HW2_sliang

Liang Shi

9/9/2020

Problem 3

Personlly, as a researcher on data science, I tune the hyperparameter all the time, and it's had to ensure every tuning can improve the model. So, version control is important for me to recall to any previous conditions of my code.

Problem 4

Load package

```
library(data.table)
library('magrittr')
library('tidyverse')
```

```
## -- Attaching packages ----------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts -------------------------------------------- tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::set_names() masks magrittr::set_names()
## x purrr::transpose() masks data.table::transpose()
```

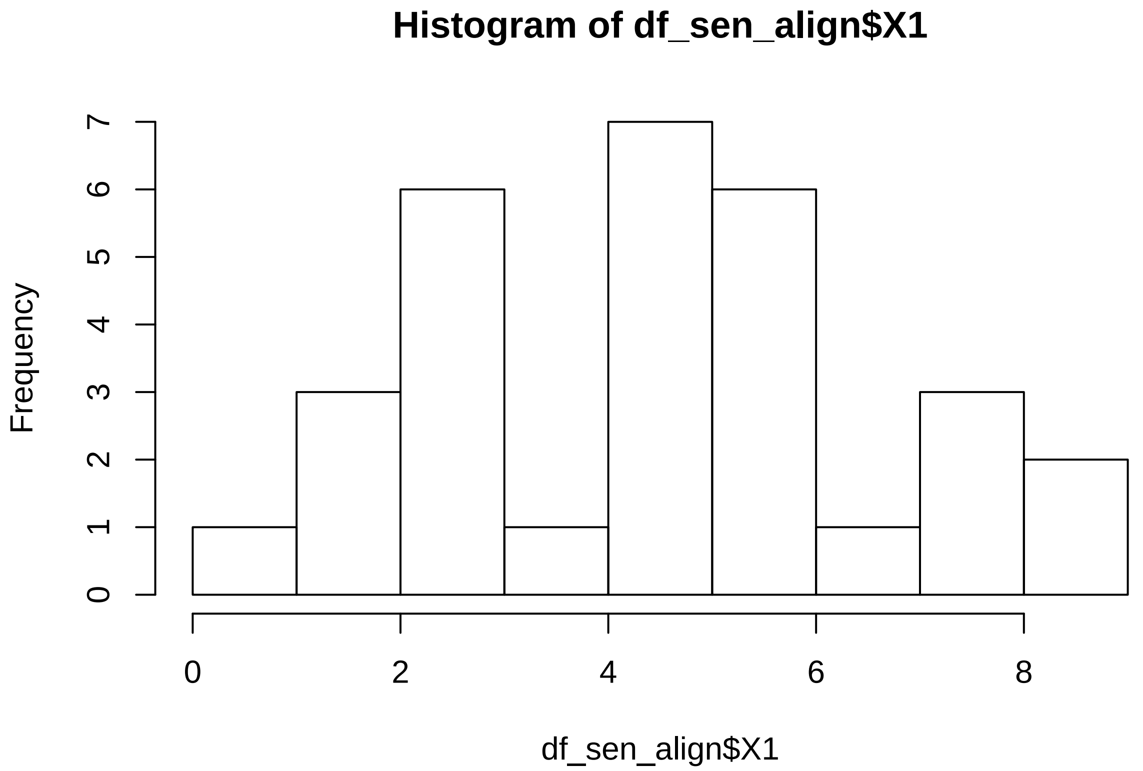
Qusetion a

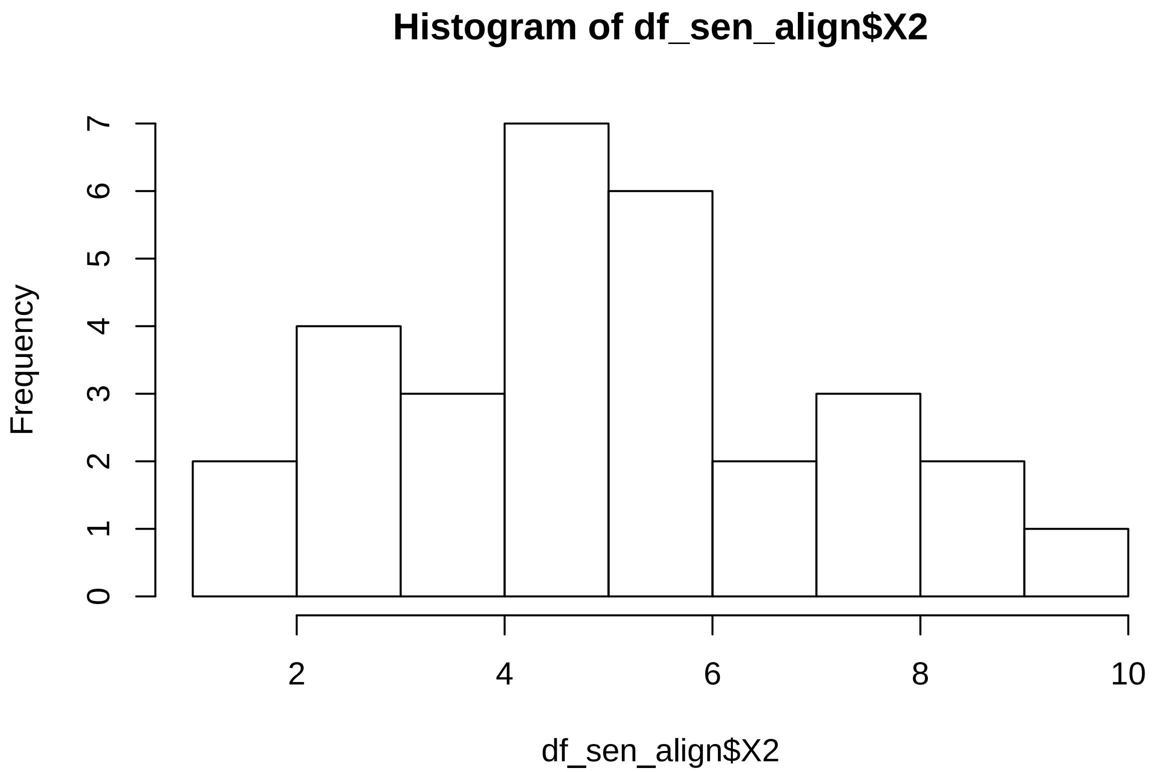
```
url_sen<-"https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
df_sen<-read.table(url_sen, skip=1, fill=TRUE, header=TRUE)
```

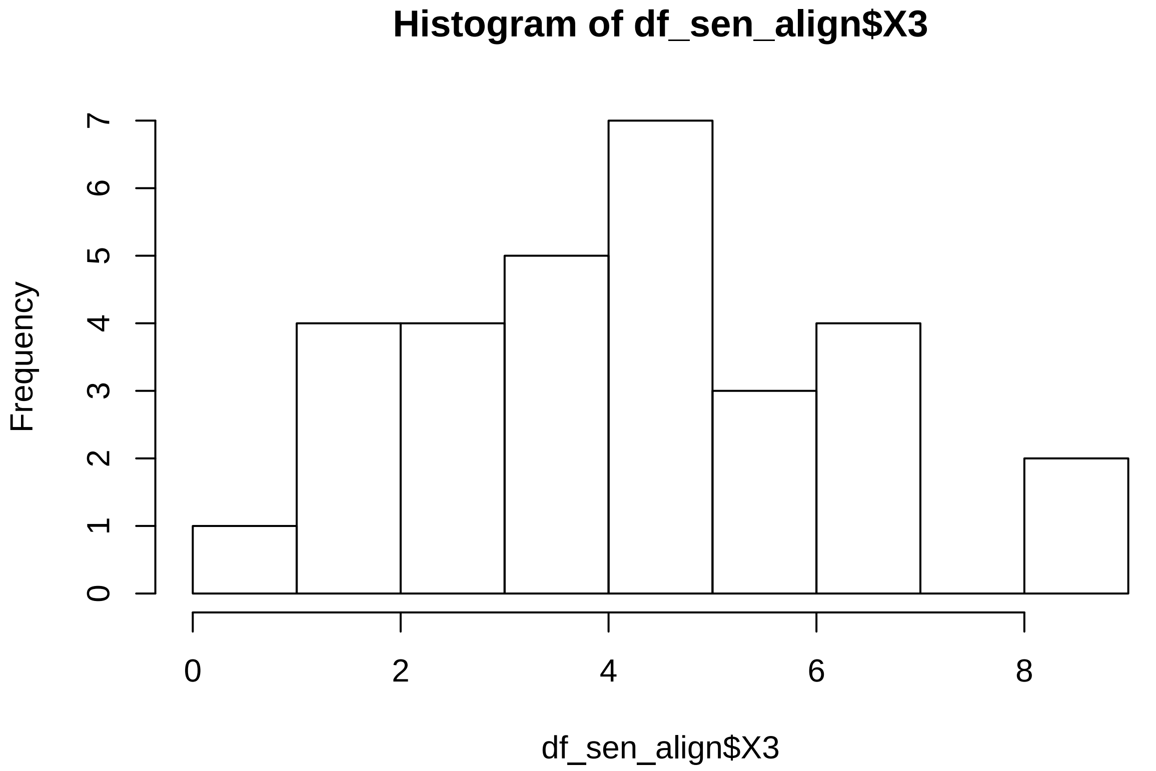
The issues in this data is the location of missing value is incorrect, so we use this function to align the data.

