,4,6)] %>% gather(key = 'name2', value = 'BRW', 1,2,3) %>%
  filter(BRW != na)
```

```
## Warning in BRW != na: longer object length is not a multiple of shorter object
## length
```

```r
# Bind the vectors as a data table and rename the categories
brain_data <- bind_cols(bw[,2], brw[,2])
```

```
## New names:
## * NA -> ...1
## * NA -> ...2
```

```r
colnames(brain_data) <- c('Body_Wt', 'Brain_Wt')
brain_data_tidyverse <- brain_data

# Final tidy data with tidyverse
brain_data_tidyverse %>% head()
```

```
## # A tibble: 6 x 2
##    Body_Wt Brain_Wt
##      <dbl>    <dbl>
## 1     3.38     44.5
## 2     0.48     15.5
## 3     1.35      8.1
## 4   465      423
## 5    36.3     120.
## 6    27.7     115
```
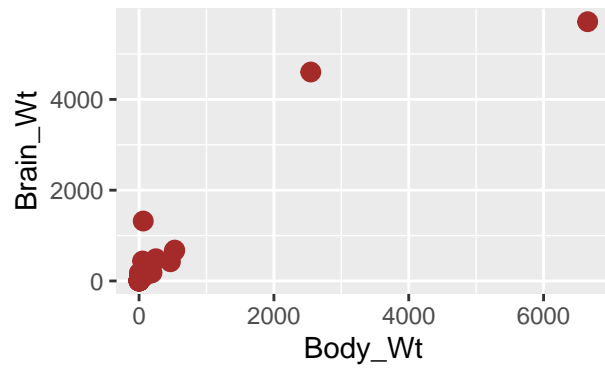
The result by tidyverse is the same with the base R function. The summary of the data converted by tidyverse is as follows.
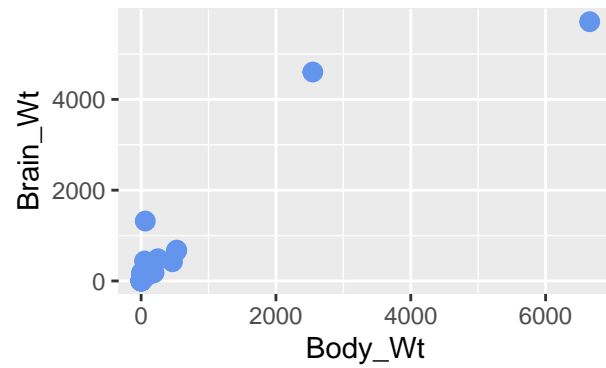
Body_Wt	Brain_Wt
Min. : 0.005	Min. : 0.10
1st Qu.: 0.600	1st Qu.: 4.25
Median : 3.342	Median : 17.25
Mean : 198.790	Mean : 283.13
3rd Qu.: 48.202	3rd Qu.: 166.00
Max. :6654.000	Max. :5712.00

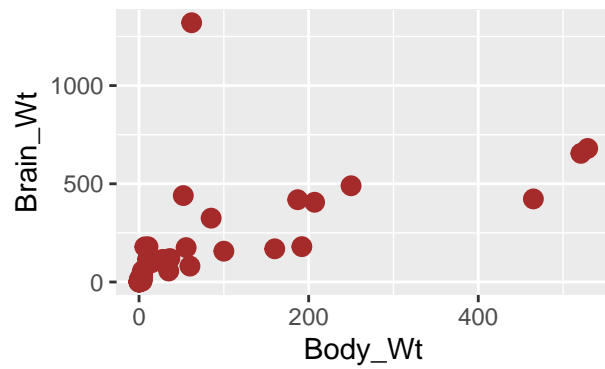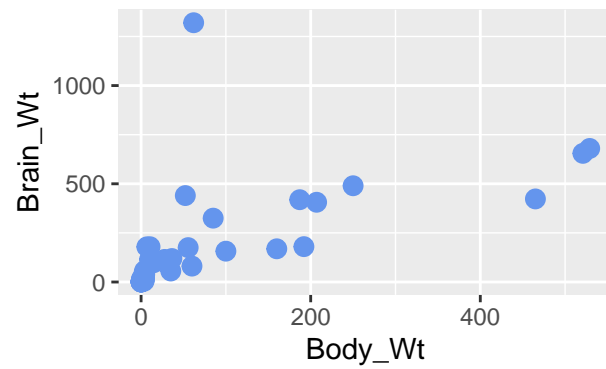As informative plots, scatter plots are used.

d. Triplicate measurements of tomato yield for two varieties of tomatos at three planting densities.
http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat

```
######## Tomato data ########
# Getting "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"
url_tomato <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"
tomato_rawdata <- fread(url_tomato, skip = 1)
```

```
## Warning in fread(url_tomato, skip = 1): 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.
```

```
saveRDS(tomato_rawdata, 'tomato_rawdata.RDS')
tomato_rawdata <- readRDS('tomato_rawdata.RDS')
```

The values are grouped in the cells of the data above. Therefore, we should split the cells into single values.

