Assign03

Shivam

2023-02-23

```r
#if(!is.null(dev.list())) dev.off()
#cat("\014")
#rm(list=ls())
options(scipen=9)
```

```r
if(!require(pastecs)){install.packages("pastecs")}
```

```
## Loading required package: pastecs
```

```r
library("pastecs")

if(!require(lattice)){install.packages("lattice")}
```

```
## Loading required package: lattice
```

```r
library("lattice")
```

```r
setwd("C:/Users/holys/OneDrive/Desktop/Data Analytics,Mathamatics,Algor/Assign03")

test_SJ <- read.table("PROG8430-23W-Assign03.txt", sep=",", header=TRUE)
    test_SJ <- as.data.frame(test_SJ)
    head(test_SJ)
```

```
##    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
```

```r
str(test_SJ)
```

```
## 'data.frame':    1059 obs. of  7 variables:
##  $ Food : num  0.043 0.123 0.043 0.119 0.122 0.084 0.089 0.03 0.134 0.035 ...
##  $ Enter: num  0.085 0.055 0.085 0.038 0.038 0.05 0.042 0.041 0.036 0.067 ...
##  $ Edu  : num  0.525 0.002 0.506 0.002 0.002 0.002 0.002 0.647 0.002 0.559 ...
##  $ Trans: num  0.18 0.169 0.193 0.301 0.225 0.285 0.215 0.156 0.281 0.168 ...
##  $ Work : num  0.005 0.121 0.006 0.139 0.095 0.079 0.204 0.003 0.141 0.003 ...
##  $ House: num  0.15 0.266 0.155 0.228 0.354 0.264 0.254 0.116 0.242 0.159 ...
##  $ Oth  : num  0.012 0.265 0.012 0.172 0.164 0.237 0.195 0.006 0.164 0.01 ...
```

1. Data Transformation

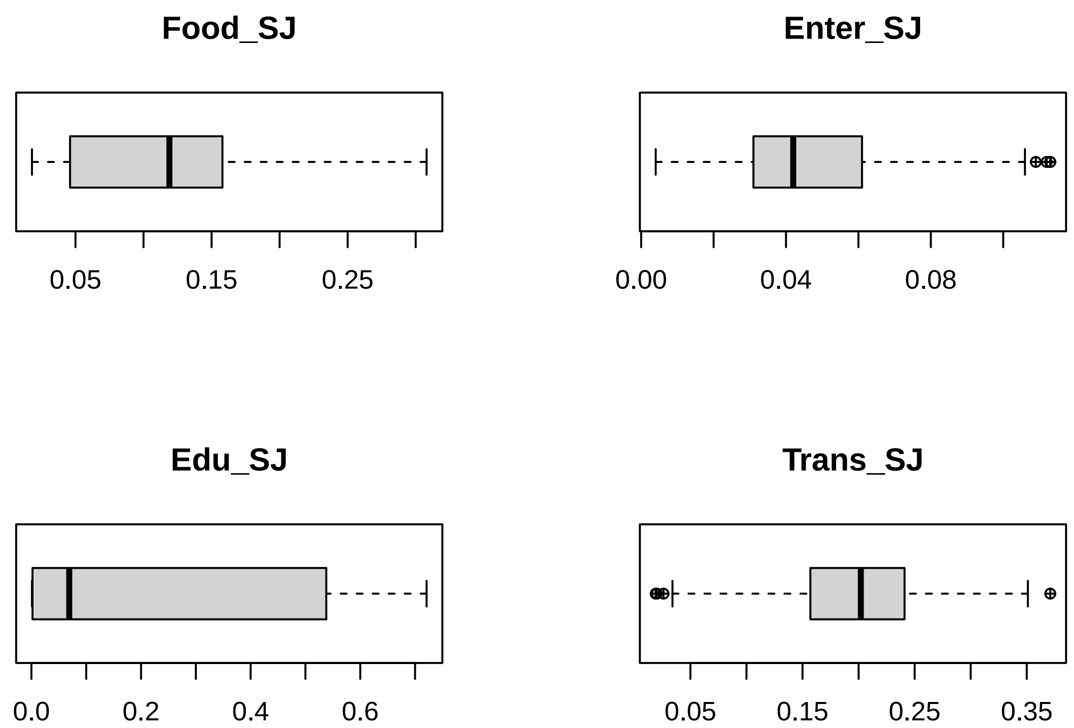
    1. Rename all variables with your initials appended (just as was done in assignment 1)

```
colnames(test_SJ) <- paste(colnames(test_SJ), "SJ", sep = "_")  # renaming all the
                                                                 # variables and adding
                                                                 # my initials SJ
  head(test_SJ)
```

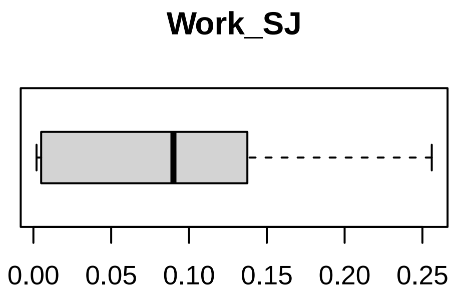
```
##   Food_SJ Enter_SJ Edu_SJ Trans_SJ Work_SJ House_SJ Oth_SJ
## 1   0.043    0.085  0.525    0.180   0.005    0.150  0.012
## 2   0.123    0.055  0.002    0.169   0.121    0.266  0.265
## 3   0.043    0.085  0.506    0.193   0.006    0.155  0.012
## 4   0.119    0.038  0.002    0.301   0.139    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
```

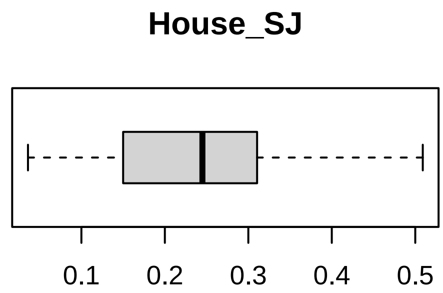
```
2. Standardize all of the variables using either of the two functions
 demonstrated in class. Describe why you chose the method you did.

-> Before standardization of any of the variable, first lets plot boxplot to
  observe the distribution of data.
```

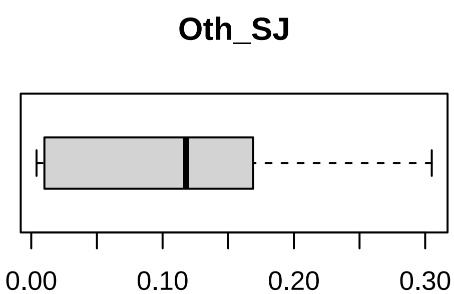
```
 par(mfrow=c(2,2))                                  #defining format of the boxplot

for (i in 1:ncol(test_SJ)) {                        # Generating box plot for each
                                                    # variable in th data set using
                                                    # for loop
  if (is.numeric(test_SJ[,i])) {
      boxplot(test_SJ[i], main= names(test_SJ)[i],
              horizontal=TRUE, pch=10)
  }
}
```




```
par(mfrow=c(1,1))
```







-> By observing and analyzing, it is interpreted that most of the data is not abnormal and very few vaiables out of the total, has a very little outliers.
In this case as the data has a very few outliers,I choose to perform min-max standardization formula for different variables. I chose this method as it preserves the range of data and this method is more suitable here as we do not want to get the data in N(0,1), rather we want it over (0,1).

```
#standardization function
Std_SJ <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}
```

```
test_SJ$Food_Std_SJ  <- Std_SJ(test_SJ$Food_SJ)   # standardizing each variable by
                                                  # passing it standardization function
test_SJ$Enter_Std_SJ <- Std_SJ(test_SJ$Enter_SJ)  # created above


test_SJ$Edu_Std_SJ   <- Std_SJ(test_SJ$Edu_SJ)

test_SJ$Trans_Std_SJ <- Std_SJ(test_SJ$Trans_SJ)

test_SJ$Work_Std_SJ  <- Std_SJ(test_SJ$Work_SJ)

test_SJ$House_Std_SJ <- Std_SJ(test_SJ$House_SJ)
```

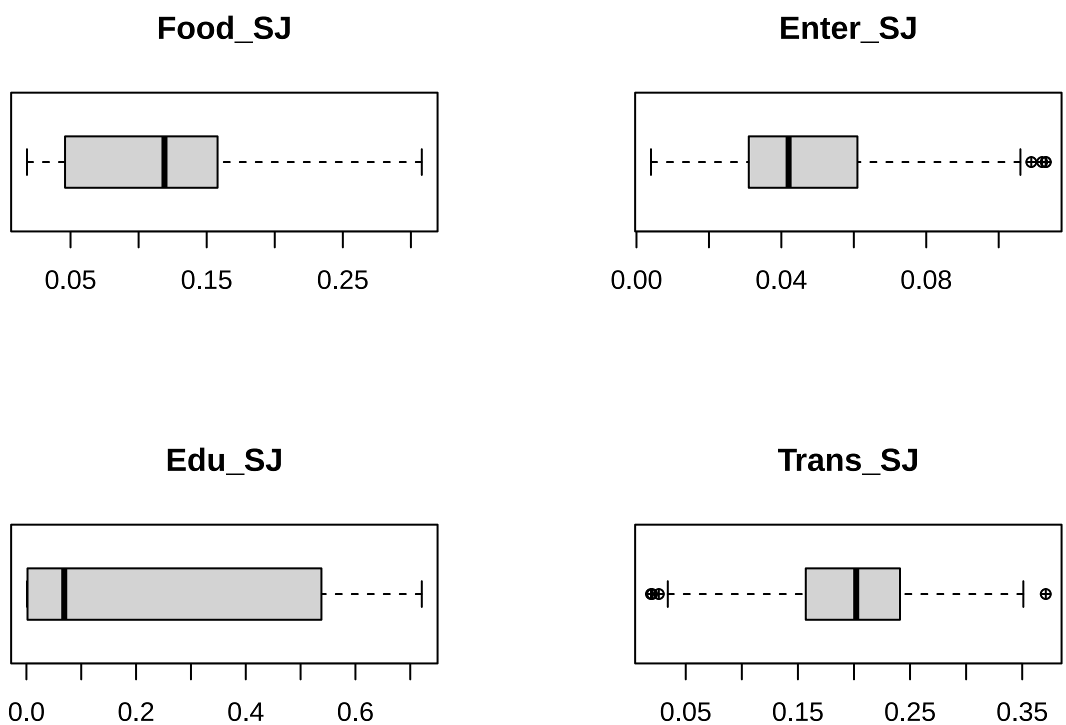
```r
test_SJ$Oth_Std_SJ   <- Std_SJ(test_SJ$Oth_SJ)

head(test_SJ)
```