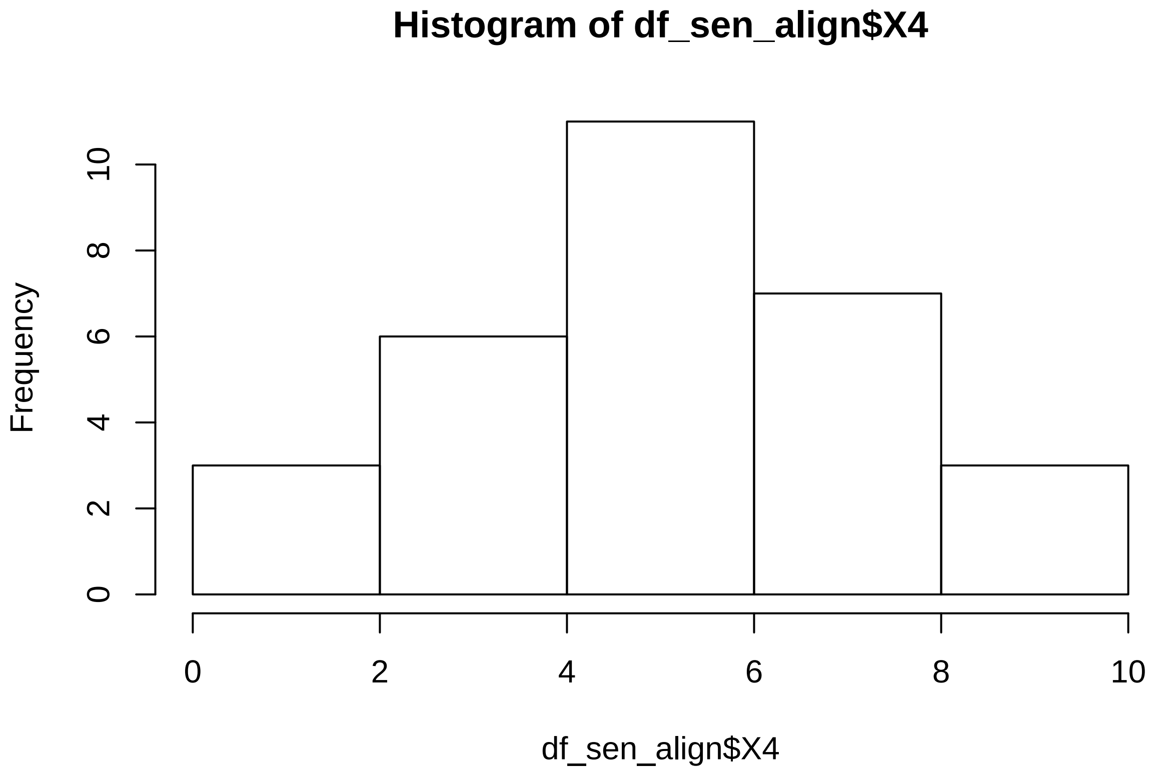
```
align_data <- function(row_){
    if (is.na(row_['X5'])){
      row_[2:6] <- row_[1:5]
      row_[1] <- NA
    }
    return(row_)
}
df_sen_align<-data.table(t(apply(df_sen, 1, align_data)))
```

