```
#### Exercise 2 ####

## Question a: Load the file zeta.csv of income data into R.
```{r}
setwd("/Users/tenzintashi/Downloads/CSc 460 - DS/Assignment")
# file <- file.choose()
# zeta <- read.csv(file)
zeta <- read.csv(file='Dataset/zeta.csv', header=TRUE, sep=",")
View(zeta)
```
Question b: Change the column names of your data frame so that zcta becomes zipCode and
```

	X	zcta	sex	meanage	meaneducation	meanemployment	meanhouseholdincome
1	1	602	F	37.40335	10.912822	0.7400294	18533.84
2	2	602	M	35.93574	10.692618	1.3438833	18533.84
3	3	604	F	31.80943	13.913371	1.0858555	40784.49
4	4	604	M	31.10425	14.264654	1.6025594	40784.49
5	5	606	F	35.99079	10.097773	0.6287526	17496.53
6	6	606	M	35.99027	9.927995	1.2169643	17496.53
7	7	610	F	37.26014	10.969157	0.8543247	19416.41
8	8	610	M	36.80649	10.850965	1.2955239	19416.41
9	9	612	F	40.42732	11.575772	0.7815393	21607.34
10	10	612	M	37.69992	11.391897	1.2723553	21607.34

```
## Question b: Change the column names of your data frame so that zcta becomes zipCode and
## meanhouseholdincome becomes income.
``````{r}
names(zeta)[2] <- "zipCode"
names(zeta)[7] <- "income"
View(zeta)
```
```

	X	zipCode	sex	meanage	meaneducation	meanemployment	income
1	1	602	F	37.40335	10.912822	0.7400294	18533.84
2	2	602	M	35.93574	10.692618	1.3438833	18533.84
3	3	604	F	31.80943	13.913371	1.0858555	40784.49
4	4	604	M	31.10425	14.264654	1.6025594	40784.49
5	5	606	F	35.99079	10.097773	0.6287526	17496.53
6	6	606	M	35.99027	9.927995	1.2169643	17496.53

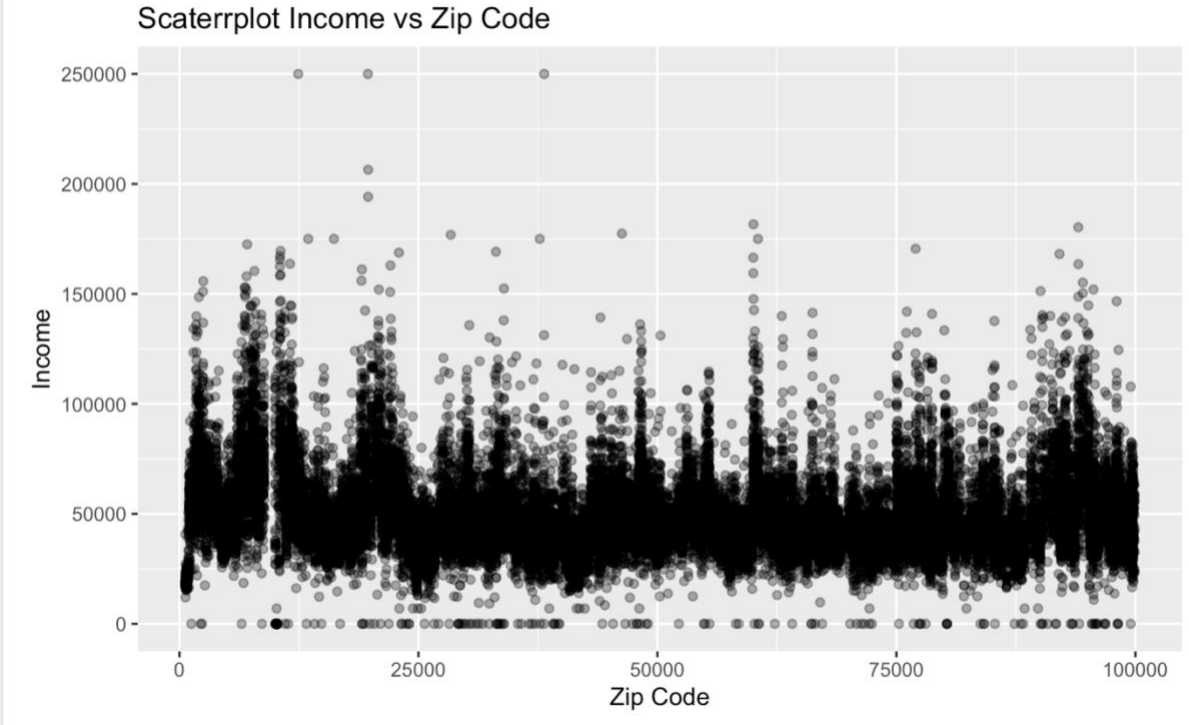
```
## Question c: Analyze the summary of your data.
```{r}
summary(zeta)
```
```

```
       X            zipCode          sex                meanage       meaneducation   meanemployment
 Min.   :    1   Min.   :  601   Length:64076       Min.   :  0.00   Min.   : 0.00   Min.   :0.000
 1st Qu.:16020   1st Qu.:27305   Class :character   1st Qu.: 36.65   1st Qu.:11.91   1st Qu.:1.542
 Median :32038   Median :49909   Mode  :character   Median : 39.30   Median :12.46   Median :1.813
 Mean   :32038   Mean   :49801                      Mean   : 39.68   Mean   :12.53   Mean   :1.787
 3rd Qu.:48057   3rd Qu.:72007                      3rd Qu.: 42.28   3rd Qu.:13.11   3rd Qu.:2.077
 Max.   :64076   Max.   :99950                      Max.   :137.08   Max.   :19.00   Max.   :3.000
     income
 Min.   :     0
 1st Qu.: 37642
 Median : 44163
 Mean   : 48245
 3rd Qu.: 54373
 Max.   :250000
```

```
## What are the mean and median average incomes?
```{r}
# Mean average Income = 48245
# Median average Income = 44163
```
```

```
## Question d: Plot a scatter plot of the data. Although this graph is not too informative,
## do you see any outlier values? If so, what are they?