```
# Using base R function only
# Need to split the values
cells <- strsplit(unlist(tomato_rawdata), split = ',', fixed = T) # split the data
values <- as.numeric(unlist(c(cells[3:8]))) # numerical data

# Combine the split values into data frame
tomato_matrix <- matrix(values, byrow = T, ncol = 3)
tomato_matrix <- t(cbind(tomato_matrix[1:2,], tomato_matrix[3:4,], tomato_matrix[5:6,]))

# Bind the data with the densities (categories)
tomato_data <- data.frame(tomato_matrix, as.character(rep(c(10000,20000,30000), each=3)))
colnames(tomato_data) <- c('IFE1', 'PusaEarlyDwarf','Density')
tomato_data_base <- tomato_data

# Final tidy data with base R functions
tomato_data_base
```

```
##   IFE1 PusaEarlyDwarf Density
## 1 16.1            8.1   10000
## 2 15.3            8.6   10000
## 3 17.5           10.1   10000
## 4 16.6           12.7   20000
## 5 19.2           13.7   20000
## 6 18.5           11.5   20000
## 7 20.8           14.4   30000
## 8 18.0           15.4   30000
## 9 21.0           13.7   30000
```

Above is the converted tidy data frames using the base R functions. A summary of the data is as follows:

IFE1	PusaEarlyDwarf	Density
Min. :15.30	Min. : 8.10	Length:9
1st Qu.:16.60	1st Qu.:10.10	Class :character
Median :18.00	Median :12.70	Mode :character
Mean :18.11	Mean :12.02	NA
3rd Qu.:19.20	3rd Qu.:13.70	NA
Max. :21.00	Max. :15.40	NA

Now, handle the same data with `tidyverse` package.

```r
# Using tidyverse package
# Need to split the values
tomato_data <- tomato_rawdata[,-1] %>%
  separate(col = '10000', into = c("1","2","3"), sep = ",", convert = T) %>%
  separate(col = '20000', into = c("4","5","6"), sep = ",", convert = T) %>%
  separate(col = '30000', into = c("7","8","9"), sep = ",", convert = T) %>%
  as.matrix() %>% t()
```

```
## Warning: Expected 3 pieces. Additional pieces discarded in 1 rows [2].
```

```r
dens <- rep(c(10000,20000,30000), each = 3) %>% as.character() # Densities

# Bind the data with the densities vector
tomato_data <- tomato_data %>% data.frame(dens)
colnames(tomato_data) <- c('IFE1', 'PusaEarlyDwarf','Density')
tomato_data_tidyverse <- tomato_data