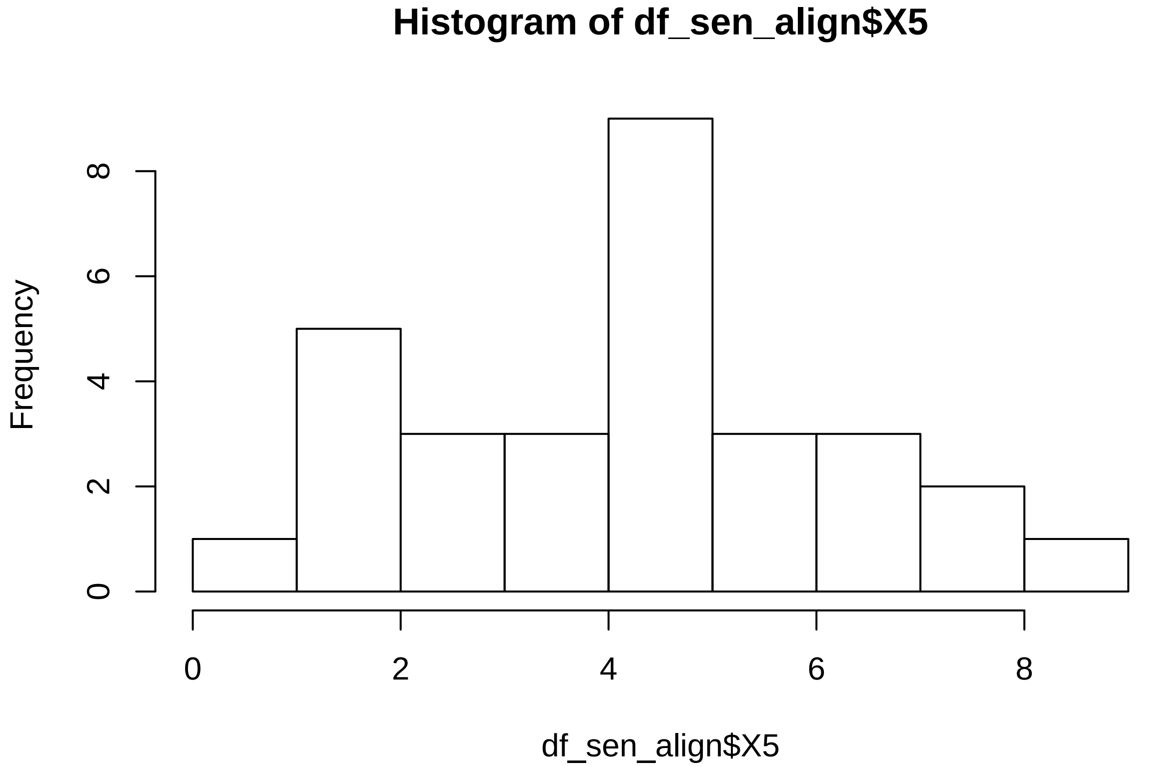
Result




Histogram of df_sen_align$X2
Frequency
0
1
2
3
4
5
6
7
2
4
6
8
10
df_sen_align$X2


Histogram of df_sen_align$X3
Frequency
0
1
2
3
4
5
6
7
0
2
4
6
8
df_sen_align$X3





X1	X2	X3	X4	X5
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

X1	X2	X3	X4	X5
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

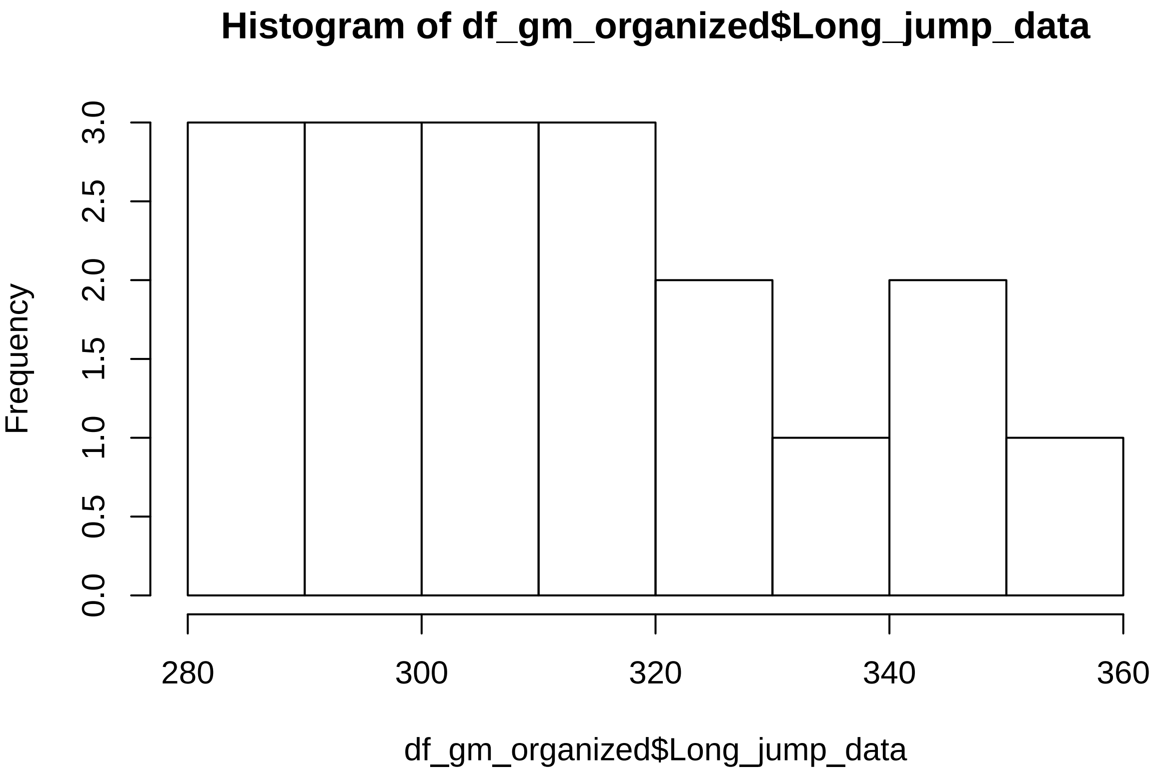
Question b

```
url_gm<-"https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
df_gm<-read.table(url_gm, skip=1, fill=TRUE, header=TRUE)
```

