PROG8430_Assignment 2

Geedhu

2023-06-21

Basic Set Up

```
# Clear plots
if(!is.null(dev.list())) dev.off()
```

```
## null device
##           1
```

```
# Clear console
cat("\014")
```

```r
# Clean workspace
rm(list=ls())
```

PART 1

```r
# Read the data Assign#2_1.txt
data_set = read.table("Assign#2_1.txt", header = TRUE,sep = ",")

head(data_set)
```

```
##   Index Manufacturer Server Conn RC  ST  SMBR   SMBT BR BT   UC      FA
## 1     1         Lled MB5755 5571 10 253 39806  91685 11 17 2000 1526223
## 2     2         Lled MB3406 6684 12 282 56410 115100 15 22 2000 1799882
## 3     3       Ovonel RQ8547 4790 10  83 55891  98534 15 18 2000 1361793
## 4     4         Lled MB3406 6163 10 247 49546 116361 14 22 2000 2365969
## 5     5         Lled MB5755 8939 12 252 61578 104176 17 20 2000 2222282
## 6     6       Ovonel RP6354 7446  8 263 47692 102983 13 19 2000 2006826
```

```r
# Removing Index Column

data_set <- data_set[,-c(1)]

head(data_set)
```

```
##   Manufacturer Server Conn RC  ST  SMBR   SMBT BR BT   UC      FA
## 1         Lled MB5755 5571 10 253 39806  91685 11 17 2000 1526223
## 2         Lled MB3406 6684 12 282 56410 115100 15 22 2000 1799882
## 3       Ovonel RQ8547 4790 10  83 55891  98534 15 18 2000 1361793
## 4         Lled MB3406 6163 10 247 49546 116361 14 22 2000 2365969
## 5         Lled MB5755 8939 12 252 61578 104176 17 20 2000 2222282
## 6       Ovonel RP6354 7446  8 263 47692 102983 13 19 2000 2006826
```

Data Transformation and Preparation

QUESTION 1: Initial Transformation

```r
# a. Append your initials to all variables in the data frame

new_names <- paste0(names(data_set), "_kv")

colnames(data_set) <- new_names

colnames(data_set)
```

```
##  [1] "Manufacturer_kv" "Server_kv"       "Conn_kv"         "RC_kv"
##  [5] "ST_kv"           "SMBR_kv"         "SMBT_kv"         "BR_kv"
##  [9] "BT_kv"           "UC_kv"           "FA_kv"
```

```r
# b. Transform character variables to factor variables.

# To check for character variables
str(data_set)
```

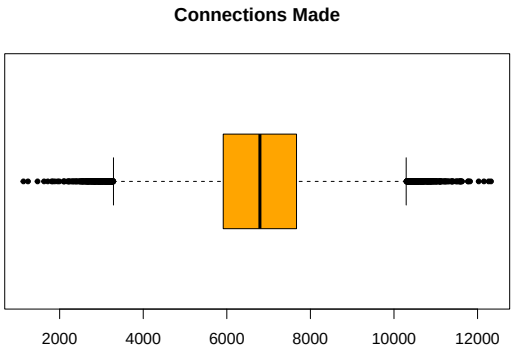
```
## 'data.frame':    82156 obs. of  11 variables:
##  $ Manufacturer_kv: chr  "Lled" "Lled" "Ovonel" "Lled" ...
##  $ Server_kv      : chr  "MB5755" "MB3406" "RQ8547" "MB3406" ...
##  $ Conn_kv        : int  5571 6684 4790 6163 8939 7446 8618 7319 5853 7667 ...
##  $ RC_kv          : int  10 12 10 10 12 8 13 3 7 12 ...
##  $ ST_kv          : int  253 282 83 247 252 263 118 271 283 256 ...
##  $ SMBR_kv        : int  39806 56410 55891 49546 61578 47692 50814 49125 62117 58279 ...
##  $ SMBT_kv        : int  91685 115100 98534 116361 104176 102983 102608 99735 127959 109037 ...
##  $ BR_kv          : int  11 15 15 14 17 13 14 13 17 16 ...
##  $ BT_kv          : int  17 22 18 22 20 19 19 19 24 20 ...
##  $ UC_kv          : int  2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 ...
##  $ FA_kv          : int  1526223 1799882 1361793 2365969 2222282 2006826 1043945 1283390 1795163 112
```

```r
# Converting character variables to factor variables
data_set$Manufacturer_kv <- factor(data_set$Manufacturer_kv)
data_set$Server_kv <- factor(data_set$Server_kv)

#check if char variables are changed to factor variables
str(data_set)
```

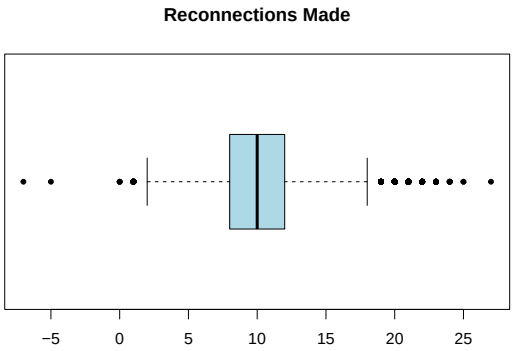
```
## 'data.frame':    82156 obs. of  11 variables:
##  $ Manufacturer_kv: Factor w/ 2 levels "Lled","Ovonel": 1 1 2 1 1 2 2 2 1 2 ...
##  $ Server_kv      : Factor w/ 7 levels "MB3406","MB5755",..: 2 1 6 1 2 5 5 5 2 4 ...
##  $ Conn_kv        : int  5571 6684 4790 6163 8939 7446 8618 7319 5853 7667 ...
##  $ RC_kv          : int  10 12 10 10 12 8 13 3 7 12 ...
##  $ ST_kv          : int  253 282 83 247 252 263 118 271 283 256 ...
##  $ SMBR_kv        : int  39806 56410 55891 49546 61578 47692 50814 49125 62117 58279 ...
##  $ SMBT_kv        : int  91685 115100 98534 116361 104176 102983 102608 99735 127959 109037 ...
##  $ BR_kv          : int  11 15 15 14 17 13 14 13 17 16 ...
##  $ BT_kv          : int  17 22 18 22 20 19 19 19 24 20 ...
##  $ UC_kv          : int  2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 ...
##  $ FA_kv          : int  1526223 1799882 1361793 2365969 2222282 2006826 1043945 1283390 1795163 112
```

QUESTION 2: Outliers

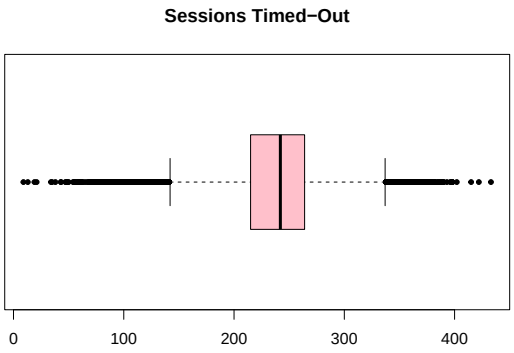
```r
###################### a. Use an appropriate technique demonstrated in class to identify outliers.
###################### b. Comment on any outliers you see and deal with them appropriately.
######################    Make sure you explain why you dealt with them the way you decided to.

### Box Plots for 1.5 IQR
#As for all numerical data, check Box Plots

boxplot(data_set$Conn_kv, horizontal=TRUE, pch=20,main = "Connections Made",col="Orange")
```