# Final tidy data with tidyverse
tomato_data_tidyverse
```

```
##   IFE1 PusaEarlyDwarf Density
## 1 16.1            8.1   10000
## 2 15.3            8.6   10000
## 3 17.5           10.1   10000
## 4 16.6           12.7   20000
## 5 19.2           13.7   20000
## 6 18.5           11.5   20000
## 7 20.8           14.4   30000
## 8 18.0           15.4   30000
## 9 21.0           13.7   30000
```

IFE1	PusaEarlyDwarf	Density
Min. :15.30	Min. : 8.10	Length:9
1st Qu.:16.60	1st Qu.:10.10	Class :character
Median :18.00	Median :12.70	Mode :character
Mean :18.11	Mean :12.02	NA
3rd Qu.:19.20	3rd Qu.:13.70	NA
Max. :21.00	Max. :15.40	NA

13

Here are the histograms of above data based on both base R and tidyverse.
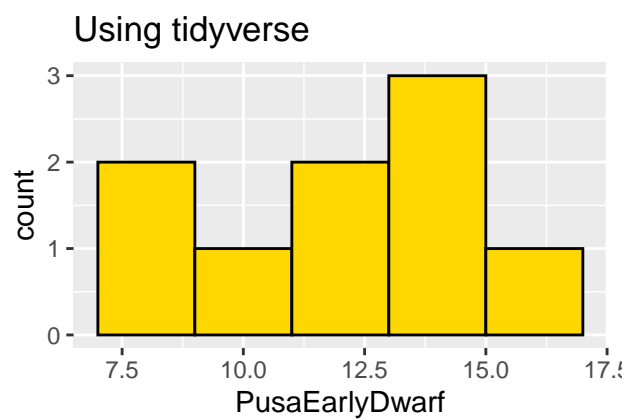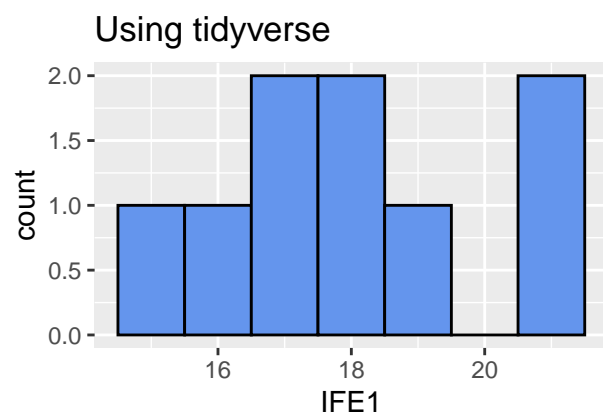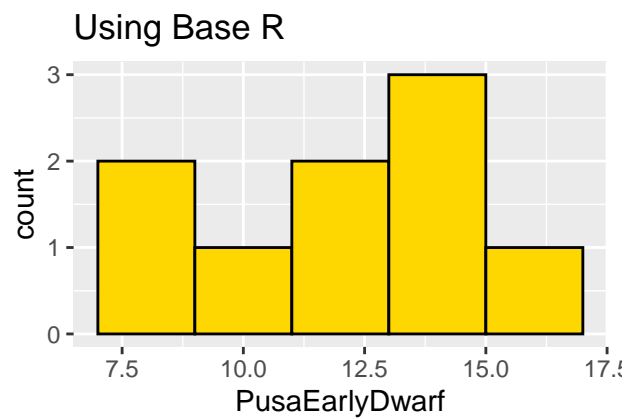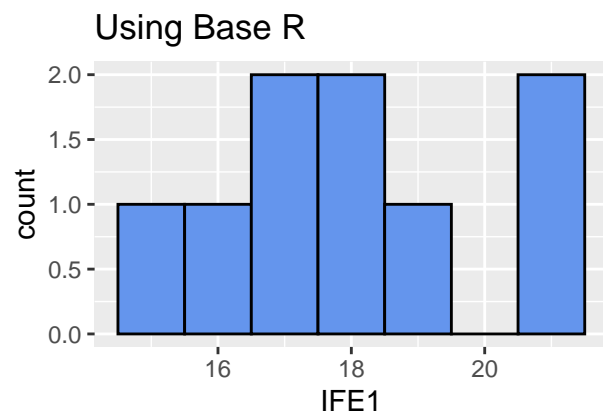
Appendix

```r
knitr::opts_chunk$set(echo = TRUE)
library(data.table)
library(tidyverse)
library(ggplot2)
library(ggpubr)
######## Sensory data ########
# Getting "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
url_sensory <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
sensory_rawdata <- fread(url_sensory, fill = TRUE, skip = 2, data.table = FALSE)
saveRDS(sensory_rawdata, 'sensory_rawdata.RDS')
sensory_rawdata <- readRDS('sensory_rawdata.RDS')

# Using base R function only
# Convert data.frame to matrix and transpose the raw data
matrix_sensory <- t(as.matrix(sensory_rawdata))

# Find where the missing values are
na <- which(is.na(matrix_sensory==TRUE))

# The indexes where Item numbers are in the data
x <- 1
item <- x
for (i in 1:9){
  x <- x+18
  item <- c(item, x)
}

# Remove missing values and 'Item's from the data
sensory_data <- t(matrix(matrix_sensory[-c(na,item)], byrow = T, nrow = 10))
sensory_data <- data.table(sensory_data)
colnames(sensory_data) <- paste('Item', 1:10) # Assign column names
Opr <- rep(paste('Opr', 1:5), 3)              # Operator names
sensory_data <- cbind(Opr,sensory_data)       # Bind Operator names and the data