The issue in this data is that the columns are messy, so we should reorganized it.

```
colnames(df_gm)<- NA
df_gm_organized <- rbind(df_gm[,1:2],df_gm[,3:4],df_gm[,5:6],df_gm[,7:8])
colnames(df_gm_organized) <- c('Year', 'Long_jump_data')
```

Result



Long_jump_data
Min. :281.5
1st Qu.:298.3
Median :312.7
Mean :313.3
3rd Qu.:327.5
Max. :350.5
NA's :2

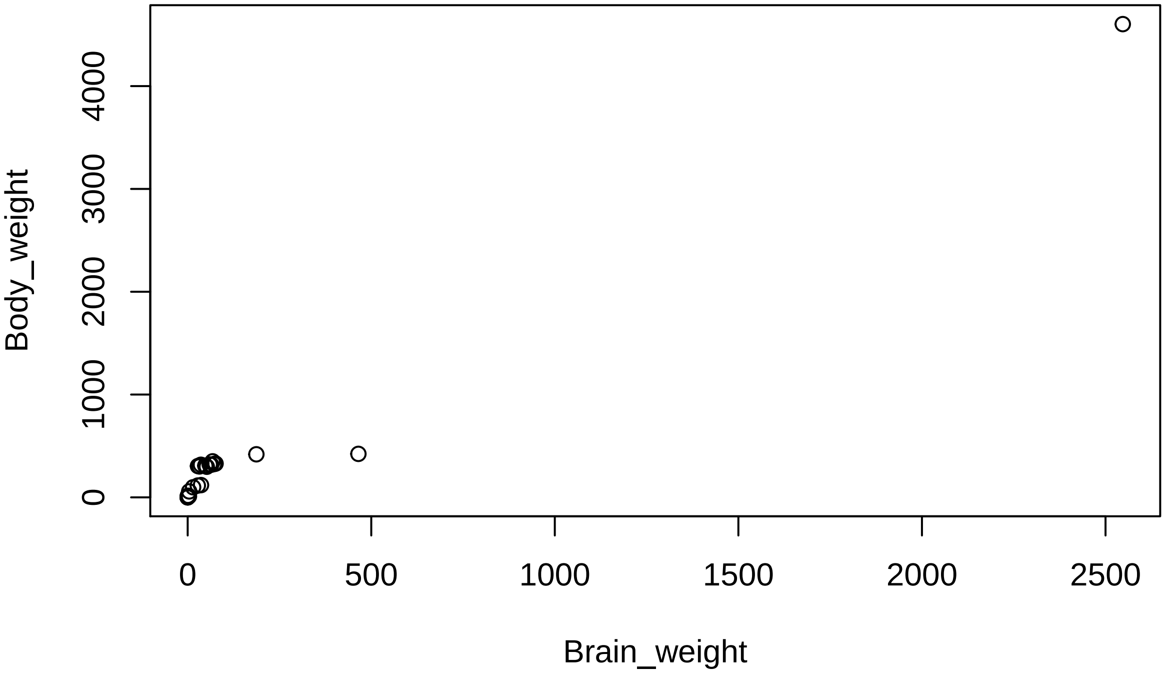
Question c

```r
url_bw<-"https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
df_bw<-read.table(url_bw, skip=1, fill=TRUE, header=TRUE)
```

