Introduction to Data Mathematics Spring 2021

Lab 2: K-Means

Jared Gridley

Prelab and Lab Overview

In **Prelab 2** you:

- Read in a csv file
- Prepared the data
- Sanity checked data
- Created K-means clusters
- Selected the number of clusters
- Intepreted the clusters using various plots

In **Lab 2** you will apply what you learned in Prelab 2. Be sure to refer back to the Prelab and its Appendix to guide you through the exercises.

This notebook includes only **Lab** sections; you were expected to complete the separate **Prelab** work before your Wednesday lab session.

Questions highlighted as *Exercises* will be graded, so be sure you have answered them in your final submission! After you have completed this notebook. "Knit"" it and upload it to LMS under "Lab 2" assignment. **Make sure that your name is at the top!**

The National Health and Nutrition Examination Survey (NHANES) is a survey research program conducted by the National Center for Health Statistics to assess the health and nutritional status of adults and children in the United States, and to track changes over time. In Lab 2, you will be analyzing data on the dietary habits of people in the United States collected from the NHANES 2005-2006 Dietary Data originally from https://wwwn.cdc.gov/nchs/nhanes/Search/DataPage.aspx?Component=Dietary&CycleBeginYear=2005.

Read the data file `~/MATP-4400/data/dietary_data_2005_complete.csv` into a data frame called `raw.df`.

```r
# This code is provided for you.
raw.df <- read.csv('~/MATP-4400/data/dietary_data_2005_complete.csv')
str(raw.df)
```

```
## 'data.frame':    3332 obs. of  17 variables:
##  $ SEQN                            : num  31131 31132 31134 31150 31152 ...
##  $ gender                          : Factor w/ 2 levels "female","male": 1 2 2 2 1 1 2 2 2 1 ...
##  $ age_years                       : int  44 70 73 79 27 44 62 38 71 39 ...
##  $ ethnicity                       : Factor w/ 5 levels "mexican_american",..: 2 3 3 3 1 4 2 4 3 3 ..
##  $ education_level                 : Factor w/ 5 levels "college","college_grad",..: 1 2 4 4 4 4 4 2
##  $ marital_status                  : Factor w/ 6 levels "divorced","living_with_partner",..: 3 3 3 1
##  $ household_income                : int  11 11 12 3 7 3 5 10 8 7 ...
##  $ bmi                             : num  30.9 24.7 30.6 28.9 39.9 ...
##  $ had_12_alcoholic_drinks_in_year: Factor w/ 2 levels "no","yes": 1 2 2 2 1 2 2 1 2 2 ...
##  $ smoked_100_cigarettes_in_life  : Factor w/ 2 levels "no","yes": 1 1 1 2 1 2 2 1 2 1 ...
##  $ veg_raw                         : num  25.2 139.8 31.6 20.5 27.5 ...
##  $ veg_cooked                      : num  14.3 24.2 78.8 0 159 ...
```

```
##  $ veg_mixed                           : num  0 0 0 183 130 ...
##  $ veg_fried                           : num  8.4 0 68.2 0 0 ...
##  $ fruit_raw                           : num  32.8 429.9 36 383.5 0 ...
##  $ fruit_juice                         : num  348.75 0 0 496.1 1.27 ...
##  $ fruit_misc                          : num  0 0 0 18.1 0 ...
```

First we need to "clean" the dataset. Here is a code sample you can use to help you clean your dataset.

We can see from the compact summary above that `raw.df` has a mix of data types.

Assume our goal is to make a numeric matrix consisting of `fruit_raw` and `gender` to run for kmeans.

In the following code we'll select just the variables `fruit_raw` and `gender` from `raw.df` using `select()` to pick the columns, and then put the results in `exampledata.df`. Here we are exploiting the `dplyr` package from the `tidyverse` family of packages which makes data cleaning so much easier!

```
# select the variables gender and fruit_raw
exampledata.df <- raw.df %>% select(c('fruit_raw','gender' ))
```

Now we want to transform gender into a number instead of a factor using the following codebook: `"male" = 1` and `"female" = 2`.