```{r}
library(ggplot2)
ggplot(zeta,aes(x= zipCode, y=income)) +geom_point(alpha=0.2) +labs(x="Zip Code",y="Income",title="Scaterrplot Income vs Zip Code")
```
```



There seem to be two outlier values are 0 and 250000

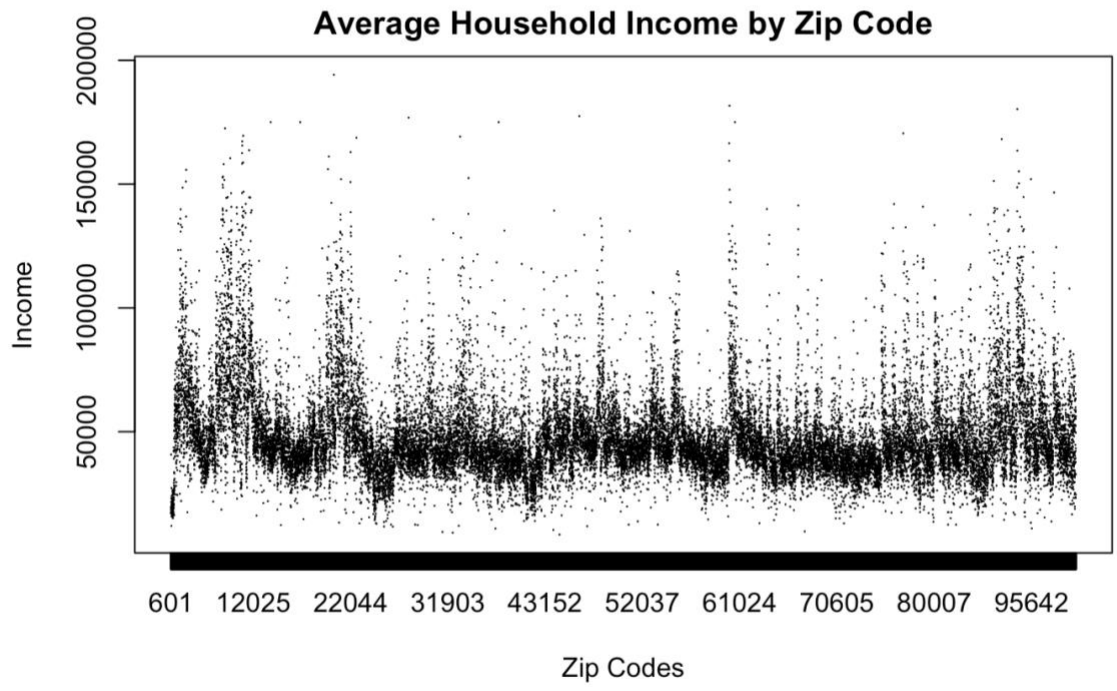
```
## Question e: In order to omit outliers, create a subset of the data so that: $7,000 < income < $200,000,
## What's your new mean?
```{r}
newData <- subset(zeta, income <200000 & income >7000)
summary(newData)
```
```

```
       X            zipCode          sex                meanage       meaneducation   meanemployment
 Min.   :    1   Min.   :  601   Length:63742       Min.   :  0.00   Min.   : 0.00   Min.   :0.000
 1st Qu.:16046   1st Qu.:27333   Class :character   1st Qu.: 36.69   1st Qu.:11.91   1st Qu.:1.546
 Median :32076   Median :49935   Mode  :character   Median : 39.31   Median :12.46   Median :1.816
 Mean   :32051   Mean   :49817                      Mean   : 39.78   Mean   :12.57   Mean   :1.795
 3rd Qu.:48047   3rd Qu.:72003                      3rd Qu.: 42.28   3rd Qu.:13.11   3rd Qu.:2.078
 Max.   :64076   Max.   :99950                      Max.   :133.11   Max.   :19.00   Max.   :3.000
     income
 Min.   :  8465
 1st Qu.: 37755
 Median : 44234
 Mean   : 48465
 3rd Qu.: 54444
 Max.   :194135
```