The issue in this data is similar to question b.

```r
colnames(df_bw)<- NA
df_bw_organized <- rbind(df_bw[,1:2],df_gm[,3:4],df_gm[,5:6])
colnames(df_bw_organized) <- c('Brain_weight', 'Body_weight')
```

Result



Brain_weight	Body_weight
Min. : 0.005	Min. : 0.10
1st Qu.: 1.010	1st Qu.: 6.45
Median : 21.245	Median : 106.60
Mean : 127.691	Mean : 302.94
3rd Qu.: 58.000	3rd Qu.: 317.64
Max. :2547.000	Max. :4603.00

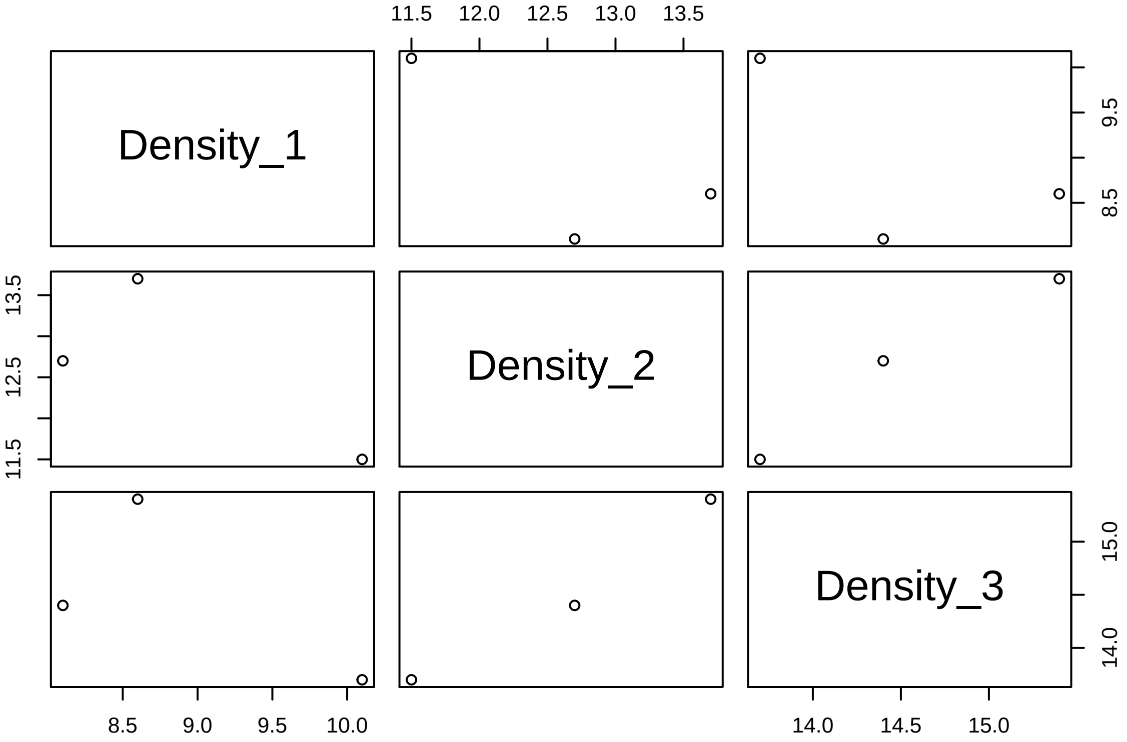
Question d

```r
url_to<-"https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"
df_to<-read.table(url_to, skip=1, fill=TRUE, header=TRUE)
```

The issue in this data is they put three densities together and splited by comma, so we first convert them to vectors then to data table.

```
get_density <- function(element){
  return(as.numeric(unlist(strsplit(element,split=","))))
}
df_to_row <- data.table(apply(df_to[2,],2,get_density))
colnames(df_to_row) <- c('Density_1', 'Density_2', 'Density_3')
```

Result



Density_1	Density_2	Density_3
Min. : 8.100	Min. :11.50	Min. :13.70
1st Qu.: 8.350	1st Qu.:12.10	1st Qu.:14.05
Median : 8.600	Median :12.70	Median :14.40
Mean : 8.933	Mean :12.63	Mean :14.50
3rd Qu.: 9.350	3rd Qu.:13.20	3rd Qu.:14.90
Max. :10.100	Max. :13.70	Max. :15.40