# Re-order the rows by names of operators
sensory_data <- sensory_data[order(sensory_data$Opr)]
sensory_data_base <- sensory_data

# Final tidy data with base R functions
head(sensory_data_base)

knitr::kable(summary(sensory_data_base))
# Sensory data with tidyverse package
# Making matrix which is the same with base R function but using pipes.
matrix_sensory <- sensory_rawdata %>% as.matrix() %>% t()
na <- which(is.na(matrix_sensory==TRUE))      # Find missing values

# The indexes where Item numbers are in the data
x <- 1
item <- x
for (i in 1:9){
```

```r
    x <- x+18
    item <- c(item, x)
}

# Remove missing values and Item numbers from the data
sensory_data <- matrix_sensory[-c(na,item)] %>% matrix(byrow = T, nrow = 10) %>% t()
sensory_data <- data.table(sensory_data)
Opr <- rep(paste('Opr', 1:5), 3)
sensory_data <- bind_cols(Opr,sensory_data) # bind operators and data
colnames(sensory_data) <- c('Opr',paste('Item', 1:10))

# Re-order the rows by names of operators
sensory_data <- sensory_data[order(sensory_data$Opr)]
sensory_data_tidyverse <- sensory_data

# Final tidy data with tidyverse
sensory_data_tidyverse %>% head()

knitr::kable(summary(sensory_data_tidyverse))
# box plots for item 1 to 4
base_item1 <- ggplot(sensory_data_base, aes(x=Opr, y= sensory_data_base[[2]])) +
  geom_boxplot(fill = 'gray') +
  labs(title="Box plot of Item 1",x="Operators", y = "Item 1")

base_item2 <- ggplot(sensory_data_base, aes(x=Opr, y= sensory_data_base[[3]])) +
  geom_boxplot(fill = 'gray') +
  labs(title="Box plot of Item 2",x="Operators", y = "Item 2")

base_item3 <- ggplot(sensory_data_base, aes(x=Opr, y= sensory_data_base[[4]])) +
  geom_boxplot(fill = 'gray') +
  labs(title="Box plot of Item 3",x="Operators", y = "Item 3")

base_item4 <- ggplot(sensory_data_base, aes(x=Opr, y= sensory_data_base[[5]])) +
  geom_boxplot(fill = 'gray') +
  labs(title="Box plot of Item 4",x="Operators", y = "Item 4")

ggarrange(base_item1, base_item2, base_item3, base_item4, ncol = 2, nrow = 2)

tidy_item5 <- ggplot(sensory_data_tidyverse, aes(x=Opr, y= sensory_data_base[[6]])) +
  geom_boxplot(fill = 'chartreuse3') +
  labs(title="Box plot of Item 5",x="Operators", y = "Item 5")

tidy_item6 <- ggplot(sensory_data_tidyverse, aes(x=Opr, y= sensory_data_base[[7]])) +
  geom_boxplot(fill = 'chartreuse3') +
  labs(title="Box plot of Item 6",x="Operators", y = "Item 6")

tidy_item7 <- ggplot(sensory_data_tidyverse, aes(x=Opr, y= sensory_data_base[[8]])) +
  geom_boxplot(fill = 'chartreuse3') +
  labs(title="Box plot of Item 7",x="Operators", y = "Item 7")

tidy_item8 <- ggplot(sensory_data_tidyverse, aes(x=Opr, y= sensory_data_base[[9]])) +
  geom_boxplot(fill = 'chartreuse3') +
  labs(title="Box plot of Item 8",x="Operators", y = "Item 8")
```

```r
ggarrange(tidy_item5, tidy_item6, tidy_item7, tidy_item8, ncol = 2, nrow = 2)
base_item9 <- ggplot(sensory_data_base, aes(x=Opr, y= sensory_data_base[[10]])) +
  geom_boxplot(fill = 'gray') +
  labs(title="Box plot of Item 9",x="Operators", y = "Item 9")

tidy_item10 <- ggplot(sensory_data_tidyverse, aes(x=Opr, y= sensory_data_base[[11]])) +
  geom_boxplot(fill = 'chartreuse3') +
  labs(title="Box plot of Item 10",x="Operators", y = "Item 10")

ggarrange(base_item9, tidy_item10, ncol = 2, nrow = 1)
######## Long Jump data ########
# Getting "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
url_medal <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
medal_rawdata <- fread(url_medal)
saveRDS(medal_rawdata, 'medal_rawdata.RDS')
medal_rawdata <- readRDS('medal_rawdata.RDS')