If we transform the gender factors into 'ordered' factors, then we can convert these to the required numbers using "as.numeric()."

```
# here is an example of how to convert gender in raw.df$gender into numbers in the correct order
gendertransformed<-as.numeric(ordered(raw.df$gender, levels = c("male", "female")))
head(gendertransformed) # show us a few variables
```

```
## [1] 2 1 1 1 2 2
```

We can create a dplyr pipeline to create `exampledata.df` in a single command. This pipeline selects the variables `gender` and `household_income` and then transforms gender to be numeric using the `mutate` function.

```
# select variables and transform gender
exampledata.df <- raw.df %>%
  select(c('gender','fruit_raw' )) %>%
  mutate(gender=as.numeric(ordered(gender, levels = c("male", "female"))))

# check that we got what we wanted.
summary(exampledata.df)
```

```
##      gender         fruit_raw
##  Min.   :1.000   Min.   :   0.00
##  1st Qu.:1.000   1st Qu.:   0.00
##  Median :2.000   Median :  64.00
##  Mean   :1.532   Mean   :  98.96
##  3rd Qu.:2.000   3rd Qu.: 148.50
##  Max.   :2.000   Max.   :1239.47
```

Exercise 1 :

Create the matrix `D.df` for analysis.

- Create a data frame consisting of the variables `veg_raw`, `veg_cooked`, `veg_mixed`, `veg_fried`, `fruit_raw`, `fruit_juice`, `fruit_misc`, `bmi`, `gender`, and `education_level`.
- Transform the factor variables into number, as follows
  - For `gender` let 1 = "male" and 2 = "female".
  - For `education_level` let 1 = "pre_highschool", 2 = "highschool", 3 = "highschool_grad", 4 = "college", 5 = "college_grad".

- Save the results in `D.df`
- Verify that the results are correct. `D.df` should have 3332 obs and 10 numeric variables.

```r
# Creating the dataframe with specified variables
D.df <- raw.df %>%
  select(c("veg_raw", "veg_cooked", "veg_mixed", "veg_fried", "fruit_raw", "fruit_juice",
           "fruit_misc", "bmi", "gender", "education_level")) %>%
  mutate(gender=as.numeric(ordered(gender, levels = c("male", "female")))) %>%
  mutate(education_level=as.numeric(ordered(education_level, levels = c("pre_highschool",
                                                                         "highschool",
                                                                         "highschool_grad",
                                                                         "college",
                                                                         "college_grad"))))

#Check the rows and columns
nrow(D.df)
```

```
## [1] 3332
```

```r
ncol(D.df)
```

```
## [1] 10
```

```r
summary(D.df)
```

```
##     veg_raw          veg_cooked       veg_mixed        veg_fried
##  Min.   :   0.00   Min.   :  0.00   Min.   :  0.00   Min.   :  0.0
##  1st Qu.:  18.89   1st Qu.: 11.65   1st Qu.:  0.00   1st Qu.:  0.0
##  Median :  65.90   Median : 50.00   Median :  0.00   Median :  0.0
##  Mean   :  93.81   Mean   : 79.05   Mean   : 31.51   Mean   : 21.2
##  3rd Qu.: 137.31   3rd Qu.:112.80   3rd Qu.:  0.00   3rd Qu.: 26.5
##  Max.   :1085.22   Max.   :968.00   Max.   :925.03   Max.   :587.5
##    fruit_raw        fruit_juice       fruit_misc            bmi
##  Min.   :   0.00   Min.   :   0.0   Min.   :  0.000   Min.   :13.36
##  1st Qu.:   0.00   1st Qu.:   0.0   1st Qu.:  0.000   1st Qu.:24.37
##  Median :  64.00   Median :   0.0   Median :  0.000   Median :27.70
##  Mean   :  98.96   Mean   : 100.4   Mean   :  3.157   Mean   :28.81
##  3rd Qu.: 148.50   3rd Qu.: 162.8   3rd Qu.:  0.000   3rd Qu.:31.90
##  Max.   :1239.47   Max.   :2024.7   Max.   :206.125   Max.   :76.07
##      gender      education_level
##  Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:2.000
##  Median :2.000   Median :4.000
##  Mean   :1.532   Mean   :3.381
##  3rd Qu.:2.000   3rd Qu.:4.000
##  Max.   :2.000   Max.   :5.000
```

Exercise 2 :