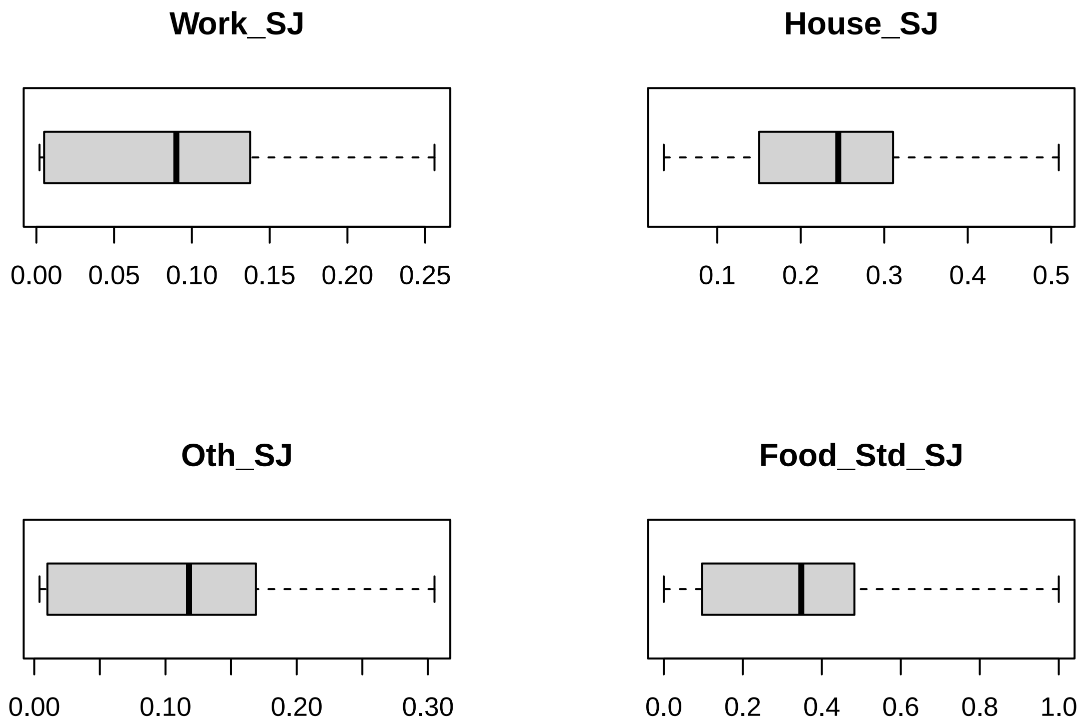
```
##   Food_SJ Enter_SJ Edu_SJ Trans_SJ Work_SJ House_SJ Oth_SJ Food_Std_SJ
## 1   0.043    0.085  0.525    0.180   0.005    0.150  0.012   0.0862069
## 2   0.123    0.055  0.002    0.169   0.121    0.266  0.265   0.3620690
## 3   0.043    0.085  0.506    0.193   0.006    0.155  0.012   0.0862069
## 4   0.119    0.038  0.002    0.301   0.139    0.228  0.172   0.3482759
## 5   0.122    0.038  0.002    0.225   0.095    0.354  0.164   0.3586207
## 6   0.084    0.050  0.002    0.285   0.079    0.264  0.237   0.2275862
##   Enter_Std_SJ  Edu_Std_SJ Trans_Std_SJ Work_Std_SJ House_Std_SJ Oth_Std_SJ
## 1    0.7431193 0.727777778    0.4573864  0.01181102    0.2410148 0.02657807
## 2    0.4678899 0.001388889    0.4261364  0.46850394    0.4862579 0.86710963
## 3    0.7431193 0.701388889    0.4943182  0.01574803    0.2515856 0.02657807
## 4    0.3119266 0.001388889    0.8011364  0.53937008    0.4059197 0.55813953
## 5    0.3119266 0.001388889    0.5852273  0.36614173    0.6723044 0.53156146
## 6    0.4220183 0.001388889    0.7556818  0.30314961    0.4820296 0.77408638
```

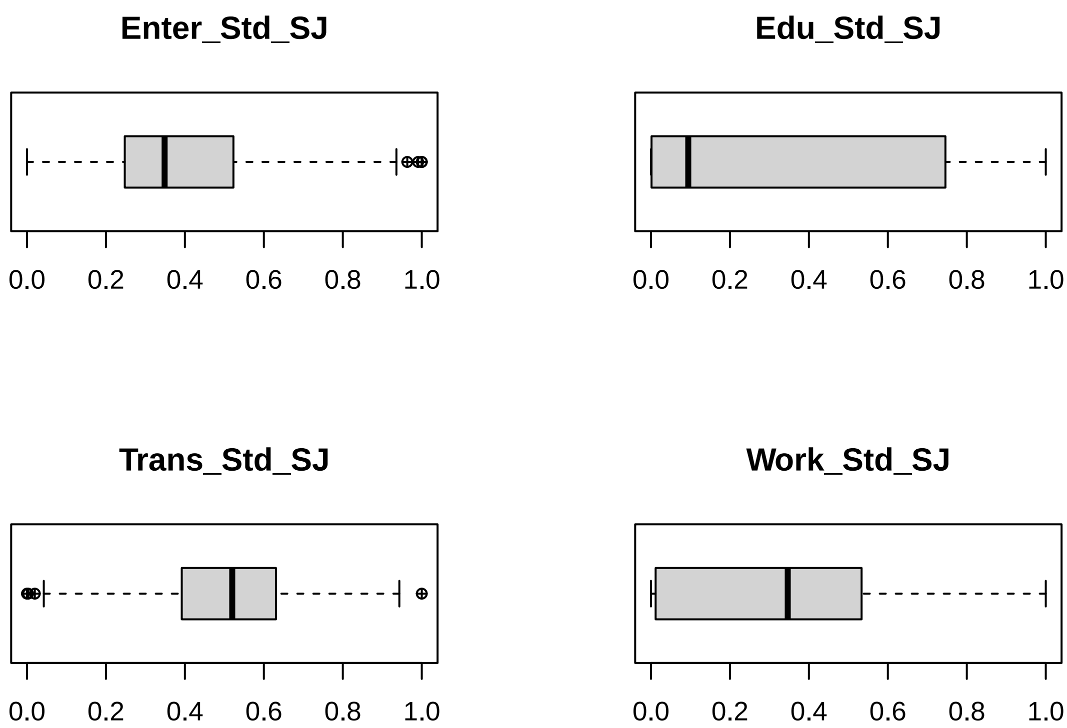
2 Descriptive Data Analysis

```
1. Create graphical summaries of the data (as demonstrated in class:
   boxplots, histograms or density plots) and comment on any
   observations you make.
  -> To graphically summarize the data, lets generate box plot:
```

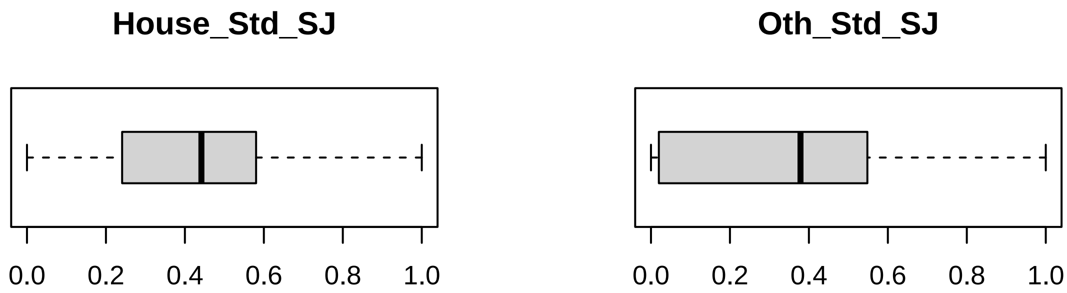
```r
par(mfrow=c(2,2))

for (i in 1:ncol(test_SJ)) {
  if (is.numeric(test_SJ[,i])) {
      boxplot(test_SJ[i], main=names(test_SJ)[i],
              horizontal=TRUE, pch=10)
  }
}                               # plotting the box plot of every variable
```


Food_SJ
0.05
0.15
0.25
Enter_SJ
0.00
0.04
0.08
Edu_SJ
0.0
0.2
0.4
0.6
Trans_SJ
0.05
0.15
0.25
0.35


Work_SJ
0.00 0.05 0.10 0.15 0.20 0.25
House_SJ
0.1 0.2 0.3 0.4 0.5
Oth_SJ
0.00 0.10 0.20 0.30
Food_Std_SJ
0.0 0.2 0.4 0.6 0.8 1.0