# Using base R function only
# Year is coded as 1900 = 0
# Extract Year and Long Jump vectors
year <- c(medal_rawdata[[1]], medal_rawdata[[3]],
          medal_rawdata[[5]], medal_rawdata[[7]]) + 1900
longjump <- c(medal_rawdata[[2]], medal_rawdata[[4]],
              medal_rawdata[[6]], medal_rawdata[[8]])

# Bind the vectors as a data table and rename the categories
medal_data <- data.table(year[1:(length(year)-2)],
                         longjump[1:(length(longjump)-2)])
colnames(medal_data) <- c('Year', 'LongJump')
medal_data_base <- medal_data

# Final tidy data with base R functions
head(medal_data_base)

knitr::kable(summary(medal_data_base))
# Using tidyverse package
# Year is coded as 1900 = 0
medal_data <- medal_rawdata[,1:8] # remove missing values only columns

# Extracting 'Year' columns and 'Long Jump' columns and remove missing values
colnames(medal_data) <- paste(rep(c('Year', 'Jump'),4), rep(1:4,each = 2))
year <- medal_data[,c(1,3,5,7)] %>%
  gather(key = 'name1', value = 'Year', 1,2,3,4) %>% filter(Year != na)
year[,2] <- year[,2] + 1900
jump <- medal_data[,c(2,4,6,8)] %>%
  gather(key = 'name2', value = 'LongJump', 1,2,3,4) %>% filter(LongJump != na)

# Bind the vectors as a data table and rename the categories
medal_data <- bind_cols(year[,2], jump[,2])
colnames(medal_data) <- c('Year', 'LongJump')
medal_data_tidyverse <- medal_data

# Final tidy data with tidyverse
```

```r
medal_data_tidyverse %>% head()


# Plot of Long Jump by Years
medal_base_plot <- ggplot(data = medal_data_base, aes(x = Year, y= LongJump)) +
  geom_line(col = 'brown') +
  labs(title = 'Plot of Long Jump by Years with Base R')

medal_tidyverse_plot <- ggplot(data = medal_data_tidyverse, aes(x = Year, y= LongJump)) +
  geom_line(col = 'cornflowerblue') +
  labs(title = 'Plot of Long Jump by Years with tidyverse')

ggarrange(medal_base_plot, medal_tidyverse_plot, ncol = 1, nrow = 2)

######## Brain weight data ########
# Getting "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
url_brain <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
brain_rawdata <- fread(url_brain)
saveRDS(brain_rawdata, 'brain_rawdata.RDS')
brain_rawdata <- readRDS('brain_rawdata.RDS')
# Using base R function only
# The method is the same with the data from part (b)
# Extract Body Wt and Brain Wt vectors
bodywt <- c(brain_rawdata[[1]], brain_rawdata[[3]], brain_rawdata[[5]])
brainwt <- c(brain_rawdata[[2]], brain_rawdata[[4]], brain_rawdata[[6]])

# Remove missing values
brain_data <- data.table(bodywt[-length(bodywt)], brainwt[-length(brainwt)])
colnames(brain_data) <- c('Body_Wt', 'Brain_Wt')
brain_data_base <- brain_data

# Final tidy data with base R functions
head(brain_data_base)

knitr::kable(summary(brain_data_base))
# Tidy data with tidyverse
# Remove vectors which have only missing values
brain_data <- brain_rawdata[,1:6]

# Extracting 'Year' columns and 'Long Jump' columns and remove missing values
colnames(brain_data) <- paste(rep(c('bw', 'brw'),3), rep(1:3,each = 2))
bw <- brain_data[,c(1,3,5)] %>% gather(key = 'name1', value = 'BW', 1,2,3) %>%
  filter(BW != na)
brw <- brain_data[,c(2,4,6)] %>% gather(key = 'name2', value = 'BRW', 1,2,3) %>%
  filter(BRW != na)

# Bind the vectors as a data table and rename the categories
brain_data <- bind_cols(bw[,2], brw[,2])
colnames(brain_data) <- c('Body_Wt', 'Brain_Wt')
brain_data_tidyverse <- brain_data