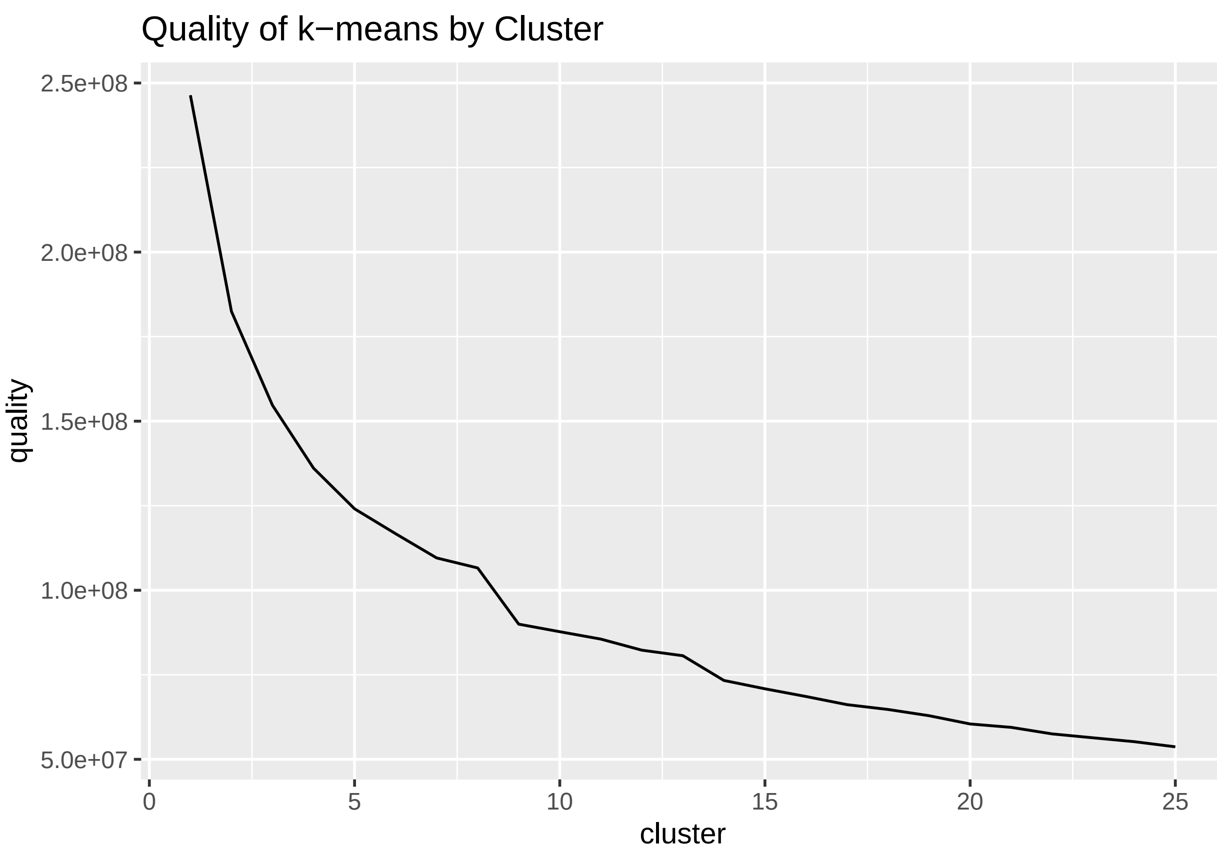
Convert `D.df` into a matrix; store the result as a variable `D.matrix`. Verify that D.matrix is correct using `str()`. 'D.matrix' should have 3332 rows and 10 columns and the entries should all be numeric.

```r
# Converting the Data frame to a matrix
D.matrix <- data.matrix(D.df)

#Checking the dimensions of the matrix
str(D.matrix)
```

```
##  num [1:3332, 1:10] 25.2 139.8 31.6 20.5 27.5 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:10] "veg_raw" "veg_cooked" "veg_mixed" "veg_fried" ...
```

Exercise 3 : Perform K-means on `D.matrix`. Use the function `wssplot` from Prelab 2 to plot the total within sum of squares method over the range k = 1 to 25. Be sure to include a plot of the sum of squares versus number of clusters. Use a random seed of 20 for each call to K-means.

```r
wssplot <- function(data, nc=25, seed=20){
  wss <- data.frame(cluster=1:nc, quality=c(0))
  for (i in 1:nc){
    set.seed(seed)
    wss[i,2] <- kmeans(data, centers=i)$tot.withinss}
  ggplot(data=wss,aes(x=cluster,y=quality)) +
    geom_line() +
    ggtitle("Quality of k-means by Cluster")
}
# Generate the plot
wssplot(D.matrix, nc=25)
```



How many clusters should we pick and why? What value for k's do you think would be reasonable based on the elbow test? Which k do you recommend?