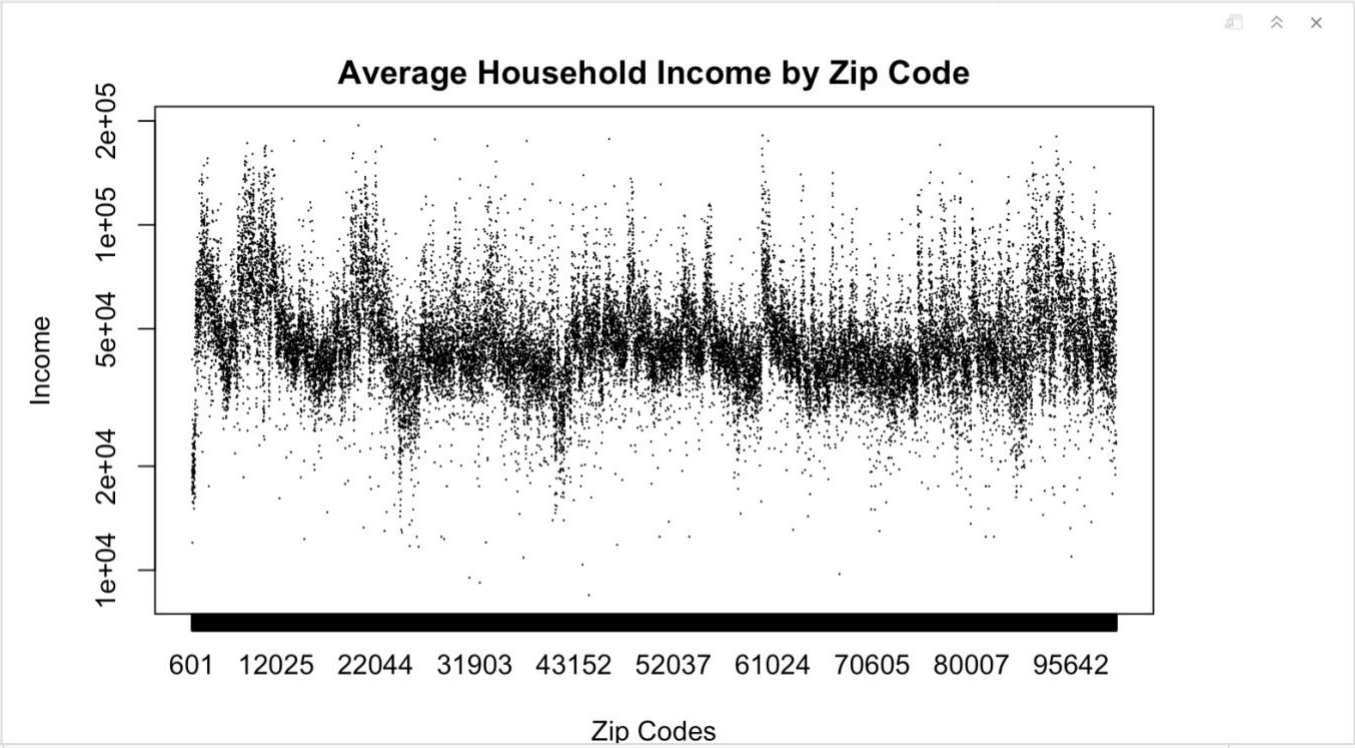
```{r}
# New Mean average Income = 48465
```
```

```
Question f: Create a simple box plot of your data. Be sure to add a title and label the axes.
```{r}
boxplot(col="white", data = newData, income ~ zipCode, main = "Average Household Income by Zip Code", xlab =
"Zip Codes", ylab = "Income")

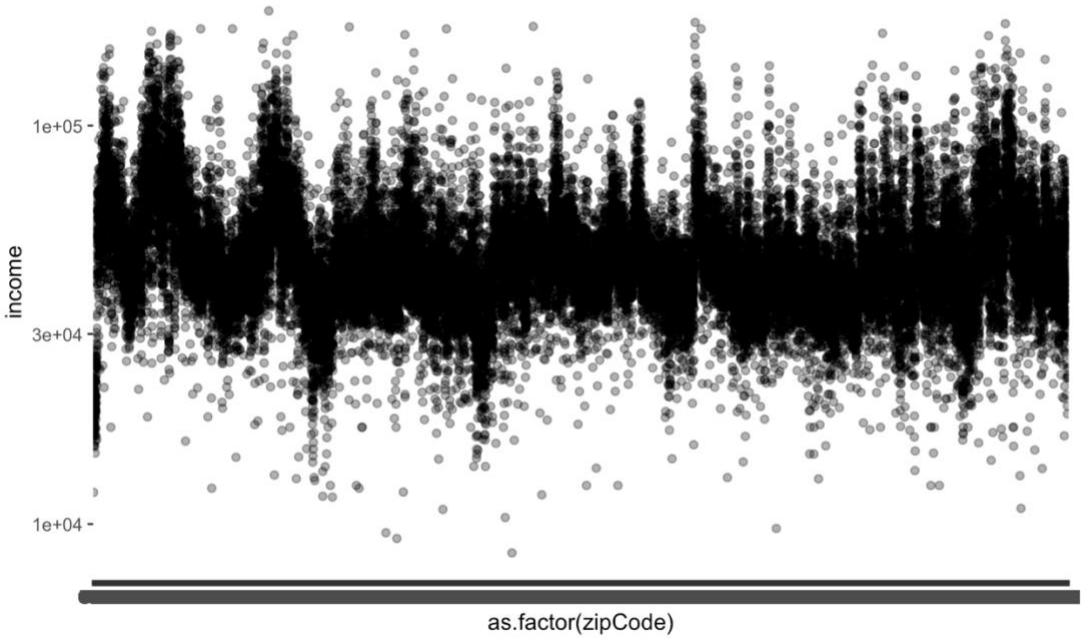
```
```