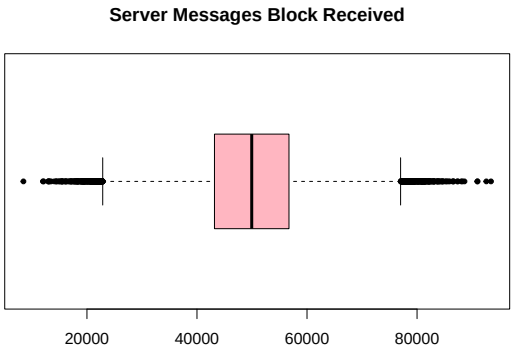

```
boxplot(data_set$RC_kv, horizontal=TRUE, pch=20,main = "Reconnections Made",col="Light Blue")
```



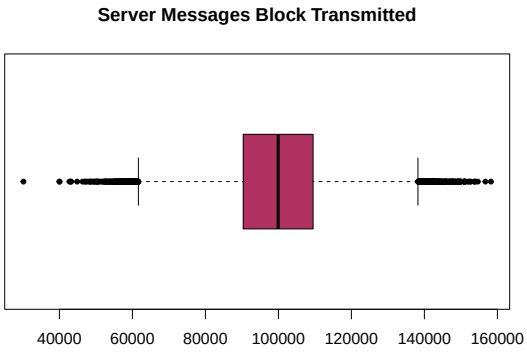

```
boxplot(data_set$ST_kv, horizontal=TRUE, pch=20,main = "Sessions Timed-Out",col="Pink")
```



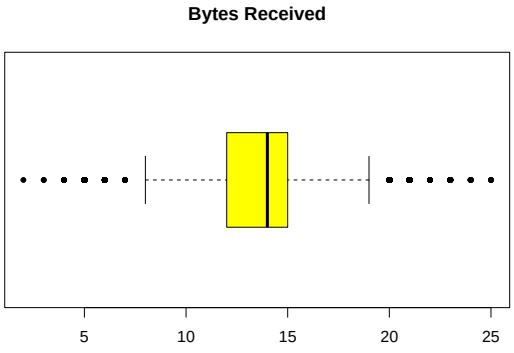

```
boxplot(data_set$SMBR_kv, horizontal=TRUE, pch=20,
        main = "Server Messages Block Received",col="Light Pink")
```



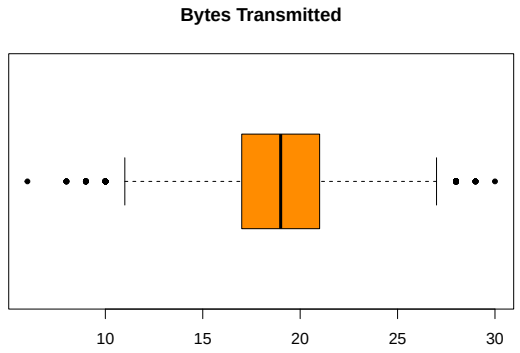

```
boxplot(data_set$SMBT_kv, horizontal=TRUE, pch=20,
        main = "Server Messages Block Transmitted",col="Maroon")
```



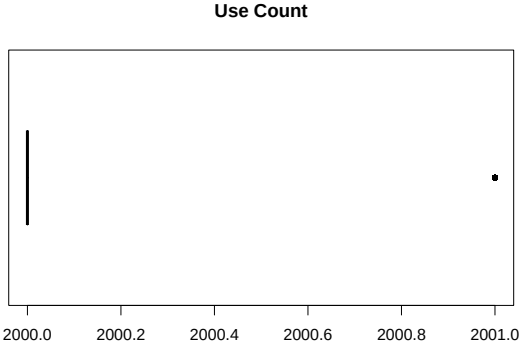

```
boxplot(data_set$BR_kv, horizontal=TRUE, pch=20,main = "Bytes Received",col="Yellow")
```



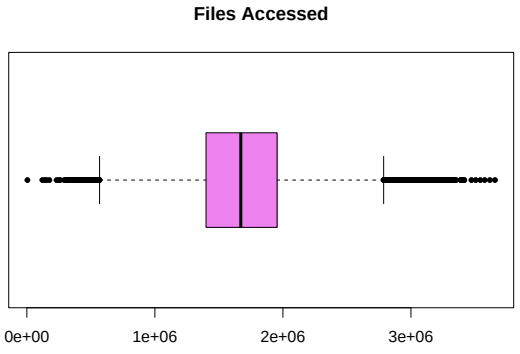

```
boxplot(data_set$BT_kv, horizontal=TRUE, pch=20,main = "Bytes Transmitted",col="Dark Orange")
```




```
boxplot(data_set$UC_kv, horizontal=TRUE, pch=20,main = "Use Count",col="Blue")
```




```
boxplot(data_set$FA_kv, horizontal=TRUE, pch=20,main = "Files Accessed",col="Violet")
```

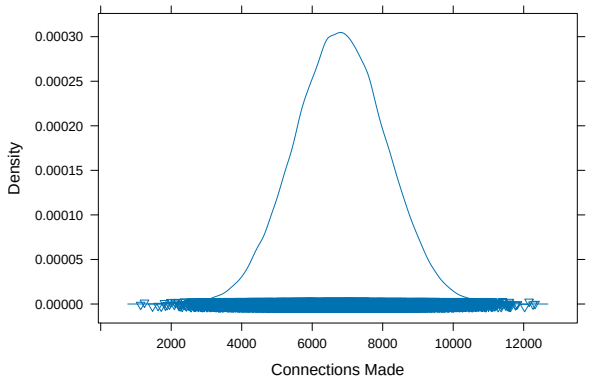


Analysis on box Plot

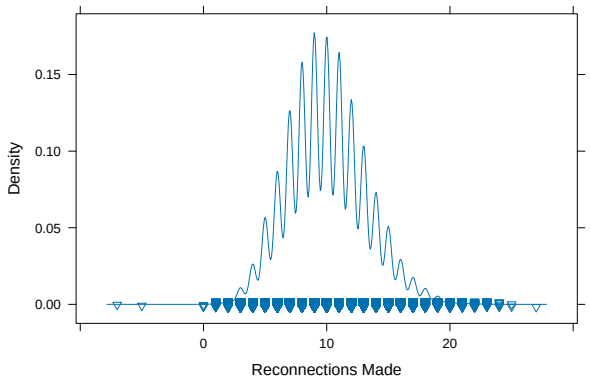
We could see that all columns have outliers, lets come to conclusion after looking into the density Plots.

```
### Density plot for further details

library("lattice")
```

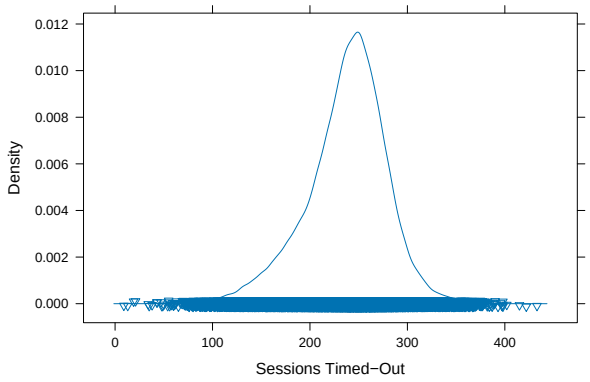
```
densityplot( ~ data_set$Conn_kv, pch=6, xlab = "Connections Made")
```



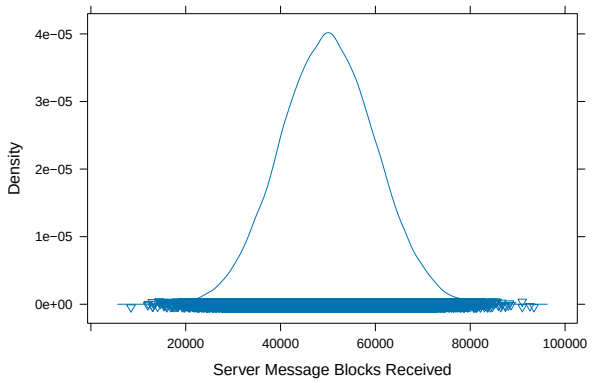

```
densityplot( ~ data_set$RC_kv, pch=6, xlab = "Reconnections Made")
```