# Final tidy data with tidyverse
brain_data_tidyverse %>% head()
```

```r
# Scatter plots of "Body Wt" and "Brain Wt"
brain_base_plot <-
  ggplot(data = brain_data_base, aes(x = Body_Wt, y= Brain_Wt)) +
  geom_point(col = 'brown', size = 3, shape = 19) +
  labs(title = 'Using base')

brain_tidyverse_plot <-
  ggplot(data = brain_data_tidyverse, aes(x = Body_Wt, y= Brain_Wt)) +
  geom_point(col = 'cornflowerblue', size = 3, shape = 19) +
  labs(title = 'Using tidyverse')

# Scatter plots without outliers
brain_base_plot_wo_outlier <-
  ggplot(data = brain_data_base[-c(19,33),], aes(x = Body_Wt, y= Brain_Wt)) +
  geom_point(col = 'brown', size = 3, shape = 19) +
  labs(title = 'Base without outlier')

brain_tidyverse_plot_wo_outlier <-
  ggplot(data = brain_data_tidyverse[-c(19,33),], aes(x = Body_Wt, y= Brain_Wt)) +
  geom_point(col = 'cornflowerblue', size = 3, shape = 19) +
  labs(title = 'Tidyverse without outlier')



ggarrange(brain_base_plot, brain_tidyverse_plot,
          brain_base_plot_wo_outlier,
          brain_tidyverse_plot_wo_outlier,
          ncol = 2, nrow = 2)


######## Tomato data ########
# Getting "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"
url_tomato <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"
tomato_rawdata <- fread(url_tomato, skip = 1)
saveRDS(tomato_rawdata, 'tomato_rawdata.RDS')
tomato_rawdata <- readRDS('tomato_rawdata.RDS')

# Using base R function only
# Need to split the values
cells <- strsplit(unlist(tomato_rawdata), split = ',', fixed = T) # split the data
values <- as.numeric(unlist(c(cells[3:8]))) # numerical data

# Combine the split values into data frame
tomato_matrix <- matrix(values, byrow = T, ncol = 3)
tomato_matrix <- t(cbind(tomato_matrix[1:2,], tomato_matrix[3:4,], tomato_matrix[5:6,]))

# Bind the data with the densities (categories)
tomato_data <- data.frame(tomato_matrix, as.character(rep(c(10000,20000,30000), each=3)))
colnames(tomato_data) <- c('IFE1', 'PusaEarlyDwarf','Density')
tomato_data_base <- tomato_data

# Final tidy data with base R functions
tomato_data_base
```

```r
knitr::kable(summary(tomato_data_base))
# Using tidyverse package
# Need to split the values
tomato_data <- tomato_rawdata[,-1] %>%
  separate(col = '10000', into = c("1","2","3"), sep = ",", convert = T) %>%
  separate(col = '20000', into = c("4","5","6"), sep = ",", convert = T) %>%
  separate(col = '30000', into = c("7","8","9"), sep = ",", convert = T) %>%
  as.matrix() %>% t()
dens <- rep(c(10000,20000,30000), each = 3) %>% as.character() # Densities

# Bind the data with the densities vector
tomato_data <- tomato_data %>% data.frame(dens)
colnames(tomato_data) <- c('IFE1', 'PusaEarlyDwarf','Density')
tomato_data_tidyverse <- tomato_data

# Final tidy data with tidyverse
tomato_data_tidyverse

# Histograms of tomato data

hist_ife_base = ggplot(tomato_data_base, aes(x=IFE1)) +
  geom_histogram(binwidth = 1, fill=I("cornflowerblue"), col = I("black")) +
  labs(title = 'Using Base R')

hist_pusa_base = ggplot(tomato_data_base, aes(x=PusaEarlyDwarf)) +
  geom_histogram(binwidth = 2, fill=I("gold"), col = I("black")) +
  labs(title = 'Using Base R')

hist_ife_tidyverse = ggplot(tomato_data_tidyverse, aes(x=IFE1)) +
  geom_histogram(binwidth = 1, fill=I("cornflowerblue"), col = I("black")) +
  labs(title = 'Using tidyverse')

hist_pusa_tidyverse = ggplot(tomato_data_tidyverse, aes(x=PusaEarlyDwarf)) +
  geom_histogram(binwidth = 2, fill=I("gold"), col = I("black")) +
  labs(title = 'Using tidyverse')


ggarrange(hist_ife_base, hist_pusa_base,
          hist_ife_tidyverse, hist_pusa_tidyverse, ncol = 2, nrow = 2)
```