```

````
## Question g: In the box plot you created, notice that all of the income data is pushed towards
## the bottom of the graph because most average incomes tend to be low. Create a new box plot
## where the y- axis uses a log scale. Be sure to add a title and label the axes.
```{r}
#Create a new box plot where the y-axis uses a log scale
boxplot(col="white", data = newData, income ~ zipCode, main = "Average Household Income by Zip Code", xlab =
"Zip Codes", ylab = "Income", log='y')
```
````



```
## Question h: Use the ggplot library in R, which enables you to create graphs with several different
## types of plots layered over each other. Be sure to read the documentation for ggplot
## and load the library ggplot2 (you may have to install this package into R).
```{r}
library(ggplot2)
#Make a ggplot that consists of just a scatter plot using the function geom_point() with position = “jitter”
ggplot(newData, aes(x = as.factor(zipCode), y=income))+geom_point(position="jitter", alpha=0.2)+scale_y_log10()
```
```




```
## Question i: Make a ggplot that consists of just a scatter plot using the function geom_point()
## with position = “jitter” so that the data points are grouped by zip code. Be sure to use ggplot’s
## function for taking the log10 of the y-axis data. (Hint: for geom_point, have alpha=0.2).
```{r}
library(ggplot2)
ggplot(newData, aes(x = as.factor(newData$zipCode), y = newData$income))+geom_point(aes(colour=factor(zipCode)),position =
'jitter',alpha=0.2)+ geom_boxplot(alpha=0.1,outlier.size =-Inf) + scale_y_log10()+labs(color="Region",x="Zip
Code",y="Income",title="Average Income by Zip Code") + theme(plot.title = element_text(size =11, face="plain",hjust = 0.5))
```
Question j: Create a new ggplot by adding a box plot layer to your previous graph.
To do this, add the ggplot function geom_boxplot(). Also, add color to the scatter plot so
that data points between different zip codes are different colors. Be sure to label the axes and
add a title to the graph. (Hint: for geom_boxplot, have alpha=0.1 and outlier.size=0).
```{r}
library(ggplot2)
ggplot(newData,aes(x=as.factor(zipCode),y=income))+geom_point(aes(colour=factor(zipCode)),position = 'jitter',alpha=0.2)+
geom_boxplot(alpha=0.1,outlier.size =0) + scale_y_log10()+ ylab("Income") + xlab("Zip Code") + ggtitle ("Average Income by Zip
Code") + labs(color="Region") + theme(plot.title = element_text(size =11, face="plain",hjust = 0.5))
```
Question k: What can you conclude from this data analysis/visualization?
```{r}
# - It is important to visualize your data in different ways.

# - Visualization enables you to better understand what your data is telling you.

# - Visualization enables you to better communicate your results to stakeholders.

# - Zip codes starting in 0 (New England) and 9 (West Coast) have higher average household incomes.
```
```

```
## Question k: What can you conclude from this data analysis/visualization?
```{r}
# - It is important to visualize your data in different ways.

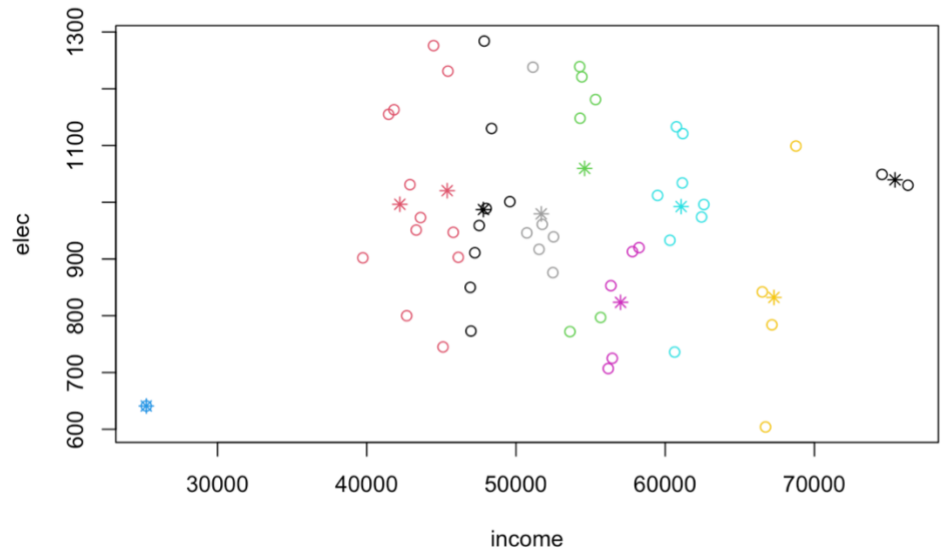
# - Visualization enables you to better understand what your data is telling you.

# - Visualization enables you to better communicate your results to stakeholders.

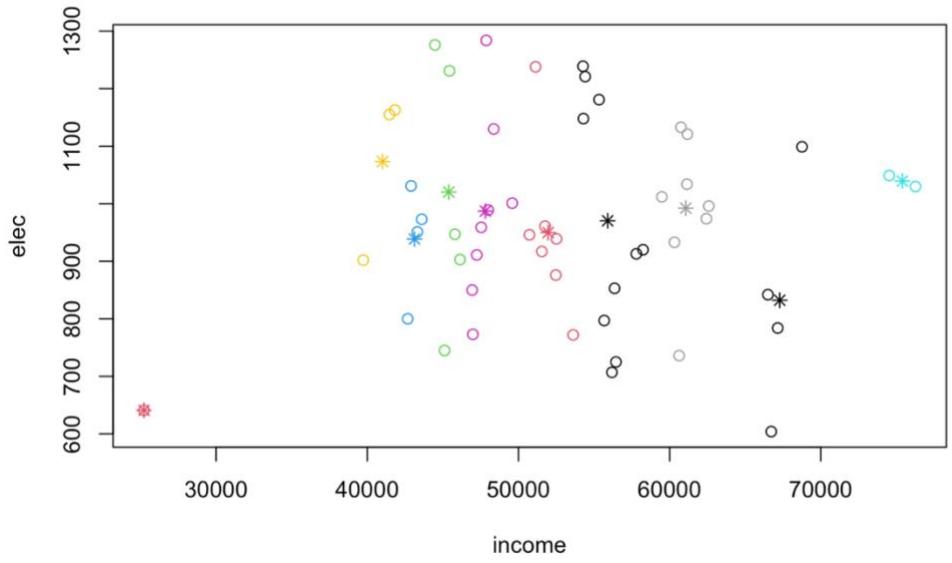
# - Zip codes starting in 0 (New England) and 9 (West Coast) have higher average household incomes.
```
```

```
#### Exercise 2.2 ####

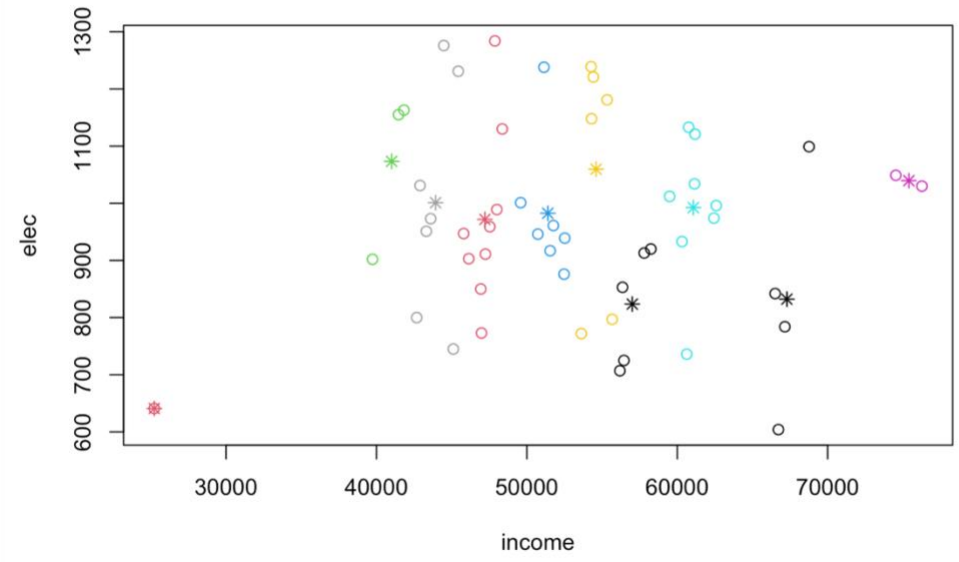
## Question a: Cluster the data and plot all 52 data points, along with the centroids.
## Mark all data points and centroids belonging to a given cluster with their own color. Here, let k=10.
```{r}
load('income_elec_state.Rdata')
k = kmeans(income_elec_state, 10)
plot(income_elec_state, col = k$cluster)
points(k$centers, col=1:10, pch=8)
```
```