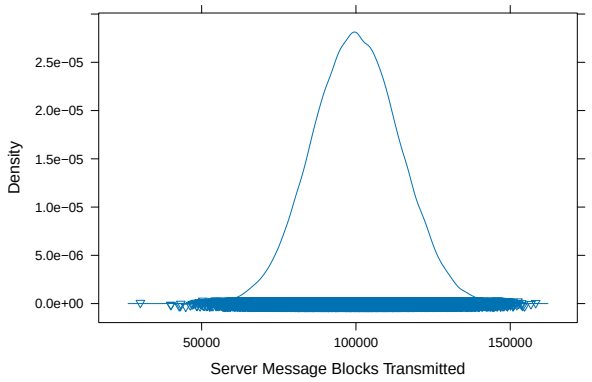

```
densityplot( ~ data_set$ST_kv, pch=6, xlab = "Sessions Timed-Out")
```



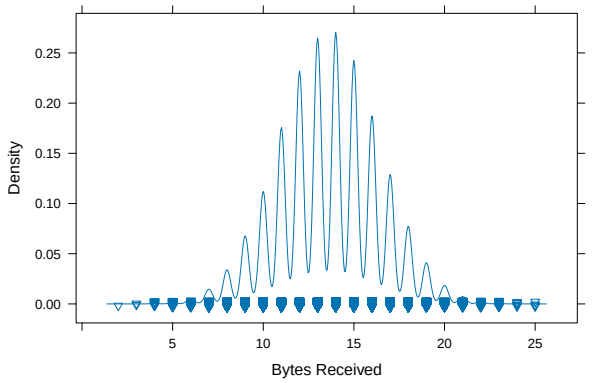

```
densityplot( ~ data_set$SMBR_kv, pch=6, xlab = "Server Message Blocks Received")
```



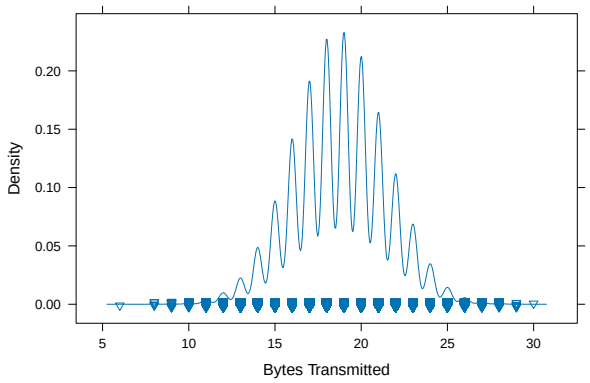

```
densityplot( ~ data_set$SMBT_kv, pch=6, xlab = "Server Message Blocks Transmitted")
```



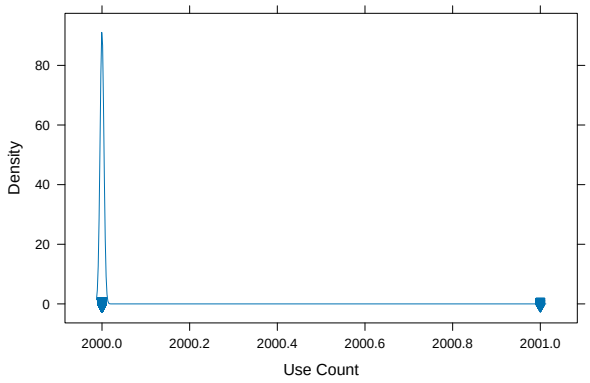

```
densityplot( ~ data_set$BR_kv, pch=6,xlab = "Bytes Received")
```



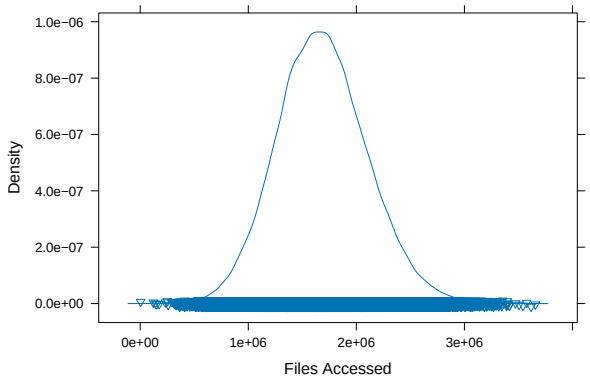

```
densityplot( ~ data_set$BT_kv, pch=6, xlab = "Bytes Transmitted")
```




```
densityplot( ~ data_set$UC_kv, pch=6, xlab = "Use Count")
```




```
densityplot( ~ data_set$FA_kv, pch=6, xlab = "Files Accessed")
```



Analysis about Outliers of each attribute based on Density Plot

Other than "Use Count" attribute, rest all attributes are not that out of range from the analysis obtained from scatter plot. Use Count is an attribute, which gives details about the number of users used, just because we have few observations for 2001, we cant ignore. There is always a possibility of special out of range condition to persist in real time. Hence not Removing the outliers of use count. Let's analyze a bit more.

```
### To ensure if there are only two distinct values in the Use column.

cat("Number of unique values in the dataset that belongs to use count: ",unique(data_set$UC_kv))
```

```
## Number of unique values in the dataset that belongs to use count:  2000 2001
```

```
### Count number of values in the dataset that belongs to use count > 2000.

use_count_values <- data_set[data_set$UC_kv != 2000,]


cat("Number of values in the dataset that belongs to use count > 2000: ",nrow(use_count_values))
```

```
## Number of values in the dataset that belongs to use count > 2000:  167
```

```
cat ('\nFINAL CONCLUSION\n')
```

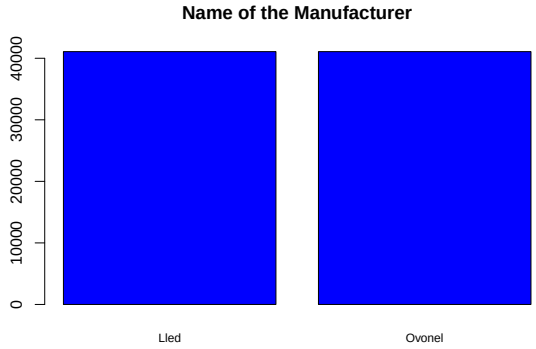
```
##
## FINAL CONCLUSION
```

```
cat ("There are only 167 observation available for 2001 out of complete set of observations that is 821
```

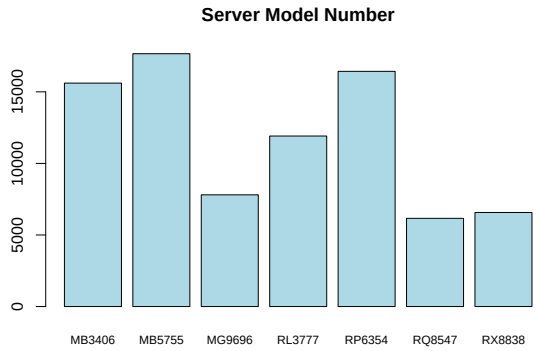
```
## There are only 167 observation available for 2001 out of complete set of observations that is 82156.
```

Let's Analyze categorical values for outliers

```
barplot(table(data_set$Manufacturer_kv), cex.names=.75,col='blue',main = "Name of the Manufacturer")
```




```
barplot(table(data_set$Server_kv), cex.names=.75,col='Light blue',main = "Server Model Number")
```



From Bar Plot Analysis, there are no outliers.