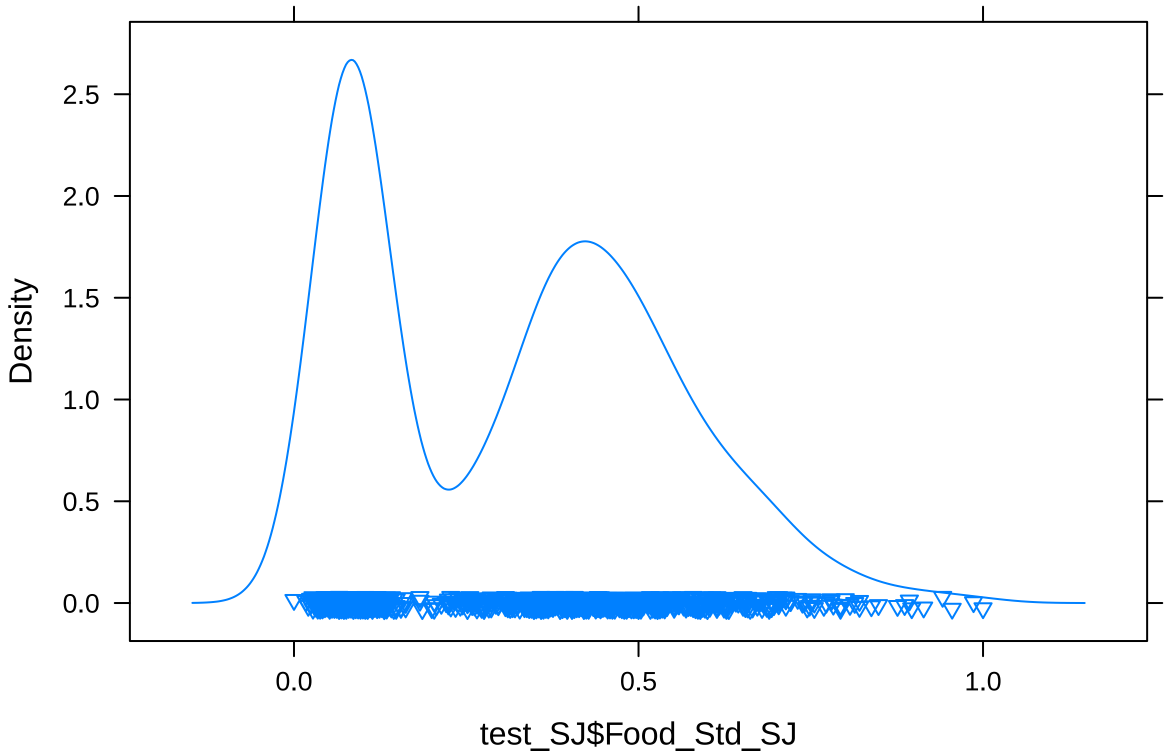



```
                                    # including the box plots of standardized
par(mfrow=c(1,1))                   # variables as well
```

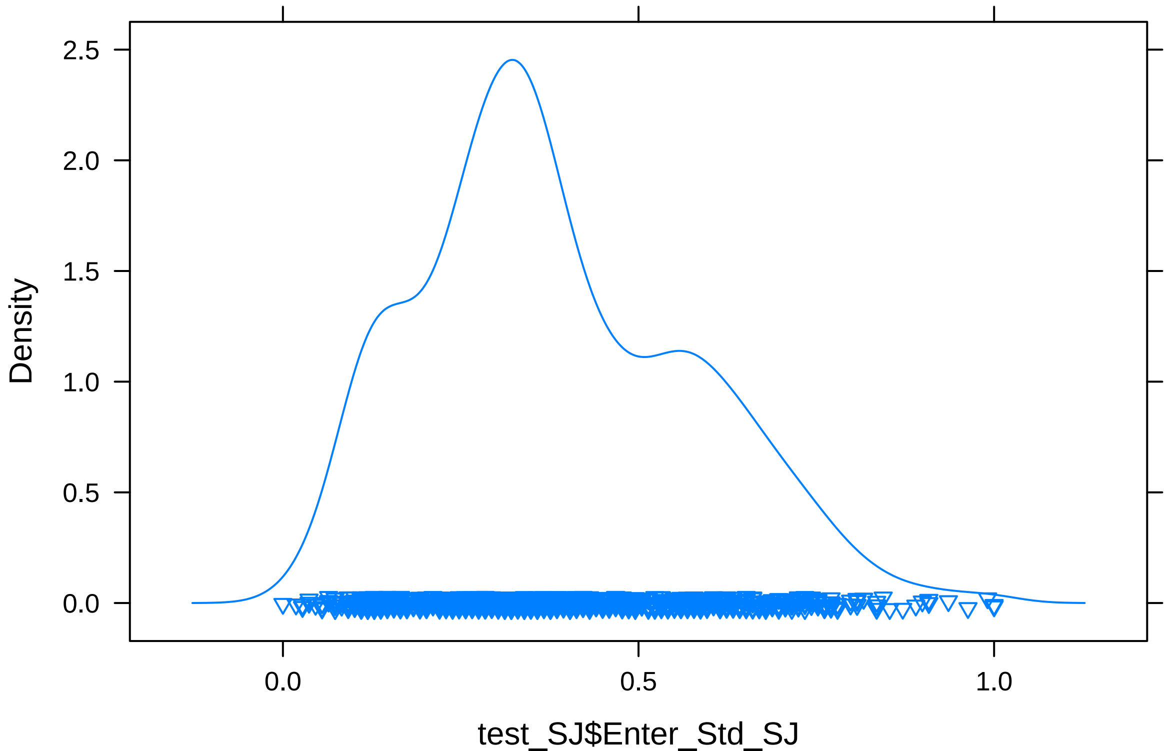


-> From the box plots, it is observed that all the variable have varried ranges and very few of them have a little outliers including Entertainment and transportation.

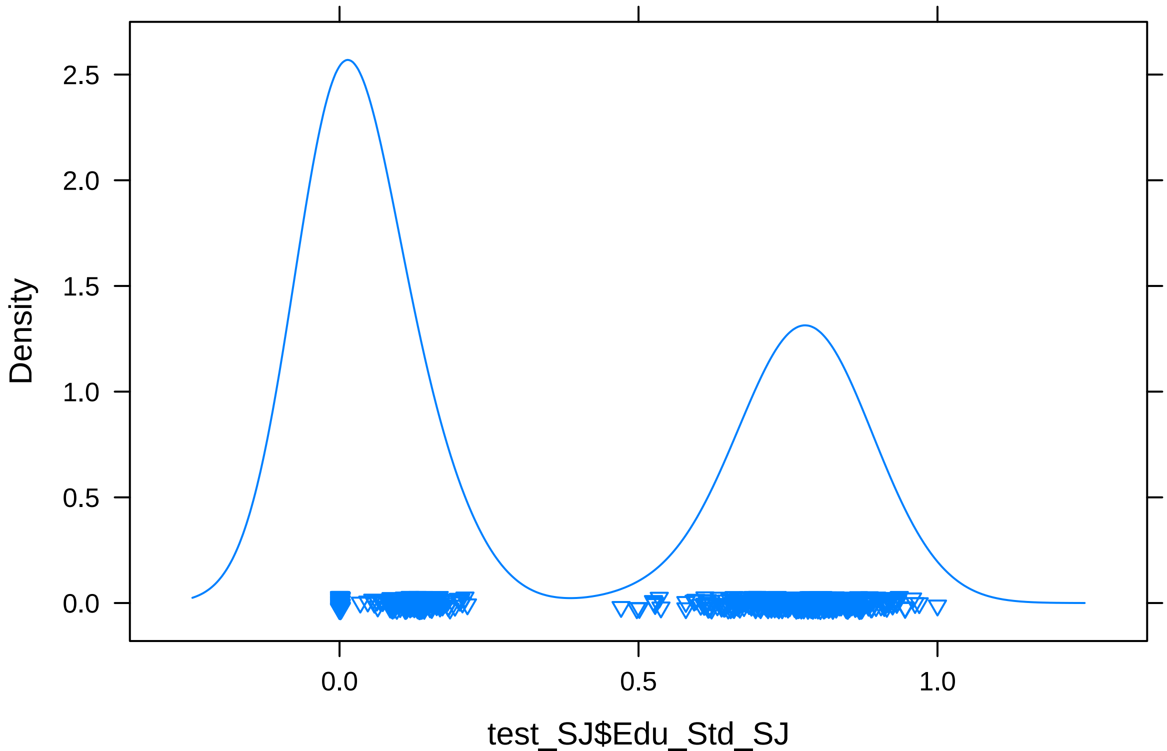
```
# plotting density plots for graphically
#analyzing data in detailed way of standardized variables
densityplot( ~ test_SJ$Food_Std_SJ, pch=6)
```



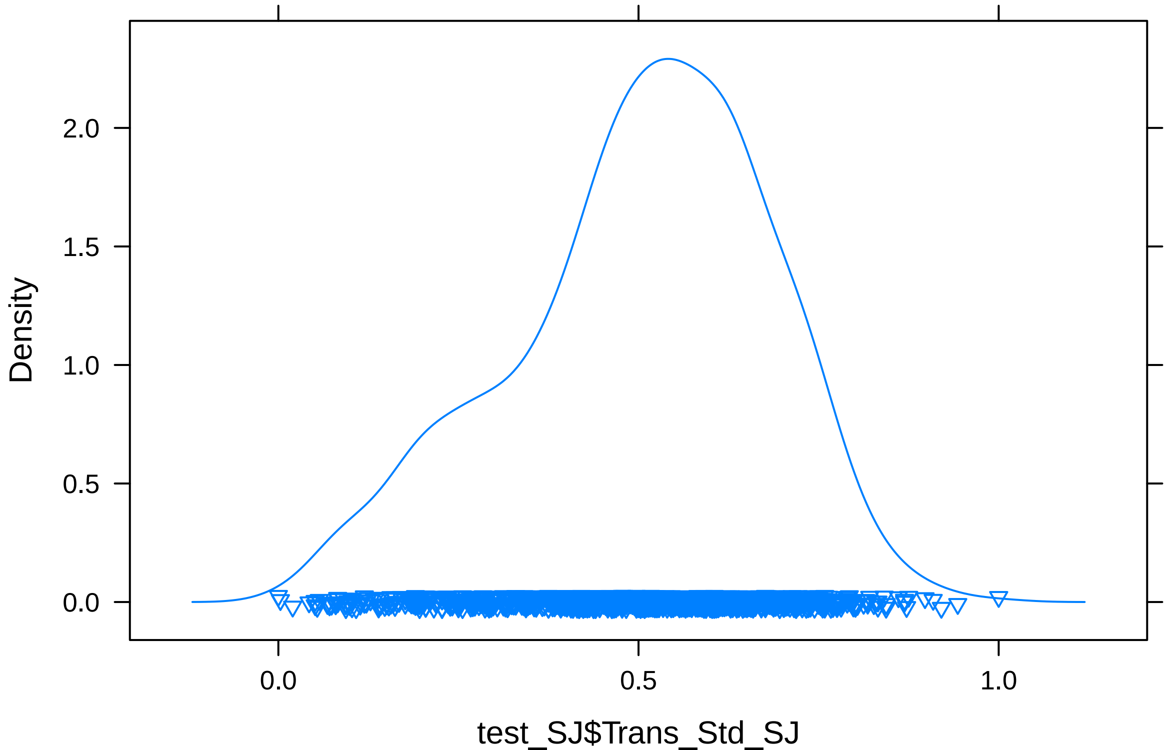

```
densityplot( ~ test_SJ$Enter_Std_SJ, pch=6)
```