Any number in the range of 4 to 8 would be reasonable for k. I would recommend k=7 for the number of clusters because that number appears to be at the point of the elbow, so it will most likely yield the strongest clusters.

Exercise 4 : For this purpose of this lab, k=6 looks good because we want a smaller number to interpret and as we will see the clusters are quite distinct in the analysis below. Picking the number of clusters is more of an art than science. If you goal is to describe a dataset, then you have the freedom to pick k, so the results are robust and insightful. So it's okay if you picked another number in Exercise 3.

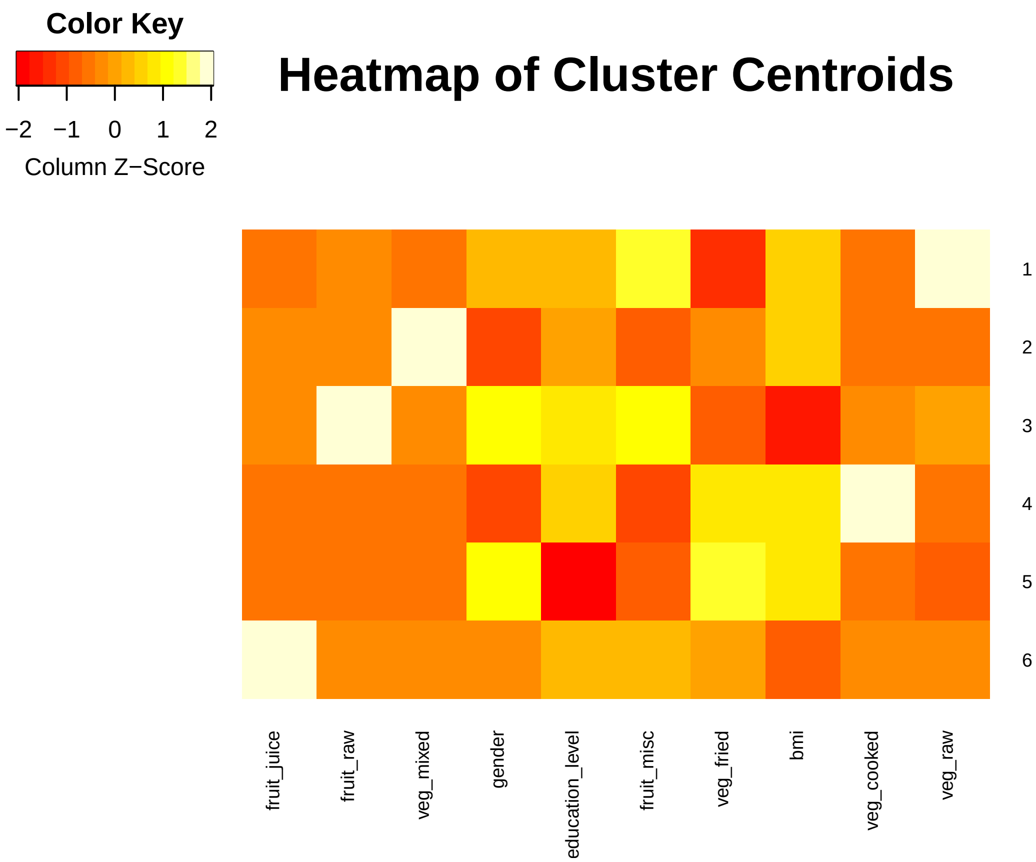
Create your final kmeans model using 6 clusters with a *random seed of 20*.

```
#Setting the random seed to be 20
set.seed(20)
D.km <- kmeans(D.matrix, centers = 6)
kclass=as.factor(D.km$cluster)
Dresults.df <-cbind.data.frame(D.matrix, kclass)

# Displaying the results of the kmeans model
str(D.km)
```

```
## List of 9
##  $ cluster     : int [1:3332] 6 3 5 6 5 6 4 3 5 1 ...
##  $ centers     : num [1:6, 1:10] 265 73.8 109.7 72 53.5 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:6] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:10] "veg_raw" "veg_cooked" "veg_mixed" "veg_fried" ...
##  $ totss       : num 2.46e+08
##  $ withinss    : num [1:6] 13744672 8262126 19019931 11561979 26962614 ...
##  $ tot.withinss: num 1.17e+08
##  $ betweenss   : num 1.3e+08
##  $ size        : int [1:6] 425 159 367 307 1637 437
##  $ iter        : int 6
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

Draw a heatmap of the centers of the 6 clusters using the `heatmap.2()` The default `heatmap.2()` command makes yellow the highest value and red the smallest.