QUESTION 3: Reduce Dimensionality

```
###### Step1: Print blank data cells
blank_data_cells <- data_set[data_set == ""]
print("Number of blank cells in the dataset are:")
```

a. Apply the missing value filter to remove appropriate columns of data.

```
## [1] "Number of blank cells in the dataset are:"
```

```
print(blank_data_cells)
```

```
## character(0)
```

```
na_data_cells <- data_set[data_set == "NA"]
print("Number of NA cells in the dataset are:")
```

```
## [1] "Number of NA cells in the dataset are:"
```

```
print(na_data_cells)
```

```
## character(0)
```

```
print("Answer: As there are no blank cells, hence no need to find the missing value ratio.")
```

```
## [1] "Answer: As there are no blank cells, hence no need to find the missing value ratio."
```

```
library("pastecs")
##### 2. Identify Low Variance
stat.desc(data_set)
```

b. Apply the low variance filter to remove appropriate columns of data.

```
##          Manufacturer_kv Server_kv      Conn_kv         RC_kv        ST_kv
## nbr.val               NA        NA 8.215600e+04  8.215600e+04 8.215600e+04
## nbr.null              NA        NA 0.000000e+00  3.000000e+00 0.000000e+00
## nbr.na                NA        NA 0.000000e+00  0.000000e+00 0.000000e+00
## min                   NA        NA 1.133000e+03 -7.000000e+00 9.000000e+00
## max                   NA        NA 1.232100e+04  2.700000e+01 4.330000e+02
## range                 NA        NA 1.118800e+04  3.400000e+01 4.240000e+02
## sum                   NA        NA 5.581174e+08  8.218390e+05 1.950666e+07
## median                NA        NA 6.792000e+03  1.000000e+01 2.420000e+02
## mean                  NA        NA 6.793386e+03  1.000340e+01 2.374344e+02
## SE.mean               NA        NA 4.541825e+00  1.106041e-02 1.447164e-01
## CI.mean               NA        NA 8.901944e+00  2.167833e-02 2.836431e-01
## var                   NA        NA 1.694728e+06  1.005037e+01 1.720579e+03
## std.dev               NA        NA 1.301817e+03  3.170232e+00 4.147986e+01
## coef.var              NA        NA 1.916301e-01  3.169155e-01 1.747003e-01
##               SMBR_kv      SMBT_kv        BR_kv        BT_kv        UC_kv
## nbr.val  8.215600e+04 8.215600e+04 8.215600e+04 8.215600e+04 8.215600e+04
## nbr.null 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## nbr.na   0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## min      8.455000e+03 3.013900e+04 2.000000e+00 6.000000e+00 2.000000e+03
## max      9.343700e+04 1.582470e+05 2.500000e+01 3.000000e+01 2.001000e+03
## range    8.498200e+04 1.281080e+05 2.300000e+01 2.400000e+01 1.000000e+00
## sum      4.105258e+09 8.213533e+09 1.119635e+06 1.540088e+06 1.643122e+08
## median   4.995200e+04 9.994000e+04 1.400000e+01 1.900000e+01 2.000000e+03
## mean     4.996906e+04 9.997484e+04 1.362816e+01 1.874590e+01 2.000002e+03
## SE.mean  3.484991e+01 4.931500e+01 9.557366e-03 9.305629e-03 1.571375e-04
## CI.mean  6.830558e+01 9.665704e+01 1.873237e-02 1.823897e-02 3.079883e-04
## var      9.977982e+07 1.998008e+08 7.504396e+00 7.114276e+00 2.028611e-03
## std.dev  9.988985e+03 1.413509e+04 2.739415e+00 2.667260e+00 4.504010e-02
## coef.var 1.999034e-01 1.413865e-01 2.010114e-01 1.422850e-01 2.252003e-05
##                FA_kv
## nbr.val  8.215600e+04
## nbr.null 0.000000e+00
## nbr.na   0.000000e+00
## min      4.412000e+03
## max      3.656283e+06
## range    3.651871e+06
## sum      1.384389e+11
## median   1.671136e+06
## mean     1.685073e+06
## SE.mean  1.452404e+03
## CI.mean  2.846701e+03
## var      1.733062e+11
## std.dev  4.163006e+05
## coef.var 2.470519e-01
```

```
print("Answer: All the attributes has coefficient value more or less comes under the same range. Hence r
```

```
## [1] "Answer: All the attributes has coefficient value more or less comes under the same range. Hence
```

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

corr_mat <- round(cor(data_set_num),2)

print(head(corr_mat))
```

c. Apply the high correlation filter to remove appropriate columns of data.

```
##         RC_kv ST_kv SMBR_kv SMBT_kv BR_kv BT_kv UC_kv FA_kv
## RC_kv       1     0    0.00    0.00  0.00  0.00     0     0
## ST_kv       0     1    0.00    0.00  0.00  0.00     0     0
## SMBR_kv     0     0    1.00    0.76  0.99  0.76     0     0
## SMBT_kv     0     0    0.76    1.00  0.76  0.99     0     0
## BR_kv       0     0    0.99    0.76  1.00  0.75     0     0
## BT_kv       0     0    0.76    0.99  0.75  1.00     0     0
```

```
print("Answer: From the output received, it is very evident that the pair: SMBR_KV and BR_KV  \n
and pair: SMBT_KV and BT_KV has high correlation with 0.99 and also both refers to the same context.\n
Hence removing BR_KV and BT_KV ")
```

```
## [1] "Answer: From the output received, it is very evident that the pair: SMBR_KV and BR_KV  \n\nand p
```

```
### Dataset before removing BR_KV and BT_KV
print(head(data_set))
```

```
##   Manufacturer_kv Server_kv Conn_kv RC_kv ST_kv SMBR_kv SMBT_kv BR_kv BT_kv
## 1            Lled    MB5755    5571    10   253   39806   91685    11    17
## 2            Lled    MB3406    6684    12   282   56410  115100    15    22
## 3          Ovonel    RQ8547    4790    10    83   55891   98534    15    18
## 4            Lled    MB3406    6163    10   247   49546  116361    14    22
## 5            Lled    MB5755    8939    12   252   61578  104176    17    20
## 6          Ovonel    RP6354    7446     8   263   47692  102983    13    19
##   UC_kv   FA_kv
## 1  2000 1526223
## 2  2000 1799882
## 3  2000 1361793
## 4  2000 2365969
## 5  2000 2222282
## 6  2000 2006826
```

```
data_set <- data_set[,-c(8,9)]

### Dataset after removing BR_KV and BT_KV
print(head(data_set))
```

```
##   Manufacturer_kv Server_kv Conn_kv RC_kv ST_kv SMBR_kv SMBT_kv UC_kv   FA_kv
## 1            Lled    MB5755    5571    10   253   39806   91685  2000 1526223
## 2            Lled    MB3406    6684    12   282   56410  115100  2000 1799882
## 3          Ovonel    RQ8547    4790    10    83   55891   98534  2000 1361793
## 4            Lled    MB3406    6163    10   247   49546  116361  2000 2365969
## 5            Lled    MB5755    8939    12   252   61578  104176  2000 2222282
## 6          Ovonel    RP6354    7446     8   263   47692  102983  2000 2006826
```

```
print("Attribute:\"Manufacturer_kv\", does not contribute much value to the prediction or training of m
```

d. Drop any variables that do not contribute any useful analytical information at all.