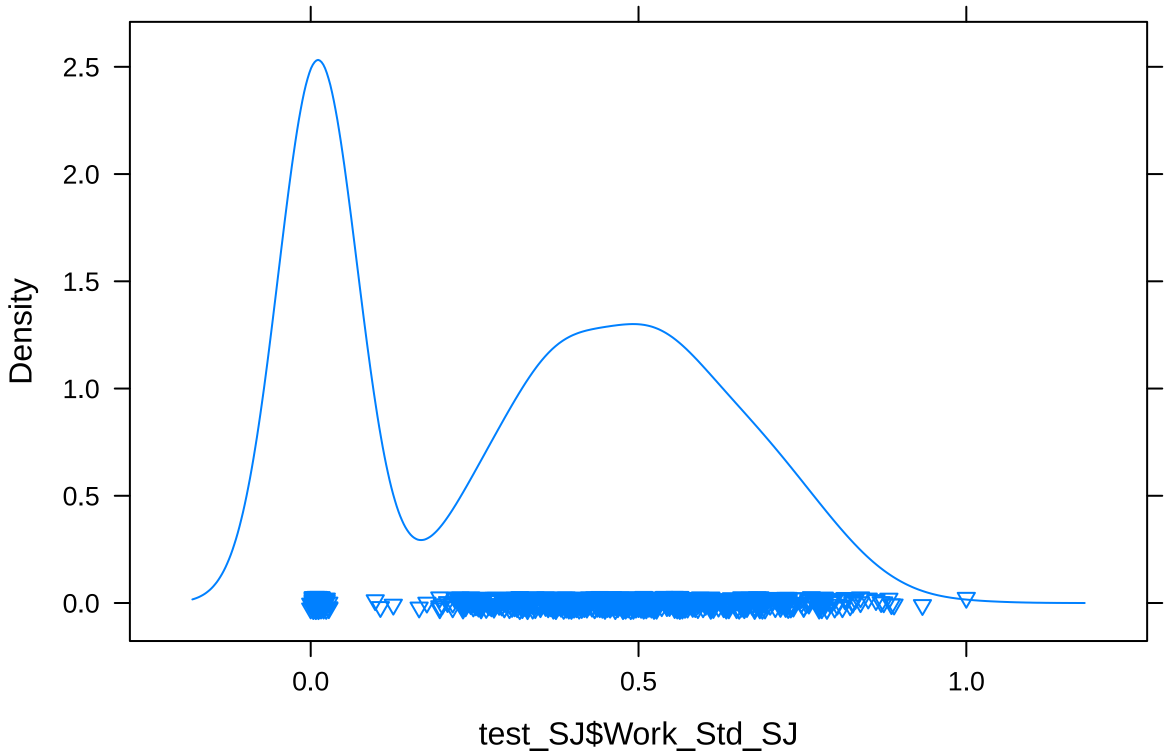

```
densityplot( ~ test_SJ$Edu_Std_SJ, pch=6)
```



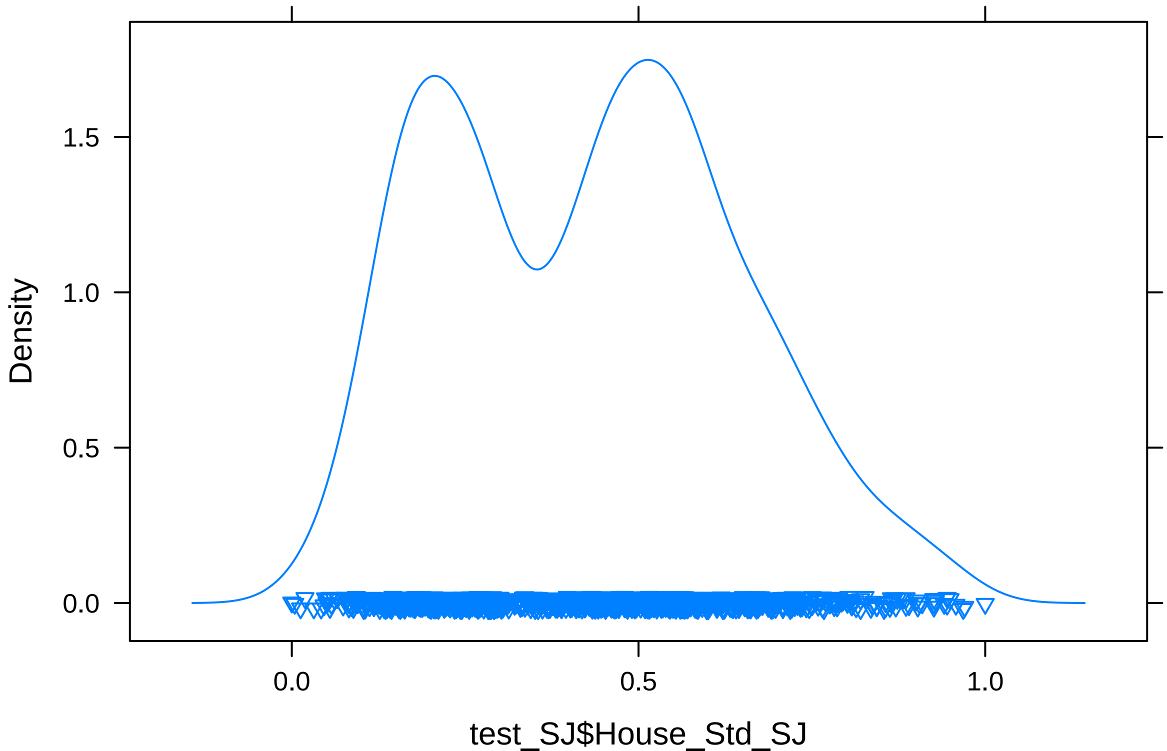

```
densityplot( ~ test_SJ$Trans_Std_SJ, pch=6)
```



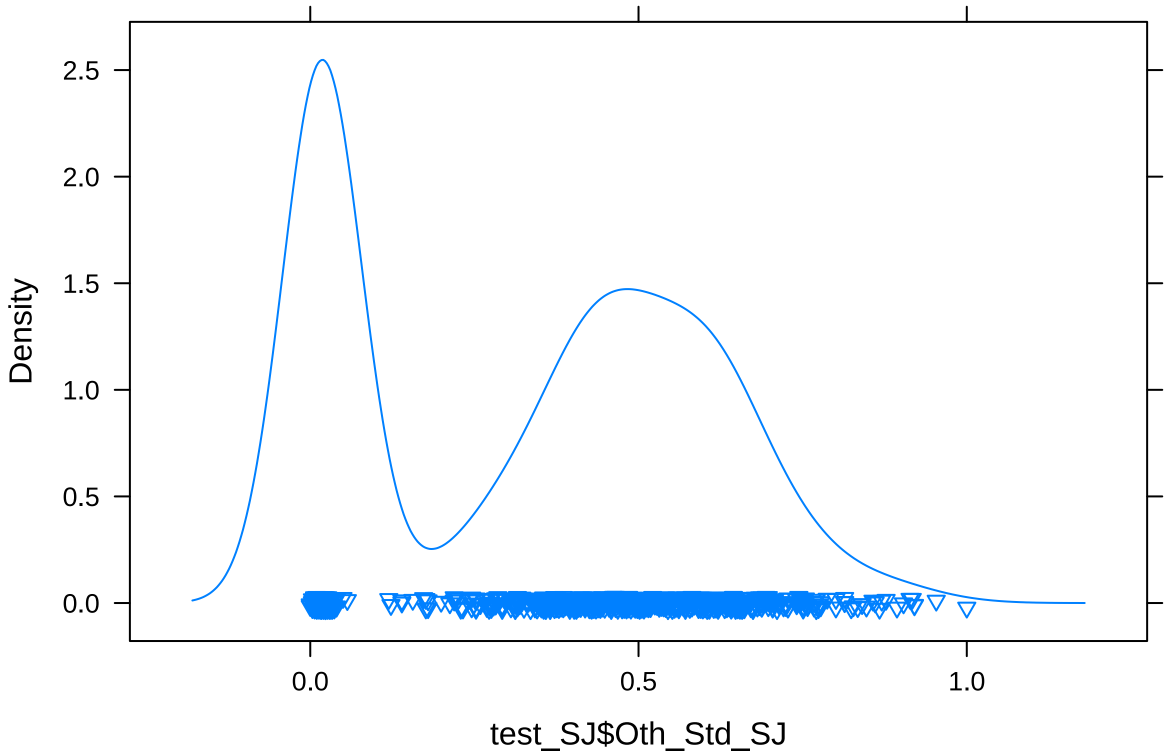

```
densityplot( ~ test_SJ$Work_Std_SJ, pch=6)
```




```
densityplot( ~ test_SJ$House_Std_SJ, pch=6)
```




```
densityplot( ~ test_SJ$Oth_Std_SJ, pch=6)
```



-> By observing the density plots, one can infer that in each the plots most of the data point are in the range and very few outliers.