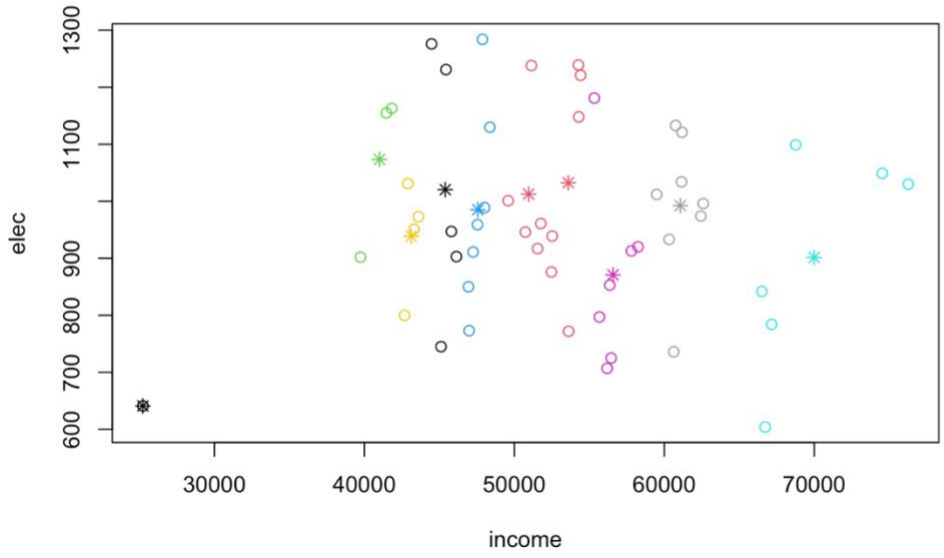
```
## Question b: Repeat step (a) several times. What can change each time you cluster the data?
## Why? How do you prevent these changes from occurring?
```{r}
k = kmeans(income_elec_state, 10)
plot(income_elec_state, col = k$cluster)
points(k$centers, col=1:10, pch=8)
```
```




```
```{r}
k = kmeans(income_elec_state, 10)
plot(income_elec_state, col = k$cluster)
points(k$centers, col=1:10, pch=8)
```
```



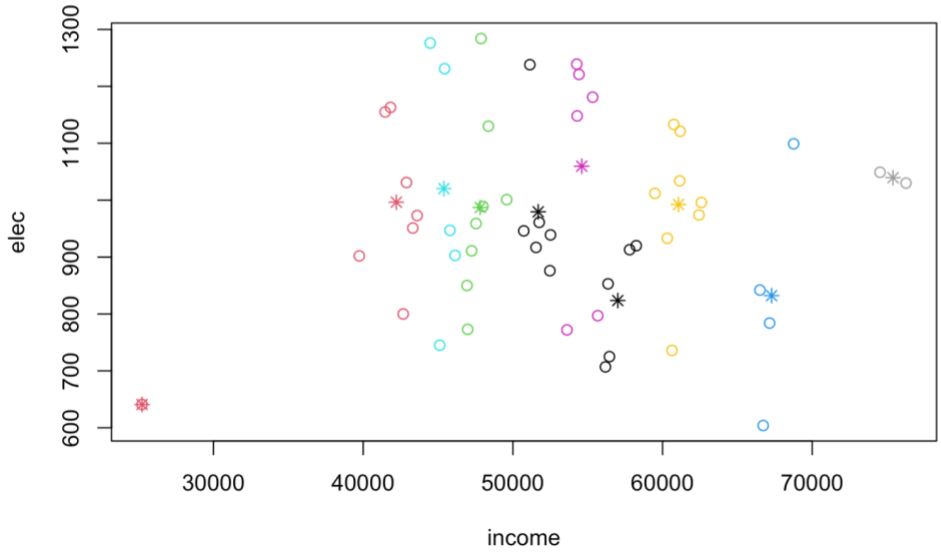
```{r}
k = kmeans(income_elec_state, 10)
plot(income_elec_state, col = k$cluster)
points(k$centers, col=1:10, pch=8)
```