```
## [1] "Attribute:\"Manufacturer_kv\", does not contribute much value to the prediction or training of m
```

```
data_set <- data_set[,-1]

print(head(data_set))
```

```
##   Server_kv Conn_kv RC_kv ST_kv SMBR_kv SMBT_kv UC_kv   FA_kv
## 1    MB5755    5571    10   253   39806   91685  2000 1526223
## 2    MB3406    6684    12   282   56410  115100  2000 1799882
## 3    RQ8547    4790    10    83   55891   98534  2000 1361793
## 4    MB3406    6163    10   247   49546  116361  2000 2365969
## 5    MB5755    8939    12   252   61578  104176  2000 2222282
## 6    RP6354    7446     8   263   47692  102983  2000 2006826
```

PART 2 (12 points) Please use the Dataset "Assign#2_2.txt" to answer following questions.

```
# Clear plots
if(!is.null(dev.list())) dev.off()
```

```
## null device
##           1
```

```
# Clear console
cat("\014")
```

```
# Clean workspace
rm(list=ls())
```

```
# Read the data Assign#2_2.txt

dataset = read.table("Assign#2_2.txt", header = TRUE,sep = ",")
print(head(dataset))
```

```
##    Food Enter   Edu Trans  Work House   Oth
## 1 0.043 0.085 0.525 0.180 0.005 0.150 0.012
## 2 0.123 0.055 0.002 0.169 0.121 0.266 0.265
## 3 0.043 0.085 0.506 0.193 0.006 0.155 0.012
## 4 0.119 0.038 0.002 0.301 0.139 0.228 0.172
## 5 0.122 0.038 0.002 0.225 0.095 0.354 0.164
## 6 0.084 0.050 0.002 0.285 0.079 0.264 0.237
```

1. Data Transformation

```
#1. Rename all variables with your initials appended (just as was done in assignment 1)

new_names <- paste0(names(dataset), "_kv")

colnames(dataset) <- new_names

colnames(dataset)
```

```
## [1] "Food_kv"  "Enter_kv" "Edu_kv"   "Trans_kv" "Work_kv"  "House_kv" "Oth_kv"
```

```
# 2. Standardize all of the variables using either of the two functions demonstrated in class.
#    Describe why you chose the method you did.

# Lets analyze the distribution of data, before coming to an conclusion.
summary(dataset)
```

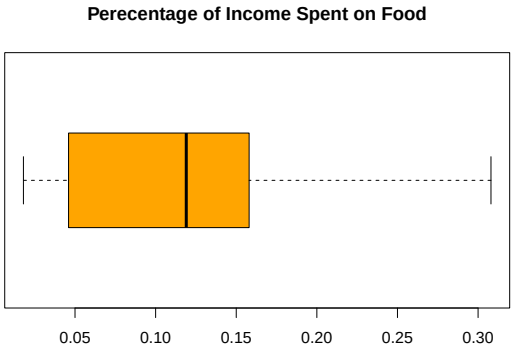
```
##     Food_kv          Enter_kv           Edu_kv          Trans_kv
##  Min.   :0.0180   Min.   :0.00400   Min.   :0.0010   Min.   :0.0190
##  1st Qu.:0.0460   1st Qu.:0.03100   1st Qu.:0.0020   1st Qu.:0.1570
##  Median :0.1190   Median :0.04200   Median :0.0690   Median :0.2020
##  Mean   :0.1111   Mean   :0.04551   Mean   :0.2271   Mean   :0.1957
##  3rd Qu.:0.1580   3rd Qu.:0.06100   3rd Qu.:0.5380   3rd Qu.:0.2410
##  Max.   :0.3080   Max.   :0.11300   Max.   :0.7210   Max.   :0.3710
##     Work_kv           House_kv          Oth_kv
##  Min.   :0.00200   Min.   :0.0360   Min.   :0.004
##  1st Qu.:0.00500   1st Qu.:0.1500   1st Qu.:0.010
##  Median :0.09000   Median :0.2450   Median :0.118
##  Mean   :0.08136   Mean   :0.2383   Mean   :0.101
##  3rd Qu.:0.13750   3rd Qu.:0.3105   3rd Qu.:0.169
##  Max.   :0.25600   Max.   :0.5090   Max.   :0.305
```

```r
# From summary, we got to know that values of all the attributes comes in between the range 0 to 1. Hen

dataset1 <- scale(dataset)

print(head(dataset1))
```

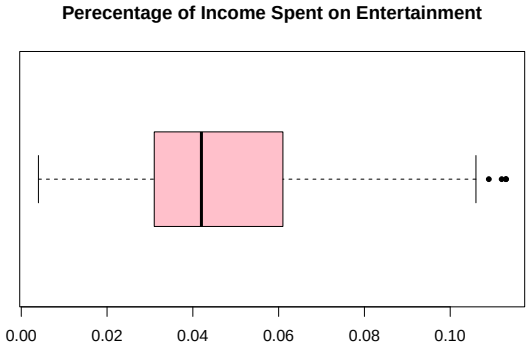
```
##          Food_kv   Enter_kv     Edu_kv    Trans_kv     Work_kv   House_kv
## [1,] -1.0673000  1.9070370  1.1268936 -0.24846555 -1.11649264 -0.8726466
## [2,]  0.1865015  0.4581869 -0.8515384 -0.42277301  0.57953573  0.2737239
## [3,] -1.0673000  1.9070370  1.0550194 -0.04246584 -1.10187171 -0.8232341
## [4,]  0.1238114 -0.3628282 -0.8515384  1.66891641  0.84271254 -0.1018113
## [5,]  0.1708290 -0.3628282 -0.8515384  0.46461038  0.19939144  1.1433843
## [6,] -0.4247267  0.2167119 -0.8515384  1.41537830 -0.03454351  0.2539589
##           Oth_kv
## [1,] -1.1035999
## [2,]  2.0345987
## [3,] -1.1035999
## [4,]  0.8810316
## [5,]  0.7818000
## [6,]  1.6872882
```

2. Descriptive Data Analysis

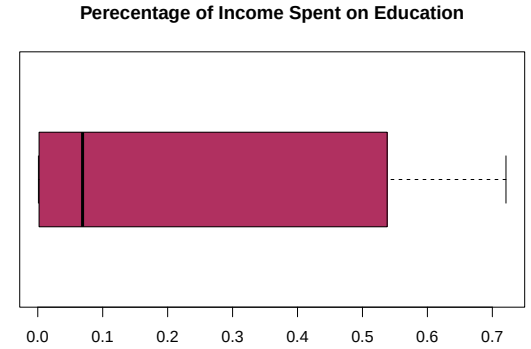
```r
# 1. Create graphical summaries of the data (as demonstrated in class: boxplots, histograms or density

boxplot(dataset$Food_kv, horizontal=TRUE, pch=20,main = "Perecentage of Income Spent on Food",col="Oran
```



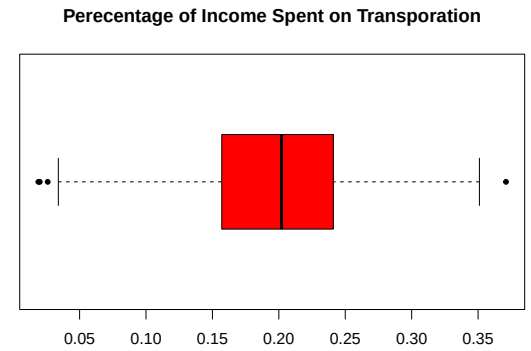
```
boxplot(dataset$Enter_kv, horizontal=TRUE, pch=20,main = "Perecentage of Income Spent on Entertainment"
```



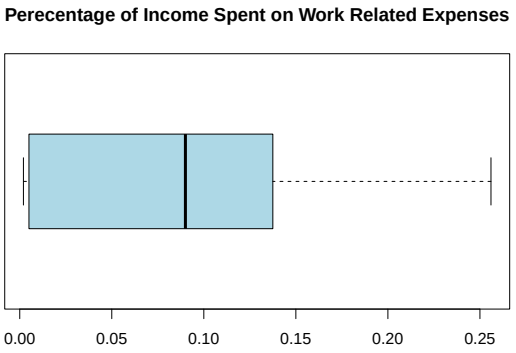