---

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)

```
-> We have already standardized two variables House and Food as Food_Std_SJ
    and House_Std_SJ.
```

```
# these are the two variables Food and House in standardized form

centroids_for_plotting_SJ <- c(test_SJ$Food_Std_SJ,test_SJ$House_Std_SJ)

head(centroids_for_plotting_SJ)
```

```
## [1] 0.0862069 0.3620690 0.0862069 0.3482759 0.3586207 0.2275862
```

```
1. Create segmentation/cluster schemes for k=2,3,4,5,6,7.
->  Now I made variables below for generating elbow chart to identify and
    choose the value of K to perform K means clustering.
```

```r
# Creating Variable for Elbow Chart
# Trying for 2 to 7 Clusters
maxk_SJ <- 7
nk_SJ <- c(2:maxk_SJ)
wss_SJ <- rep(0,maxk_SJ-1)
```

```r
# As asked in the question,firstly I created clusters keeping the value of k=2

#Set Number of Clusters
k=2

Clstr_SJ <- kmeans(test_SJ[,c(8,13)], iter.max=10, centers=k, nstart=10)
Clstr_SJ$size
```

```
## [1] 417 642
```

```r
Clstr_SJ$centers
```

```
##   Food_Std_SJ House_Std_SJ
## 1  0.08833209    0.2114216
## 2  0.47218283    0.5681802
```

```r
Clstr_SJ$betweenss/Clstr_SJ$totss
```

```
## [1] 0.6967588
```

```r
test_SJ$cluster <- factor(Clstr_SJ$cluster)   # Adding Cluster tags to variables
head(test_SJ)
```

```
##   Food_SJ Enter_SJ Edu_SJ Trans_SJ Work_SJ House_SJ Oth_SJ Food_Std_SJ
## 1   0.043    0.085  0.525    0.180   0.005    0.150  0.012   0.0862069
## 2   0.123    0.055  0.002    0.169   0.121    0.266  0.265   0.3620690
## 3   0.043    0.085  0.506    0.193   0.006    0.155  0.012   0.0862069
## 4   0.119    0.038  0.002    0.301   0.139    0.228  0.172   0.3482759
## 5   0.122    0.038  0.002    0.225   0.095    0.354  0.164   0.3586207
## 6   0.084    0.050  0.002    0.285   0.079    0.264  0.237   0.2275862
##   Enter_Std_SJ  Edu_Std_SJ Trans_Std_SJ Work_Std_SJ House_Std_SJ Oth_Std_SJ
## 1    0.7431193 0.727777778    0.4573864  0.01181102    0.2410148 0.02657807
## 2    0.4678899 0.001388889    0.4261364  0.46850394    0.4862579 0.86710963
## 3    0.7431193 0.701388889    0.4943182  0.01574803    0.2515856 0.02657807
## 4    0.3119266 0.001388889    0.8011364  0.53937008    0.4059197 0.55813953
## 5    0.3119266 0.001388889    0.5852273  0.36614173    0.6723044 0.53156146
## 6    0.4220183 0.001388889    0.7556818  0.30314961    0.4820296 0.77408638
##   cluster
## 1       1
## 2       2
## 3       1
## 4       2
## 5       2
## 6       2
```

```r
centers_SJ <- data.frame(cluster=factor(1:k), Clstr_SJ$centers)

wss_SJ[k-1] <- Clstr_SJ$tot.withinss
```

```r
#Set Number of Clusters by putting the value of k as 3.
k=3

Clstr_SJ <- kmeans(test_SJ[,c(8,13)], iter.max=10, centers=k, nstart=10)
Clstr_SJ$size
```

```
## [1] 409 186 464
```

```r
Clstr_SJ$centers
```

```
##   Food_Std_SJ House_Std_SJ
## 1  0.08507714    0.2077981
## 2  0.61166110    0.7267499
## 3  0.41252229    0.5016585
```

```r
Clstr_SJ$betweenss/Clstr_SJ$totss
```

```
## [1] 0.8161152
```

```r
test_SJ$cluster <- factor(Clstr_SJ$cluster)   # Adding Cluster tags to variables
head(test_SJ)
```

```
##   Food_SJ Enter_SJ Edu_SJ Trans_SJ Work_SJ House_SJ Oth_SJ Food_Std_SJ
## 1   0.043    0.085  0.525    0.180   0.005    0.150  0.012   0.0862069
## 2   0.123    0.055  0.002    0.169   0.121    0.266  0.265   0.3620690
## 3   0.043    0.085  0.506    0.193   0.006    0.155  0.012   0.0862069
## 4   0.119    0.038  0.002    0.301   0.139    0.228  0.172   0.3482759
## 5   0.122    0.038  0.002    0.225   0.095    0.354  0.164   0.3586207
## 6   0.084    0.050  0.002    0.285   0.079    0.264  0.237   0.2275862
##   Enter_Std_SJ  Edu_Std_SJ Trans_Std_SJ Work_Std_SJ House_Std_SJ Oth_Std_SJ
## 1    0.7431193 0.727777778    0.4573864  0.01181102    0.2410148 0.02657807
## 2    0.4678899 0.001388889    0.4261364  0.46850394    0.4862579 0.86710963
## 3    0.7431193 0.701388889    0.4943182  0.01574803    0.2515856 0.02657807
## 4    0.3119266 0.001388889    0.8011364  0.53937008    0.4059197 0.55813953
## 5    0.3119266 0.001388889    0.5852273  0.36614173    0.6723044 0.53156146
## 6    0.4220183 0.001388889    0.7556818  0.30314961    0.4820296 0.77408638
##   cluster
## 1       1
## 2       3
## 3       1
## 4       3
## 5       3
## 6       3
```

```r
centers_SJ <- data.frame(cluster=factor(1:k), Clstr_SJ$centers)

wss_SJ[k-1] <- Clstr_SJ$tot.withinss
```

```r
#Setting the value of k as 4 and creating clusters
k=4

Clstr_SJ <- kmeans(test_SJ[,c(8,13)], iter.max=10, centers=k, nstart=10)
Clstr_SJ$size
```

```
## [1] 165 228 262 404
```

```r
Clstr_SJ$centers
```

```
##   Food_Std_SJ House_Std_SJ
## 1   0.6252038    0.7368057
## 2   0.4941168    0.4409054
## 3   0.3427086    0.5665155
## 4   0.0850717    0.2039803
```

```r
Clstr_SJ$betweenss/Clstr_SJ$totss
```

```
## [1] 0.8609939
```

```r
test_SJ$cluster <- factor(Clstr_SJ$cluster)   # Adding Cluster tags to variables
head(test_SJ)
```

```
##   Food_SJ Enter_SJ Edu_SJ Trans_SJ Work_SJ House_SJ Oth_SJ Food_Std_SJ
## 1   0.043    0.085  0.525    0.180   0.005    0.150  0.012   0.0862069
## 2   0.123    0.055  0.002    0.169   0.121    0.266  0.265   0.3620690
## 3   0.043    0.085  0.506    0.193   0.006    0.155  0.012   0.0862069
## 4   0.119    0.038  0.002    0.301   0.139    0.228  0.172   0.3482759
## 5   0.122    0.038  0.002    0.225   0.095    0.354  0.164   0.3586207
## 6   0.084    0.050  0.002    0.285   0.079    0.264  0.237   0.2275862
##   Enter_Std_SJ  Edu_Std_SJ Trans_Std_SJ Work_Std_SJ House_Std_SJ Oth_Std_SJ
## 1    0.7431193 0.727777778    0.4573864  0.01181102    0.2410148 0.02657807
## 2    0.4678899 0.001388889    0.4261364  0.46850394    0.4862579 0.86710963
## 3    0.7431193 0.701388889    0.4943182  0.01574803    0.2515856 0.02657807
## 4    0.3119266 0.001388889    0.8011364  0.53937008    0.4059197 0.55813953
## 5    0.3119266 0.001388889    0.5852273  0.36614173    0.6723044 0.53156146
## 6    0.4220183 0.001388889    0.7556818  0.30314961    0.4820296 0.77408638
##   cluster
## 1       4
## 2       3
## 3       4
## 4       2
## 5       3
## 6       3
```

```r
centers_SJ <- data.frame(cluster=factor(1:k), Clstr_SJ$centers)

wss_SJ[k-1] <- Clstr_SJ$tot.withinss
```

```
# For value of k as 5
k=5

Clstr_SJ <- kmeans(test_SJ[,c(8,13)], iter.max=10, centers=k, nstart=10)
Clstr_SJ$size
```

```
## [1] 214 230 403  75 137
```

```
Clstr_SJ$centers
```

```
##   Food_Std_SJ House_Std_SJ
## 1  0.49444086    0.4403983
## 2  0.33134933    0.5375402
## 3  0.08470951    0.2035736
## 4  0.74013793    0.6800282
## 5  0.49859049    0.7446181
```

```
Clstr_SJ$betweenss/Clstr_SJ$totss
```

```
## [1] 0.8836823
```

```
test_SJ$cluster <- factor(Clstr_SJ$cluster)   # Adding Cluster tags to variables
head(test_SJ)
```

```
##   Food_SJ Enter_SJ Edu_SJ Trans_SJ Work_SJ House_SJ Oth_SJ Food_Std_SJ
## 1   0.043    0.085  0.525    0.180   0.005    0.150  0.012   0.0862069
## 2   0.123    0.055  0.002    0.169   0.121    0.266  0.265   0.3620690
## 3   0.043    0.085  0.506    0.193   0.006    0.155  0.012   0.0862069
## 4   0.119    0.038  0.002    0.301   0.139    0.228  0.172   0.3482759
## 5   0.122    0.038  0.002    0.225   0.095    0.354  0.164   0.3586207
## 6   0.084    0.050  0.002    0.285   0.079    0.264  0.237   0.2275862
##   Enter_Std_SJ  Edu_Std_SJ Trans_Std_SJ Work_Std_SJ House_Std_SJ Oth_Std_SJ
## 1    0.7431193 0.727777778    0.4573864  0.01181102    0.2410148 0.02657807
## 2    0.4678899 0.001388889    0.4261364  0.46850394    0.4862579 0.86710963
## 3    0.7431193 0.701388889    0.4943182  0.01574803    0.2515856 0.02657807
## 4    0.3119266 0.001388889    0.8011364  0.53937008    0.4059197 0.55813953
## 5    0.3119266 0.001388889    0.5852273  0.36614173    0.6723044 0.53156146
## 6    0.4220183 0.001388889    0.7556818  0.30314961    0.4820296 0.77408638
##   cluster
## 1       3
## 2       2
## 3       3
## 4       2
## 5       2
## 6       2
```

```
centers_SJ <- data.frame(cluster=factor(1:k), Clstr_SJ$centers)

wss_SJ[k-1] <- Clstr_SJ$tot.withinss
```

```r
# For value of k=6
k=6

Clstr_SJ <- kmeans(test_SJ[,c(8,13)], iter.max=10, centers=k, nstart=10)
Clstr_SJ$size
```

```
## [1]  62  80 403 155 194 165
```

```r
Clstr_SJ$centers
```

```
##    Food_Std_SJ House_Std_SJ
## 1  0.75194661    0.6488781
## 2  0.56607759    0.8194767
## 3  0.08470951    0.2035736
## 4  0.44129032    0.6184273
## 5  0.49335229    0.4271703
## 6  0.30194357    0.5235057
```

```r
Clstr_SJ$betweenss/Clstr_SJ$totss
```

```
## [1] 0.8992075
```

```r
test_SJ$cluster <- factor(Clstr_SJ$cluster)   # Adding Cluster tags to variables
head(test_SJ)
```

```
##   Food_SJ Enter_SJ Edu_SJ Trans_SJ Work_SJ House_SJ Oth_SJ Food_Std_SJ
## 1   0.043    0.085  0.525    0.180   0.005    0.150  0.012   0.0862069
## 2   0.123    0.055  0.002    0.169   0.121    0.266  0.265   0.3620690
## 3   0.043    0.085  0.506    0.193   0.006    0.155  0.012   0.0862069
## 4   0.119    0.038  0.002    0.301   0.139    0.228  0.172   0.3482759
## 5   0.122    0.038  0.002    0.225   0.095    0.354  0.164   0.3586207
## 6   0.084    0.050  0.002    0.285   0.079    0.264  0.237   0.2275862
##   Enter_Std_SJ  Edu_Std_SJ Trans_Std_SJ Work_Std_SJ House_Std_SJ Oth_Std_SJ
## 1    0.7431193 0.727777778    0.4573864  0.01181102    0.2410148 0.02657807
## 2    0.4678899 0.001388889    0.4261364  0.46850394    0.4862579 0.86710963
## 3    0.7431193 0.701388889    0.4943182  0.01574803    0.2515856 0.02657807
## 4    0.3119266 0.001388889    0.8011364  0.53937008    0.4059197 0.55813953
## 5    0.3119266 0.001388889    0.5852273  0.36614173    0.6723044 0.53156146
## 6    0.4220183 0.001388889    0.7556818  0.30314961    0.4820296 0.77408638
##   cluster
## 1       3
## 2       6
## 3       3
## 4       6
## 5       4
## 6       6
```

```r
centers_SJ <- data.frame(cluster=factor(1:k), Clstr_SJ$centers)

wss_SJ[k-1] <- Clstr_SJ$tot.withinss
```

```
#Set Number of Clusters as k= 7
k=7

Clstr_SJ <- kmeans(test_SJ[,c(8,13)], iter.max=10, centers=k, nstart=10)
Clstr_SJ$size
```

```
## [1] 178 156 171  91 190 212  61
```

```
Clstr_SJ$centers
```

```
##   Food_Std_SJ House_Std_SJ
## 1  0.50896939    0.4347103
## 2  0.31279841    0.4835475
## 3  0.40619076    0.6183623
## 4  0.56411520    0.8033362
## 5  0.08767695    0.2672193
## 6  0.08096942    0.1462264
## 7  0.75353307    0.6476623
```

```
Clstr_SJ$betweenss/Clstr_SJ$totss
```

```
## [1] 0.9141358
```