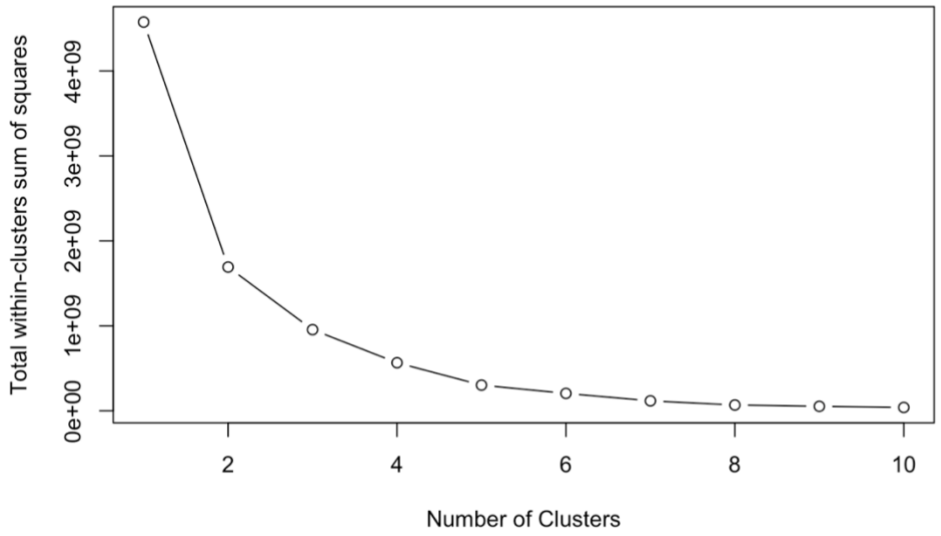
```{r}
# Sizes, centers' position, sum of squares of clusters can change after each time repeat above step.
# Because by default nstart = 1: having only one random starting set can result in different
# clusterings over multiple runs.

# To prevent these changes from occurring, we can:
  # - Increase "nstart" to improve the likelihood of obtaining the globally optimal clustering.
  # - Increasing the "iter.max" parameter reduces the likelihood that the kmeans algorithm terminates
  #   prematurely.
```

```{r}
k = kmeans(income_elec_state, 10, nstart=100, iter.max = 50)
plot(income_elec_state, col = k$cluster)
points(k$centers, col=1:10, pch=8)
```



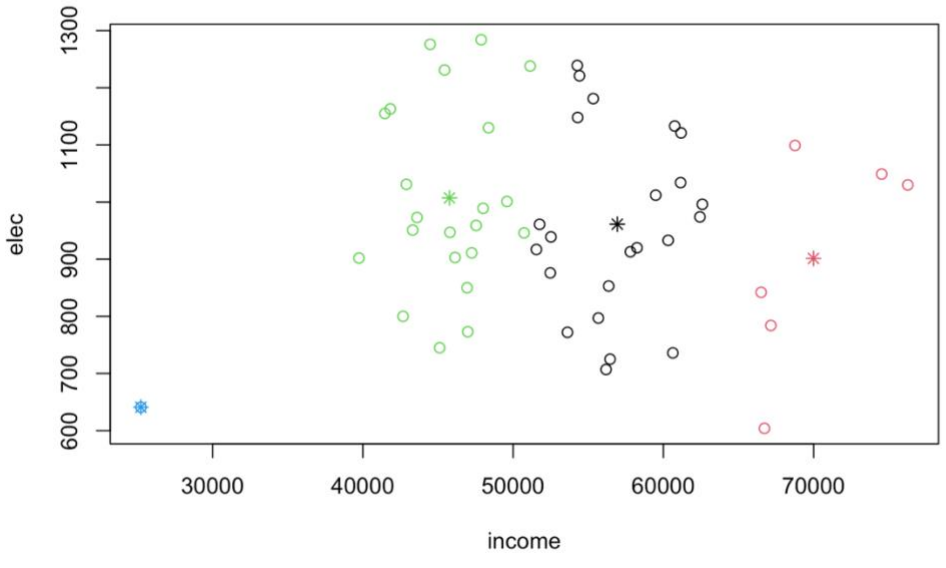
```
## Question c: Once you've accounted for the issues in the previous step, determine a reasonable
## value of k. Why would you suggest this value of k?
```{r}
wss =numeric(10)
for (i in 1:10) wss[i] = sum(kmeans(income_elec_state, centers=i, nstart = 100, iter.max = 50)$tot.withinss)
plot(1:10, wss, type="b", xlab="Number of Clusters", ylab="Total within-clusters sum of squares")
```
```




```
We can see here "elbow" = 4

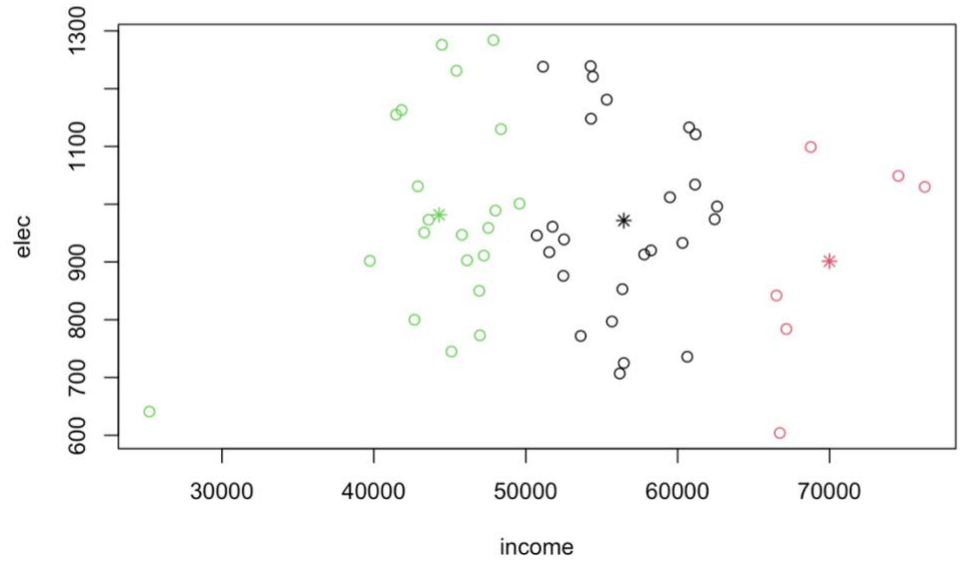
With k=4