```
boxplot(dataset$Edu_kv, horizontal=TRUE, pch=20,main = "Perecentage of Income Spent on Education",col="
```



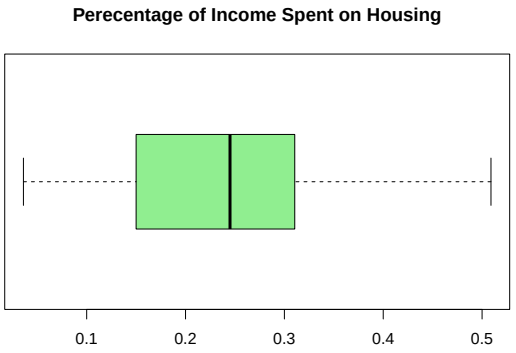

```
boxplot(dataset$Trans_kv, horizontal=TRUE, pch=20,main = "Perecentage of Income Spent on Transporation"
```



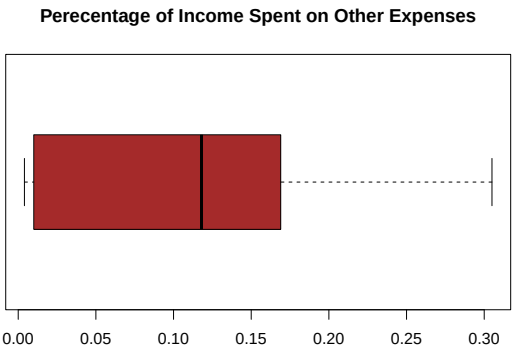

```
boxplot(dataset$Work_kv, horizontal=TRUE, pch=20,main = "Perecentage of Income Spent on Work Related Ex
```



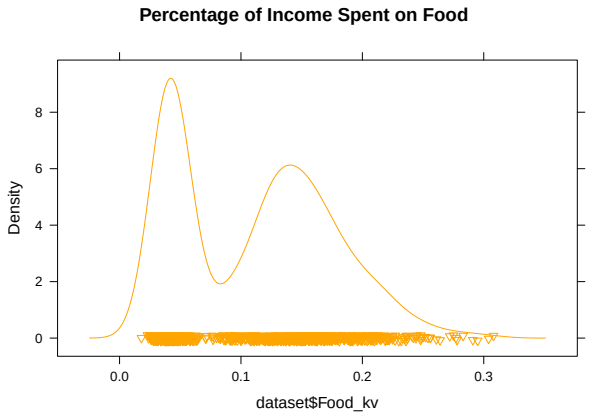

```
boxplot(dataset$House_kv, horizontal=TRUE, pch=20,main = "Perecentage of Income Spent on Housing",col="
```



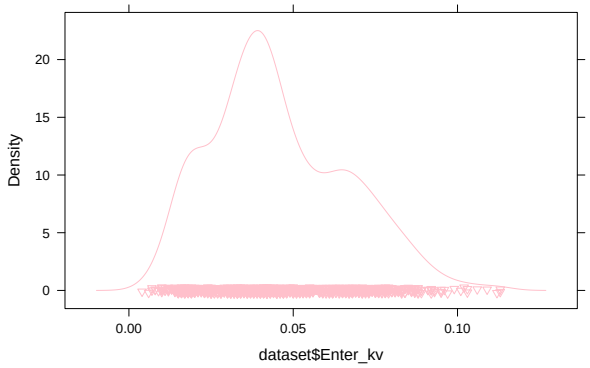

```
boxplot(dataset$Oth_kv, horizontal=TRUE, pch=20,main = "Perecentage of Income Spent on Other Expenses",
```



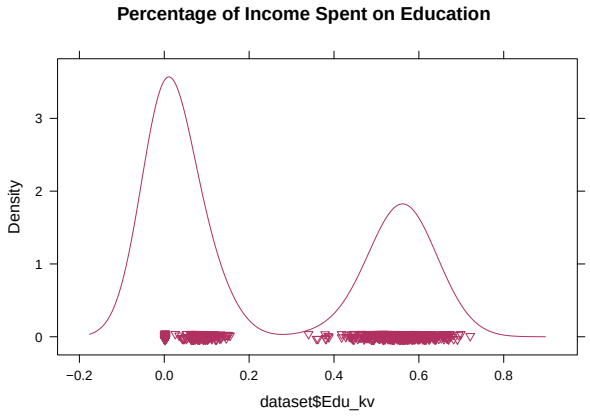

```
#### Density plot

densityplot( ~ dataset$Food_kv, pch=6,,main = "Percentage of Income Spent on Food",col="Orange")
```



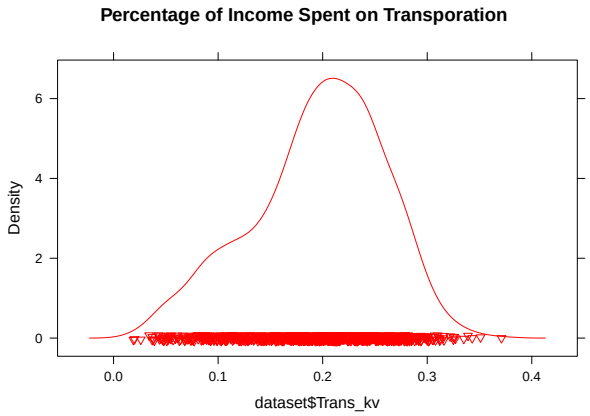

```
densityplot( ~ dataset$Enter_kv, pch=6,,main = "Percentage of Income Spent on Entertainment",col="Pink")
```



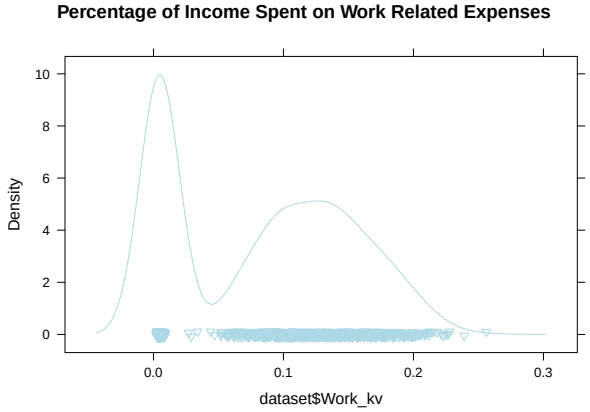

```
densityplot( ~ dataset$Edu_kv, pch=6,,main = "Percentage of Income Spent on Education",col="Maroon")
```



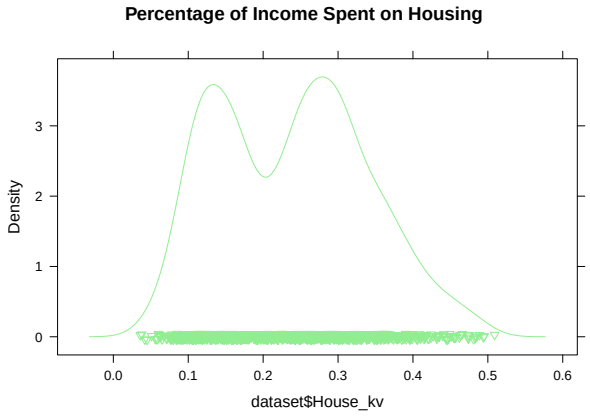