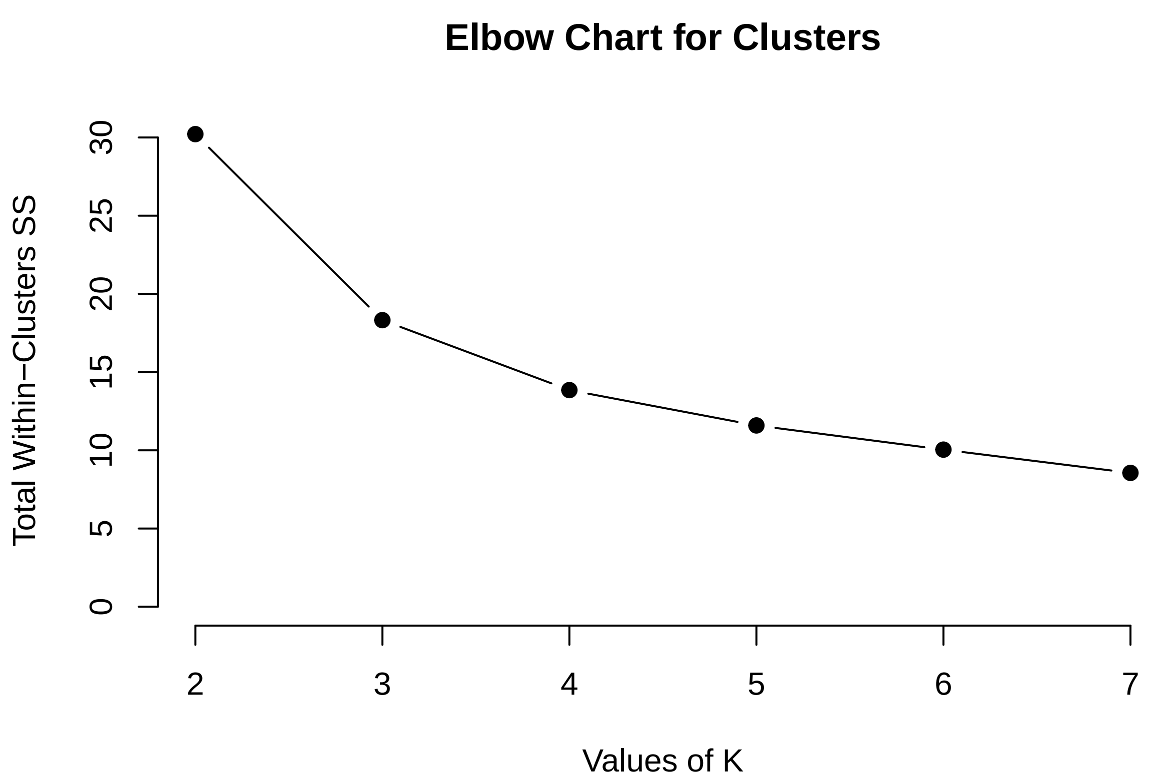
```
test_SJ$cluster <- factor(Clstr_SJ$cluster)   # Adding Cluster tags to variables
head(test_SJ)
```

```
##   Food_SJ Enter_SJ Edu_SJ Trans_SJ Work_SJ House_SJ Oth_SJ Food_Std_SJ
## 1   0.043    0.085  0.525    0.180   0.005    0.150  0.012   0.0862069
## 2   0.123    0.055  0.002    0.169   0.121    0.266  0.265   0.3620690
## 3   0.043    0.085  0.506    0.193   0.006    0.155  0.012   0.0862069
## 4   0.119    0.038  0.002    0.301   0.139    0.228  0.172   0.3482759
## 5   0.122    0.038  0.002    0.225   0.095    0.354  0.164   0.3586207
## 6   0.084    0.050  0.002    0.285   0.079    0.264  0.237   0.2275862
##   Enter_Std_SJ  Edu_Std_SJ Trans_Std_SJ Work_Std_SJ House_Std_SJ Oth_Std_SJ
## 1    0.7431193 0.727777778    0.4573864  0.01181102    0.2410148 0.02657807
## 2    0.4678899 0.001388889    0.4261364  0.46850394    0.4862579 0.86710963
## 3    0.7431193 0.701388889    0.4943182  0.01574803    0.2515856 0.02657807
## 4    0.3119266 0.001388889    0.8011364  0.53937008    0.4059197 0.55813953
## 5    0.3119266 0.001388889    0.5852273  0.36614173    0.6723044 0.53156146
## 6    0.4220183 0.001388889    0.7556818  0.30314961    0.4820296 0.77408638
##   cluster
## 1       5
## 2       2
## 3       5
## 4       2
## 5       3
## 6       2
```

```
centers_SJ <- data.frame(cluster=factor(1:k), Clstr_SJ$centers)

wss_SJ[k-1] <- Clstr_SJ$tot.withinss
```

```
2. Create the WSS plots as demonstrated in class and select a
   suitable k value based on the "elbow". [NOTE - Use the code
   that I provided to do this. Using other functions will yield
   different results.

  -> As thought in the class, and taking the reference form the example provided,
   lets make an elbow plot to choose the value of k.
```

```
# Plotting the elbow chart after trying the value of k=2,3,4,5,6,7
plot(2:maxk_SJ, wss_SJ,
     type="b", pch = 19, frame = FALSE,
     main="Elbow Chart for Clusters",
     xlab="Values of K",
     ylab="Total Within-Clusters SS",
     ylim=c(0,max(wss_SJ)))
```



-> As observed from the elbow plot,it is clearly observed that the bend is observed when the value of k is 3. Hence, value of k=3.

---

4. Evaluation of Clusters
   1. Based on the "k" chosen above, create a scatter plot showing the clusters and colour-coded data points for each of "k-1", "k", "k+1". For example, if you think the "elbow" is at k=4 create the charts for k=3, k=4 and k=5.