```{r}
k = kmeans(income_elec_state, 4, nstart=100, iter.max = 50)
plot(income_elec_state, col = k$cluster)
points(k$centers, col=1:4, pch=8)
```
```



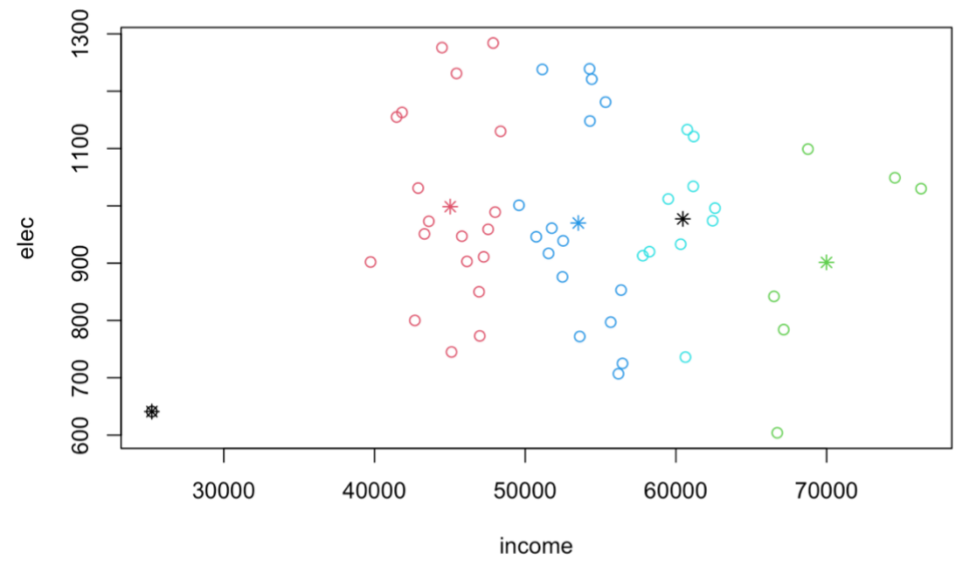
Repeat the modeling with k=3

```{r}
k = kmeans(income_elec_state, 3, nstart=100, iter.max = 50)
plot(income_elec_state, col = k$cluster)
points(k$centers, col=1:4, pch=8)
```



Repeat the modeling with k=5

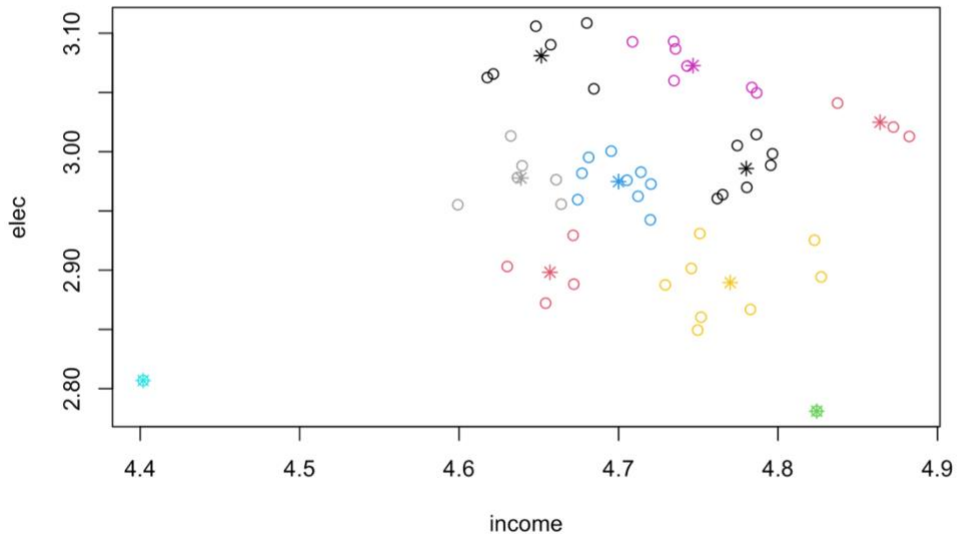
```{r}
k = kmeans(income_elec_state, 5, nstart=100, iter.max = 50)
plot(income_elec_state, col = k$cluster)
points(k$centers, col=1:4, pch=8)
```



Chosen k=4. Because we see that Puerto Rico is an outlier, and should perhaps belong to its own cluster. It is the smallest k such that Puerto Rico
belongs to its own cluster, so this k would be a good value to suggest.

Question d: Convert the mean household income and mean electricity usage to a log10 scale and
cluster this transformed dataset. How has the clustering changed? Why?