```
densityplot( ~ dataset$Trans_kv, pch=6,,main = "Percentage of Income Spent on Transporation",col="Red")
```



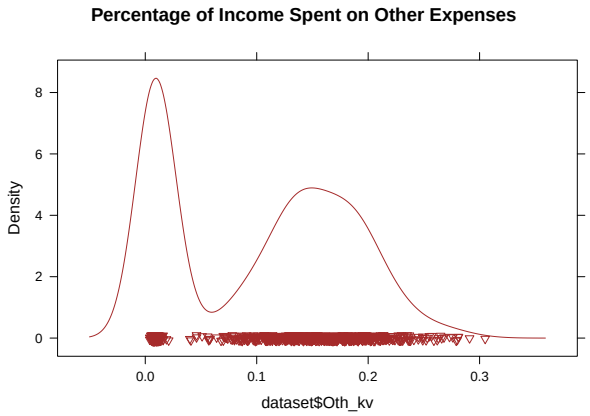

```
densityplot( ~ dataset$Work_kv, pch=6,,main = "Percentage of Income Spent on Work Related Expenses",col=
```




```
densityplot( ~ dataset$House_kv, pch=6,,main =  "Percentage of Income Spent on Housing",col="Light Green
```




```
densityplot( ~ dataset$Oth_kv, pch=6,,main = "Percentage of Income Spent on Other Expenses",col="Brown"
```



Analysis on Box Plot and Density Plot

From **Box Plot** it is very clear from boxplot that, there exists outliers in Percentage of Income Spent on Transporation and Entertainment. Lets Analyze a bit more by using density plot.

From **Density Plot**:

Using density plot we got to know that there is no need to remove the outliers.

General Observation When comapring the plots, a general observation is that most of the money is spent on Education.

Considering Exach Attribute separately:

1.Percentage of Income Spent on Food: * Graph is left skewed. * There exist two different peaks meaning there are two clusters. * Cluster one in betwwen 0.01 and 0.07 and the other cluster between 0.09 to 0.3. * Cluster 1 has the highest peak density.

2.Percentage of Income Spent on Entertainment: * Graph is some what uniformly distributed.

3.Percentage of Income Spent on Education: * Graph is some what uniformly distributed. * It has two distinct clusters one with the highest peak with density more than 3. * The other cluster with density almost equal to 1.8.

4.Percentage of Income Spent on Transportation: * Graph is right skewed.

5.Percentage of Income Spent on Work Related Expenses: * It two different peaks indicating two different clusters. * Cluster1 with highest peak has less number of observations. * Cluster 2 with the lower peak has many number observations.

6.Percentage of Income Spent on Housing: * Graph is some what normally distributed with a bend at 0.2 with density value 2.2 and then getting back to the peak density value 3.7.

7.Percentage of Income Spent on Other Expenses: * Graph is left skewed with two different peaks mentioning two clusters. * Cluster1 having the highest peak with density value 8.4 with lesser number of observations. * Cluster2 having peak density value 4.4 with many number of observations when compared to the other.

3. Clustering

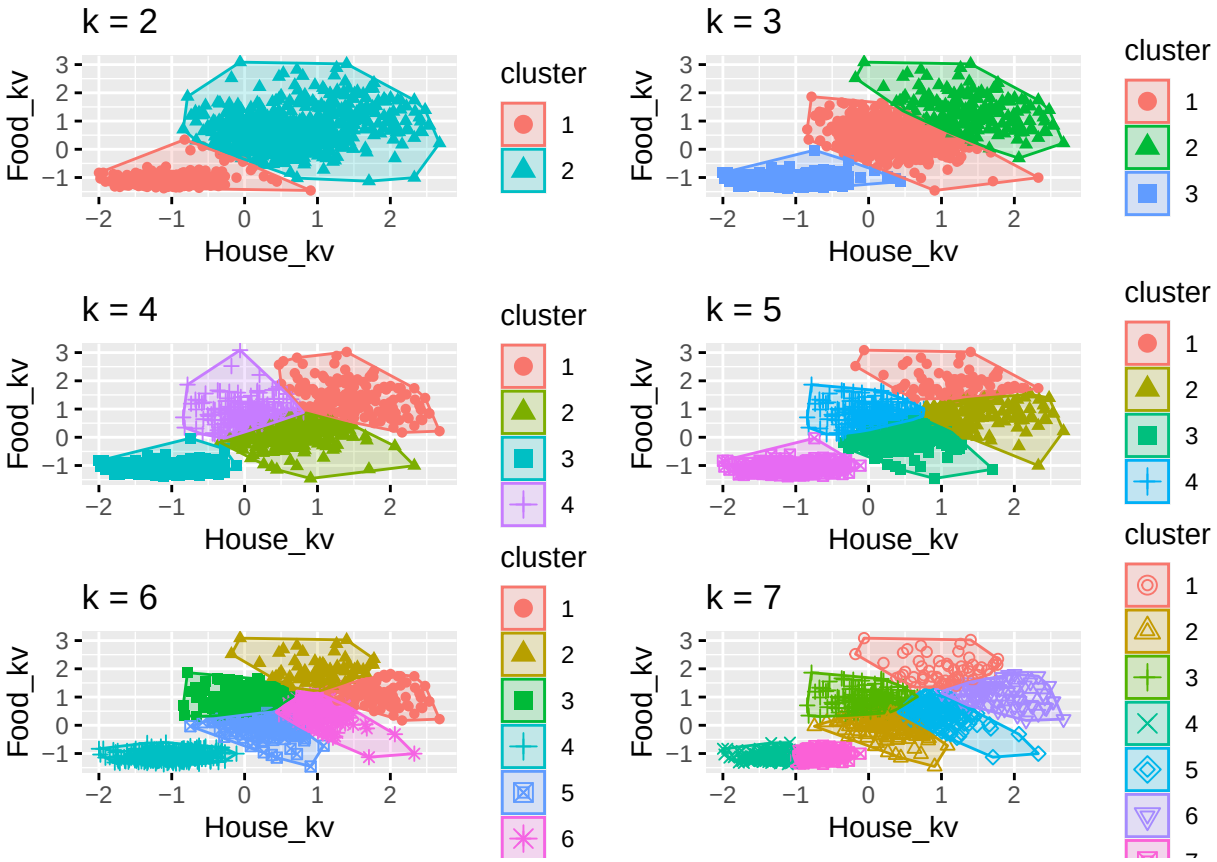
```
# Using the K-Means procedure as demonstrated in class, create clusters
# with k=2,3,4,5,6,7.
# You will be using only two variables as your centroids (House and Food)

# Create a dataframe for centroids
```

```r
centroids <- c("House_kv", "Food_kv")
df_centroid <- dataset1[,centroids]

################### Question 1: Create segmentation/cluster schemes for k=2,3,4,5,6,7.