-> As value of k is 3, lets make scatter plot for k=2, 3 and 4.

```r
k=2

Clstr_SJ <- kmeans(test_SJ[,c(8,13)], iter.max=10, centers=k, nstart=10)
Clstr_SJ$size
```

```
## [1] 417 642
```

```r
Clstr_SJ$centers
```

```
##   Food_Std_SJ House_Std_SJ
## 1  0.08833209    0.2114216
## 2  0.47218283    0.5681802
```

```r
Clstr_SJ$betweenss/Clstr_SJ$totss
```

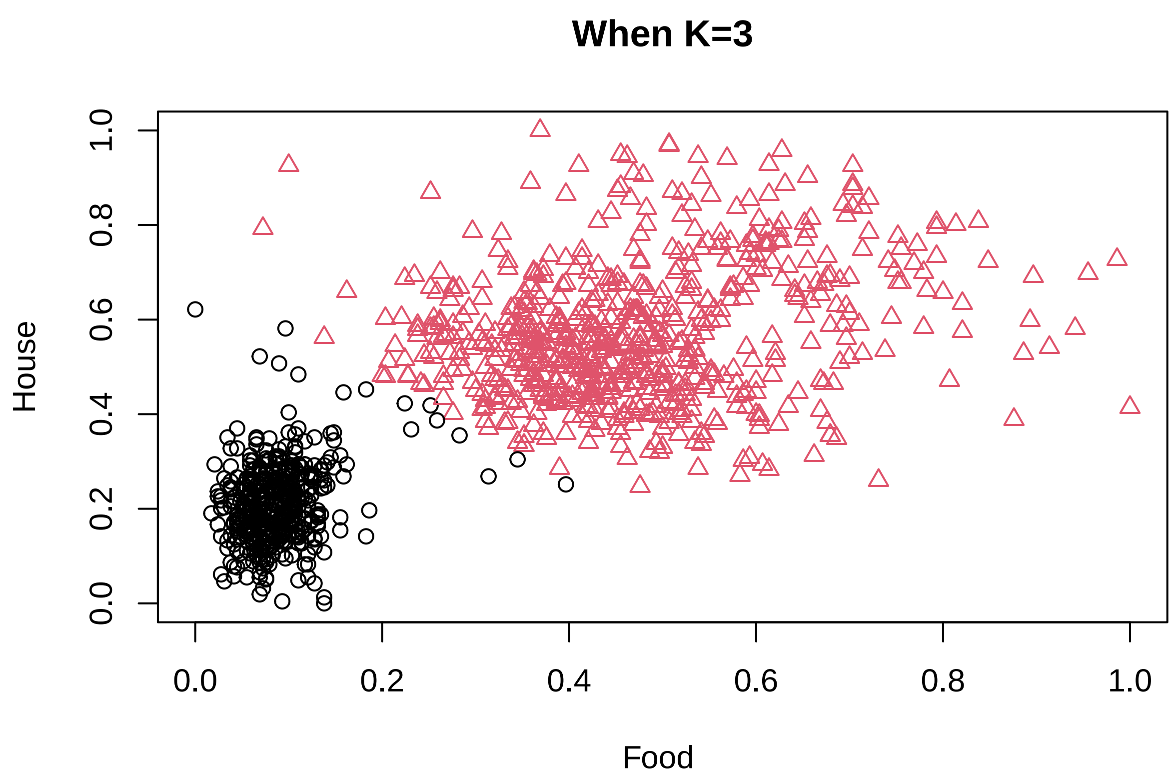
```
## [1] 0.6967588
```

```r
test_SJ$cluster <- factor(Clstr_SJ$cluster)
head(test_SJ)
```

```
##   Food_SJ Enter_SJ Edu_SJ Trans_SJ Work_SJ House_SJ Oth_SJ Food_Std_SJ
## 1   0.043    0.085  0.525    0.180   0.005    0.150  0.012   0.0862069
## 2   0.123    0.055  0.002    0.169   0.121    0.266  0.265   0.3620690
## 3   0.043    0.085  0.506    0.193   0.006    0.155  0.012   0.0862069
## 4   0.119    0.038  0.002    0.301   0.139    0.228  0.172   0.3482759
## 5   0.122    0.038  0.002    0.225   0.095    0.354  0.164   0.3586207
## 6   0.084    0.050  0.002    0.285   0.079    0.264  0.237   0.2275862
##   Enter_Std_SJ  Edu_Std_SJ Trans_Std_SJ Work_Std_SJ House_Std_SJ Oth_Std_SJ
## 1    0.7431193 0.727777778    0.4573864  0.01181102    0.2410148 0.02657807
## 2    0.4678899 0.001388889    0.4261364  0.46850394    0.4862579 0.86710963
## 3    0.7431193 0.701388889    0.4943182  0.01574803    0.2515856 0.02657807
## 4    0.3119266 0.001388889    0.8011364  0.53937008    0.4059197 0.55813953
## 5    0.3119266 0.001388889    0.5852273  0.36614173    0.6723044 0.53156146
## 6    0.4220183 0.001388889    0.7556818  0.30314961    0.4820296 0.77408638
##   cluster
## 1       1
## 2       2
## 3       1
## 4       2
## 5       2
## 6       2
```

```r
centers_SJ <- data.frame(cluster=factor(1:k), Clstr_SJ$centers)

wss_SJ[k-1] <- Clstr_SJ$tot.withinss

plot(test_SJ$Food_Std_SJ, test_SJ$House_Std_SJ, #ploting scatter plot for analyzing
     col=test_SJ$cluster,
      main=" When K=3 ",
```

```
     xlab="Food ",
     ylab="House",
     pch=as.numeric(test_SJ$cluster)) # when k= 2
points(centers_SJ$Food_Std_SJ, centers_SJ$House_Std_SJ,
       col=centers_SJ$cluster, pch=as.numeric(centers_SJ$cluster),
       cex=3, lwd=3)
```




```
k=3

Clstr_SJ <- kmeans(test_SJ[,c(8,13)], iter.max=10, centers=k, nstart=10)
Clstr_SJ$size
```

```
## [1] 464 186 409
```

```
Clstr_SJ$centers
```

```
##   Food_Std_SJ House_Std_SJ
## 1  0.41252229    0.5016585
## 2  0.61166110    0.7267499
## 3  0.08507714    0.2077981
```

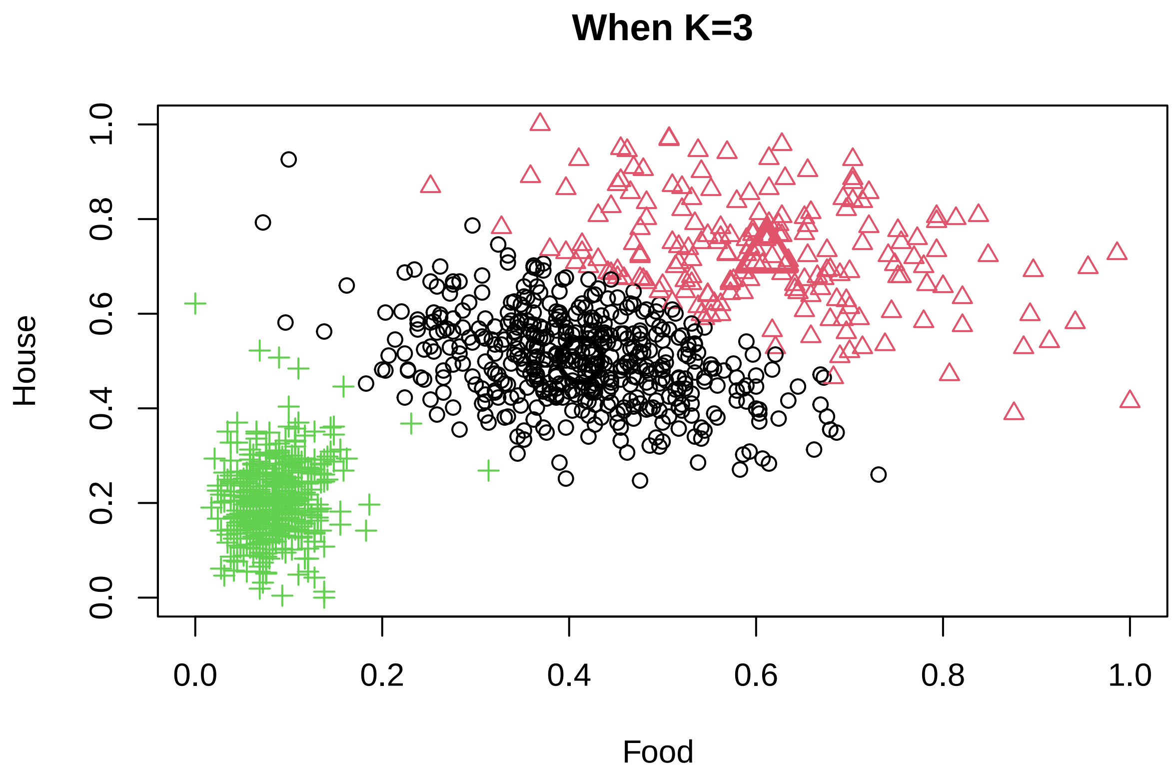
```
Clstr_SJ$betweenss/Clstr_SJ$totss
```

```
## [1] 0.8161152
```

```r
test_SJ$cluster <- factor(Clstr_SJ$cluster)
head(test_SJ)
```

```
##   Food_SJ Enter_SJ Edu_SJ Trans_SJ Work_SJ House_SJ Oth_SJ Food_Std_SJ
## 1   0.043    0.085  0.525    0.180   0.005    0.150  0.012   0.0862069
## 2   0.123    0.055  0.002    0.169   0.121    0.266  0.265   0.3620690
## 3   0.043    0.085  0.506    0.193   0.006    0.155  0.012   0.0862069
## 4   0.119    0.038  0.002    0.301   0.139    0.228  0.172   0.3482759
## 5   0.122    0.038  0.002    0.225   0.095    0.354  0.164   0.3586207
## 6   0.084    0.050  0.002    0.285   0.079    0.264  0.237   0.2275862
##   Enter_Std_SJ  Edu_Std_SJ Trans_Std_SJ Work_Std_SJ House_Std_SJ Oth_Std_SJ
## 1    0.7431193 0.727777778    0.4573864  0.01181102    0.2410148 0.02657807
## 2    0.4678899 0.001388889    0.4261364  0.46850394    0.4862579 0.86710963
## 3    0.7431193 0.701388889    0.4943182  0.01574803    0.2515856 0.02657807
## 4    0.3119266 0.001388889    0.8011364  0.53937008    0.4059197 0.55813953
## 5    0.3119266 0.001388889    0.5852273  0.36614173    0.6723044 0.53156146
## 6    0.4220183 0.001388889    0.7556818  0.30314961    0.4820296 0.77408638
##   cluster
## 1       3
## 2       1
## 3       3
## 4       1
## 5       1
## 6       1
```

```r
centers_SJ <- data.frame(cluster=factor(1:k), Clstr_SJ$centers)

wss_SJ[k-1] <- Clstr_SJ$tot.withinss

plot(test_SJ$Food_Std_SJ, test_SJ$House_Std_SJ, #ploting scatter plot for analyzing
     col=test_SJ$cluster,
      main=" When K=3 ",
     xlab="Food ",
     ylab="House",
     pch=as.numeric(test_SJ$cluster)) # when k=3
points(centers_SJ$Food_Std_SJ, centers_SJ$House_Std_SJ,
       col=centers_SJ$cluster, pch=as.numeric(centers_SJ$cluster),
       cex=3, lwd=3)
```




```
k=4

Clstr_SJ <- kmeans(test_SJ[,c(8,13)], iter.max=10, centers=k, nstart=10)
Clstr_SJ$size
```

```
## [1] 404 228 165 262
```

```
Clstr_SJ$centers
```

```
##   Food_Std_SJ House_Std_SJ
## 1   0.0850717    0.2039803
## 2   0.4941168    0.4409054
## 3   0.6252038    0.7368057
## 4   0.3427086    0.5665155
```

```
Clstr_SJ$betweenss/Clstr_SJ$totss
```