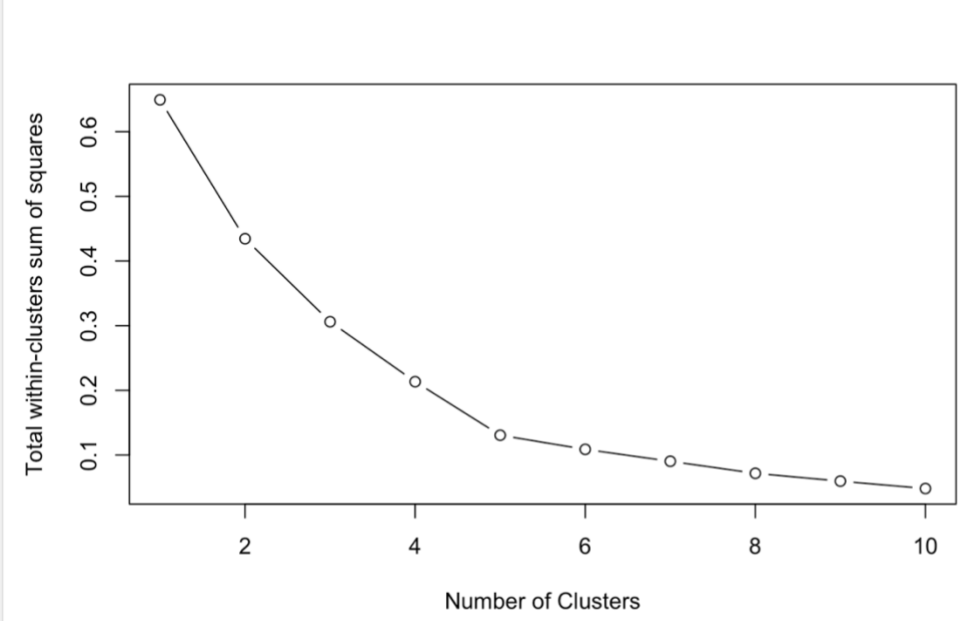
```{r}
new = log10(income_elec_state)
k = kmeans(new, 10, nstart=100, iter.max = 50)
plot(new, col = k$cluster)
points(k$centers, col=1:10, pch=8)
```



K-means clustering is not scale-invariant, so any adjustments made to the units of the data may impact the clustering.

Question e: Reevaluate your choice of k. Would you now choose k differently? Why or why not?

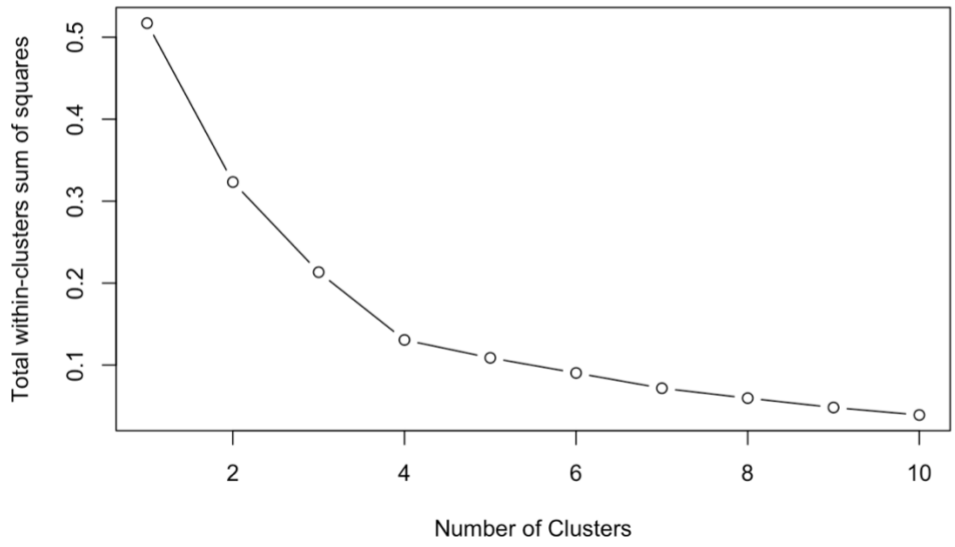
```{r}
wss =numeric(10)
for (i in 1:10) wss[i] = sum(kmeans(new, centers=i, nstart = 100, iter.max = 50)$tot.withinss)
plot(1:10, wss, type="b", xlab="Number of Clusters", ylab="Total within-clusters sum of squares")
```



We see more clear elbow in the different position: k=5

```r
## Question f: Have you observed an outlier in the data? Remove the outlier and, once again, reevaluate your
## choice of k
```{r}
new <- subset(new, rownames(new) != "PR")

wss =numeric(10)
for (i in 1:10) wss[i] = sum(kmeans(new, centers=i, nstart = 100, iter.max = 50)$tot.withinss)
plot(1:10, wss, type="b", xlab="Number of Clusters", ylab="Total within-clusters sum of squares")
```
```



After removing the outliers, it is clear that elbow on the plot change its position to smaller value. k=4

Question g: Color a map of the U.S. according to the clustering you obtained. To simplify this task, use
the “maps” package and color only the 48 contiguous states and Washington D.C.

```{r}
library(maps)

km <- kmeans(new,4,nstart = 100, iter.max = 50)
#Prepare vector with state order
map_order <- c('AL', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DE', 'DC', 'FL',
'GA', 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME',
'MD', 'MA', 'MA', 'MA', 'MI', 'MI', 'MN', 'MS', 'MO',
'MT', 'NE', 'NV', 'NH', 'NJ', 'NM', 'NY', 'NY', 'NY',
'NY', 'NC', 'NC', 'NC', 'ND', 'OH', 'OK', 'OR', 'PA',
'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VT', 'VA', 'VA',
'VA', 'WA', 'WA', 'WA', 'WA', 'WA', 'WV', 'WI', 'WY')
#Prepare color vector
map_color <- km$cluster[map_order]
map('state', col = map_color,fill=TRUE)
```