# k = 2
kmeans_result_2 <- kmeans(df_centroid, centers = 2, nstart = 25)

# k = 3
kmeans_result_3 <- kmeans(df_centroid, centers = 3, nstart = 25)

# k = 4
kmeans_result_4 <- kmeans(df_centroid, centers = 4, nstart = 25)

# k = 5
kmeans_result_5 <- kmeans(df_centroid, centers = 5, nstart = 25)

# k = 6
kmeans_result_6 <- kmeans(df_centroid, centers = 6, nstart = 25)

# k = 7
kmeans_result_7 <- kmeans(df_centroid, centers = 7, nstart = 25)
```

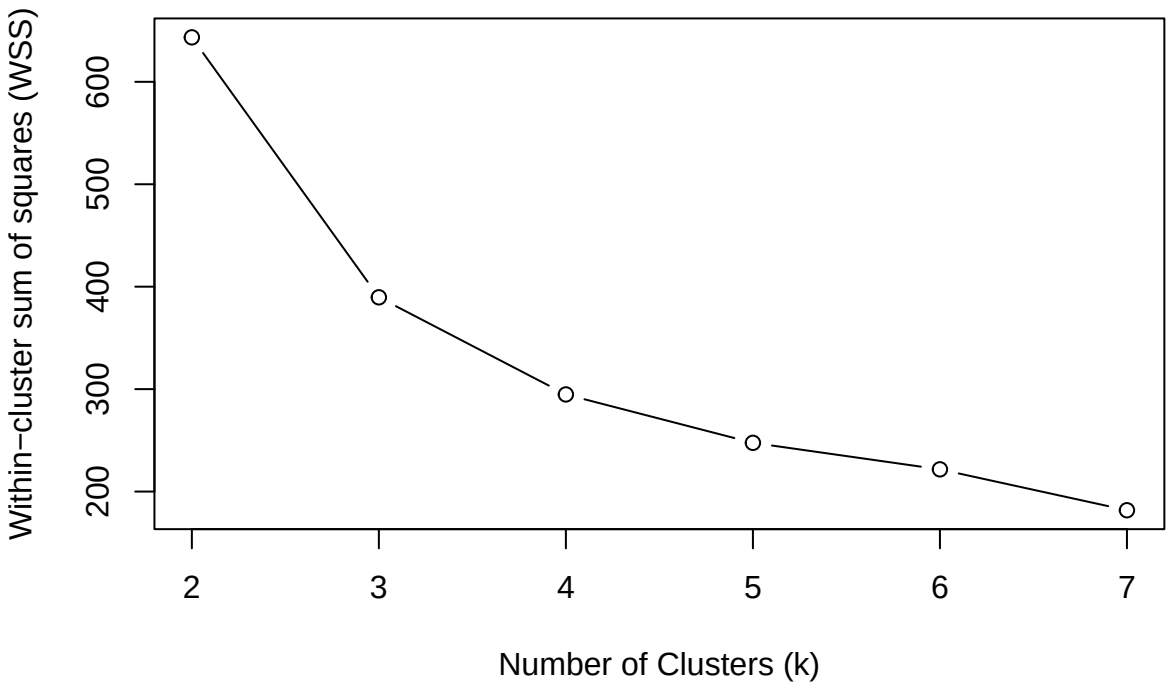
```r
# Visualize the results

library("factoextra")
```

```
## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

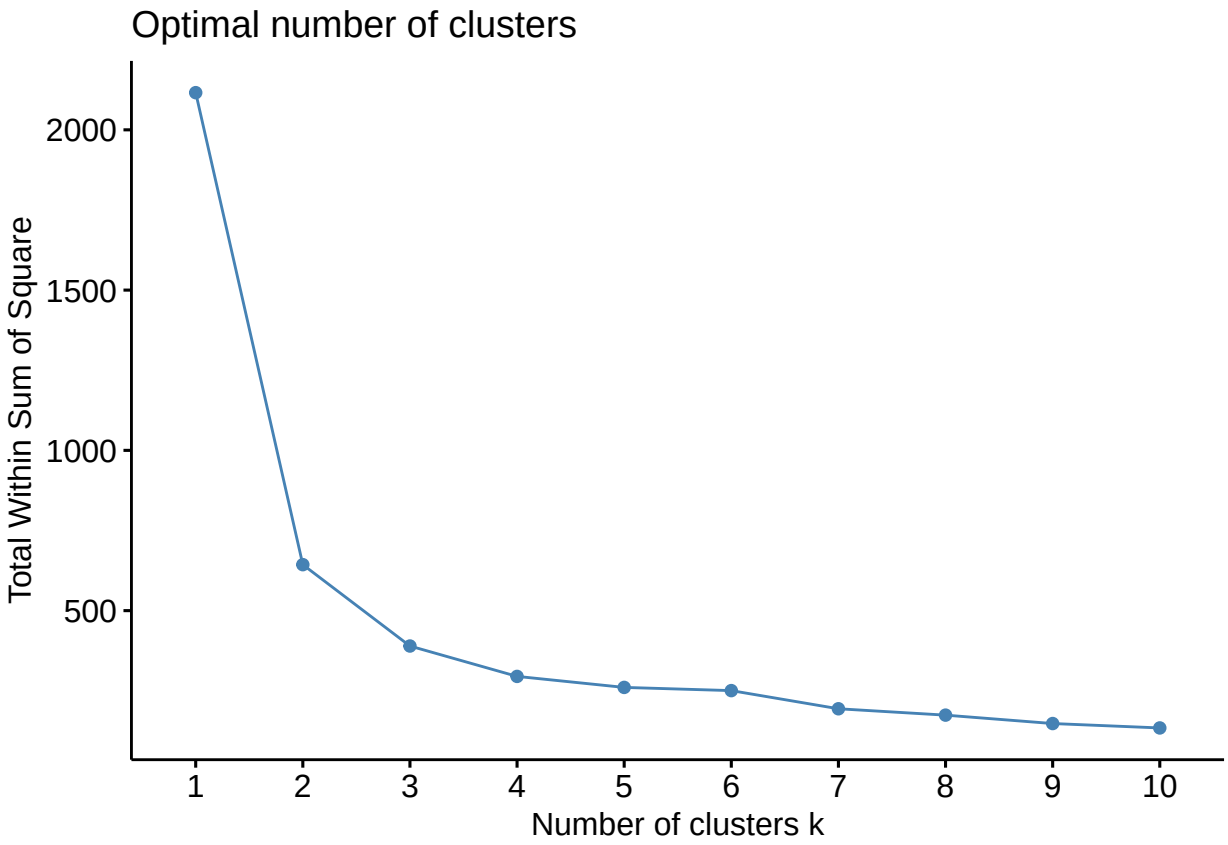
```r
p1 <- fviz_cluster(kmeans_result_2, geom = "point", data = df_centroid,ellipse = TRUE) + ggtitle("k = 2
p2 <- fviz_cluster(kmeans_result_3, geom = "point", data = df_centroid,ellipse = TRUE) + ggtitle("k = 3
p3 <- fviz_cluster(kmeans_result_4, geom = "point", data = df_centroid,ellipse = TRUE) + ggtitle("k = 4
p4 <- fviz_cluster(kmeans_result_5, geom = "point", data = df_centroid,ellipse = TRUE) + ggtitle("k = 5
p5 <- fviz_cluster(kmeans_result_6, geom = "point", data = df_centroid,ellipse = TRUE) + ggtitle("k = 6
p6 <- fviz_cluster(kmeans_result_7, geom = "point", data = df_centroid,ellipse = TRUE) + ggtitle("k = 7

library(gridExtra)
grid.arrange(p1, p2, p3, p4, p5, p6 ,nrow = 3)
```




```
################### Question 2:Create the WSS plots as demonstrated in class and select a suitable k
###################             value based on the "elbow". [Use the code that we discuss in the class.

library("cluster")
#function to compute total within-cluster sum of square
wss <- sapply(2:7, function(k) {
  kmeans_model <- kmeans(df_centroid, centers = k, nstart = 10)
  return(kmeans_model$tot.withinss)
})

# Plot the WSS Values
plot(2:7, wss, type = "b", xlab = "Number of Clusters (k)", ylab = "Within-cluster sum of squares (WSS)")
```




```
# Identify the elbow point
fviz_nbclust(df_centroid, kmeans, method = "wss")
```




```
print("The optimal number of clusters is 3, as thereafter there is no much change in variation.")
```

```
## [1] "The optimal number of clusters is 3, as thereafter there is no much change in variation."
```