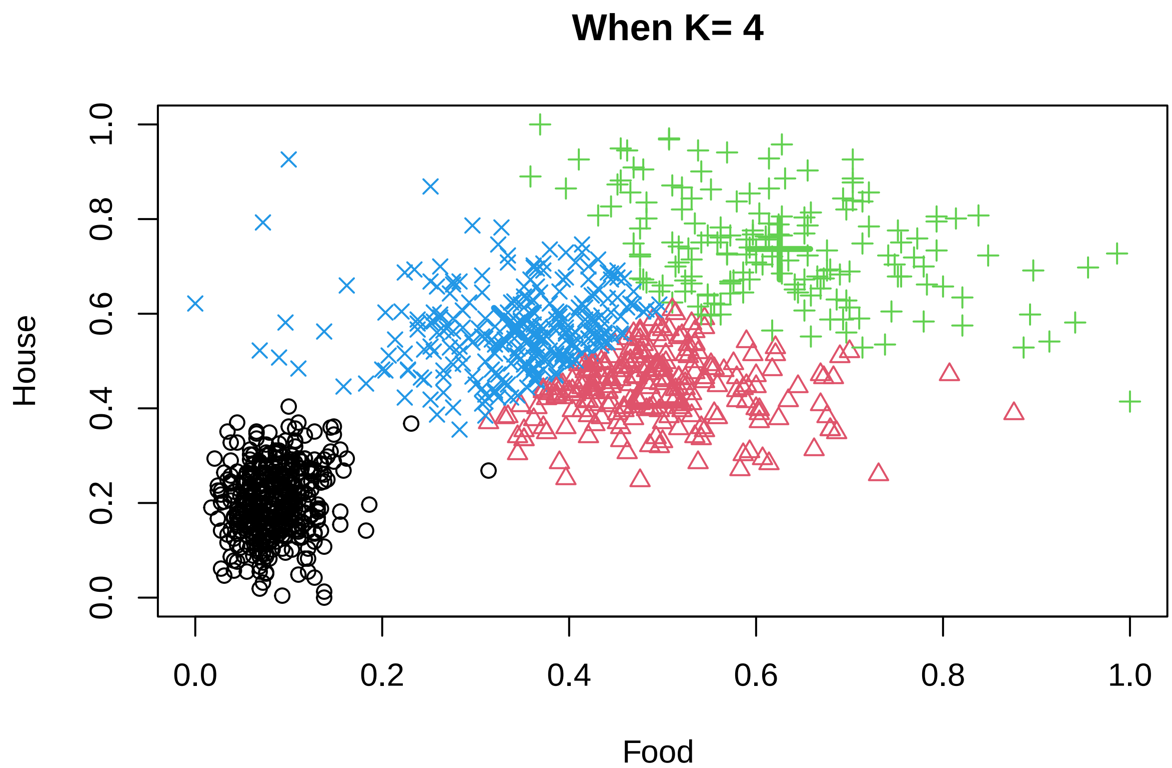
```
## [1] 0.8609939
```

```
test_SJ$cluster <- factor(Clstr_SJ$cluster)
head(test_SJ)
```

```
##   Food_SJ Enter_SJ Edu_SJ Trans_SJ Work_SJ House_SJ Oth_SJ Food_Std_SJ
```

```
## 1   0.043    0.085  0.525    0.180   0.005    0.150  0.012    0.0862069
## 2   0.123    0.055  0.002    0.169   0.121    0.266  0.265    0.3620690
## 3   0.043    0.085  0.506    0.193   0.006    0.155  0.012    0.0862069
## 4   0.119    0.038  0.002    0.301   0.139    0.228  0.172    0.3482759
## 5   0.122    0.038  0.002    0.225   0.095    0.354  0.164    0.3586207
## 6   0.084    0.050  0.002    0.285   0.079    0.264  0.237    0.2275862
##   Enter_Std_SJ  Edu_Std_SJ Trans_Std_SJ Work_Std_SJ House_Std_SJ Oth_Std_SJ
## 1    0.7431193 0.727777778    0.4573864  0.01181102    0.2410148 0.02657807
## 2    0.4678899 0.001388889    0.4261364  0.46850394    0.4862579 0.86710963
## 3    0.7431193 0.701388889    0.4943182  0.01574803    0.2515856 0.02657807
## 4    0.3119266 0.001388889    0.8011364  0.53937008    0.4059197 0.55813953
## 5    0.3119266 0.001388889    0.5852273  0.36614173    0.6723044 0.53156146
## 6    0.4220183 0.001388889    0.7556818  0.30314961    0.4820296 0.77408638
##   cluster
## 1       1
## 2       4
## 3       1
## 4       2
## 5       4
## 6       4
```

```
centers_SJ <- data.frame(cluster=factor(1:k), Clstr_SJ$centers)

wss_SJ[k-1] <- Clstr_SJ$tot.withinss

plot(test_SJ$Food_Std_SJ, test_SJ$House_Std_SJ, #ploting scatter plot for analyzing
     col=test_SJ$cluster,
      main=" When K= 4",
     xlab="Food ",
     ylab="House",
     pch=as.numeric(test_SJ$cluster)) # when k= 4
points(centers_SJ$Food_Std_SJ, centers_SJ$House_Std_SJ,
       col=centers_SJ$cluster, pch=as.numeric(centers_SJ$cluster),
       cex=3, lwd=3)
```



2. Based on the WSS plot (3.2) and the charts (4.1) choose one set of clusters that best describes the data.

-> By observing the elbow chart and then plotting the scatter plot, the cluster which is having lowest values for houses and food, is the one in which the points are most thighly bound. So, it describes data the best when the value of k is 3.

3. Create summary tables for the segmentation/clustering scheme (selected in step 4.2).

```
SummClusters_SJ <- aggregate(cbind(Food_SJ,Enter_SJ ,Edu_SJ,Trans_SJ,
                             Work_SJ,House_SJ,Oth_SJ) ~ cluster, test_SJ,
                             FUN=mean)
SummClusters_SJ
```

```
##   cluster    Food_SJ   Enter_SJ      Edu_SJ  Trans_SJ    Work_SJ  House_SJ
## 1       1 0.04267079 0.06599505 0.555388614 0.1873317 0.00545297 0.1324827
## 2       2 0.16129386 0.03821053 0.004850877 0.2371096 0.14145614 0.2445482
## 3       3 0.19930909 0.01863030 0.076236364 0.1050485 0.09116970 0.3845091
## 4       4 0.11738550 0.03721374 0.009320611 0.2295763 0.13994275 0.3039618
##       Oth_SJ
## 1 0.01065594
```

```
## 2 0.17264912
## 3 0.12506061
## 4 0.16269084
```

4. Create suitable descriptive names for each cluster.

```
-> By analyzing the three cluster s while putting the value of k as 3,
   the relevent titles of each of them as as follows:
       Cluster 1: Food and Housing Insecurity/ Lower Class / Poor
                 - as observed, people having very less food and
                   very little house or no houses, this is the category
                   or cluster of people belonging to very poor financial
                   background.
       Cluster 2: Basic needs met/ Middle class/ Modetate
                 - this is the cluster which belongs to the middle clas
                 and upper middle class pople having gtheir basic
                 neccecities fulfiled.
       Cluster 3: Food and Housing Security/ High class / Rich
                 - this is the cluster representing the priviledged people
                  who are financially free.
```

5. Suggest possible uses for this clustering scheme.

```
 -> By this clustering, one group unorganized and unlabelled under
    differnet clusters, depending on the value of k decided.
   By using the clustering sceme the government can analyze and make
   dicisions for the betterment of any of the group identifiend.

  According to the clustering, one can know where more number of houses
  are there, this can help in real estate market analysis.The urabn planning
  of the city can also be done on the basis of the inffered data.

  Planning of food supply, and schemes for food in low prizes can be done
  by the government by the data.
```