```
#Creating the heatmap for our data
heatmap.2(D.km$centers,
          dendrogram='none',
          main='Heatmap of Cluster Centroids',
          cexRow=0.75,
          cexCol=0.75,
          Rowv=FALSE,
          scale = 'col',
          tracecol=NA,
          density.info='none'
          )
```



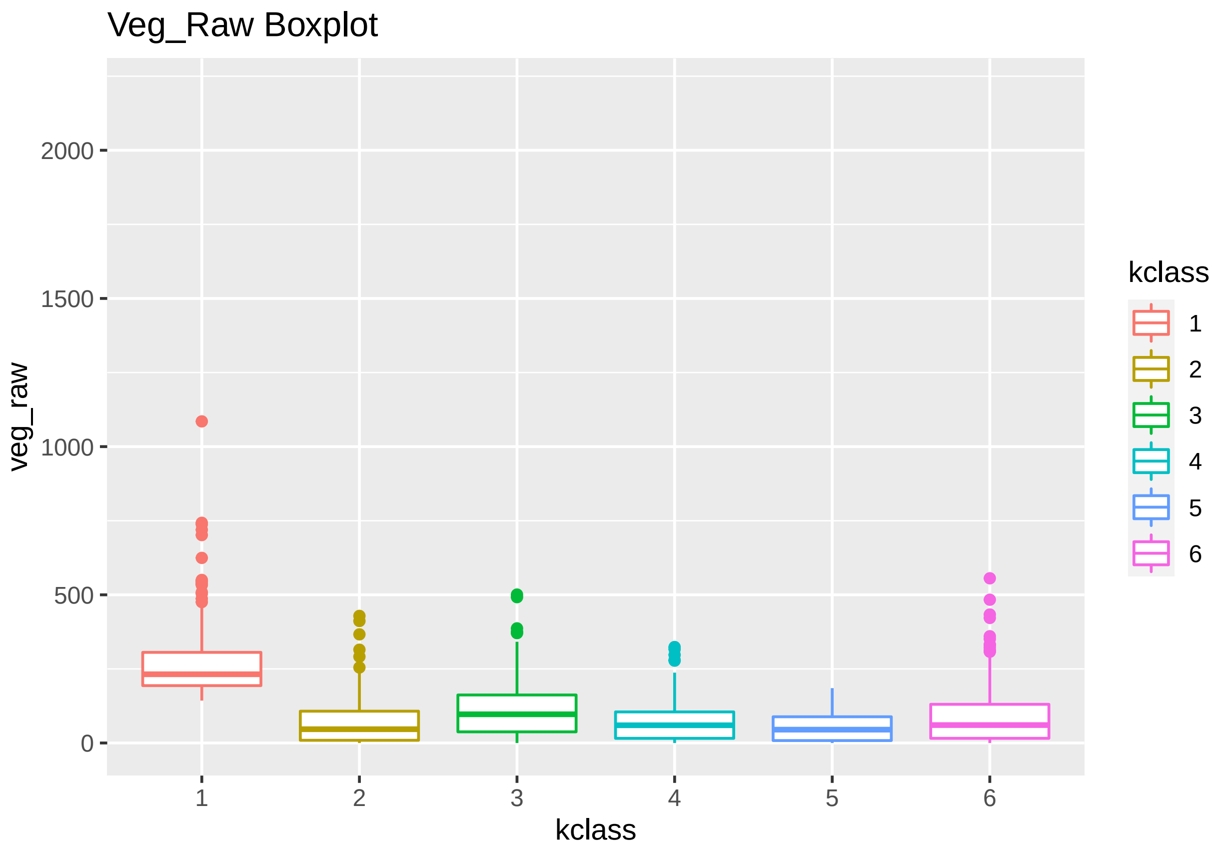
Exercise 5 : Make boxplots of the consumers that belong to each cluster. Make sure they all show the same y ranges on the plots.

One way is to make data frames containing the points in each cluster that you made. Call them `Cluster1`, `Cluster2`,... Then make a boxplot for each cluster showing the distributions of its features. Make sure to give each plot a title. It's helpful to make the cluster boxplots all have the same range, so add `ylim=c(0,2200)` to your boxplot command. Also make the labels vertical by adding 'las=2' The command will have the form. `boxplot(mydata,main="My title", las=2, ylim=c(0,2200))`

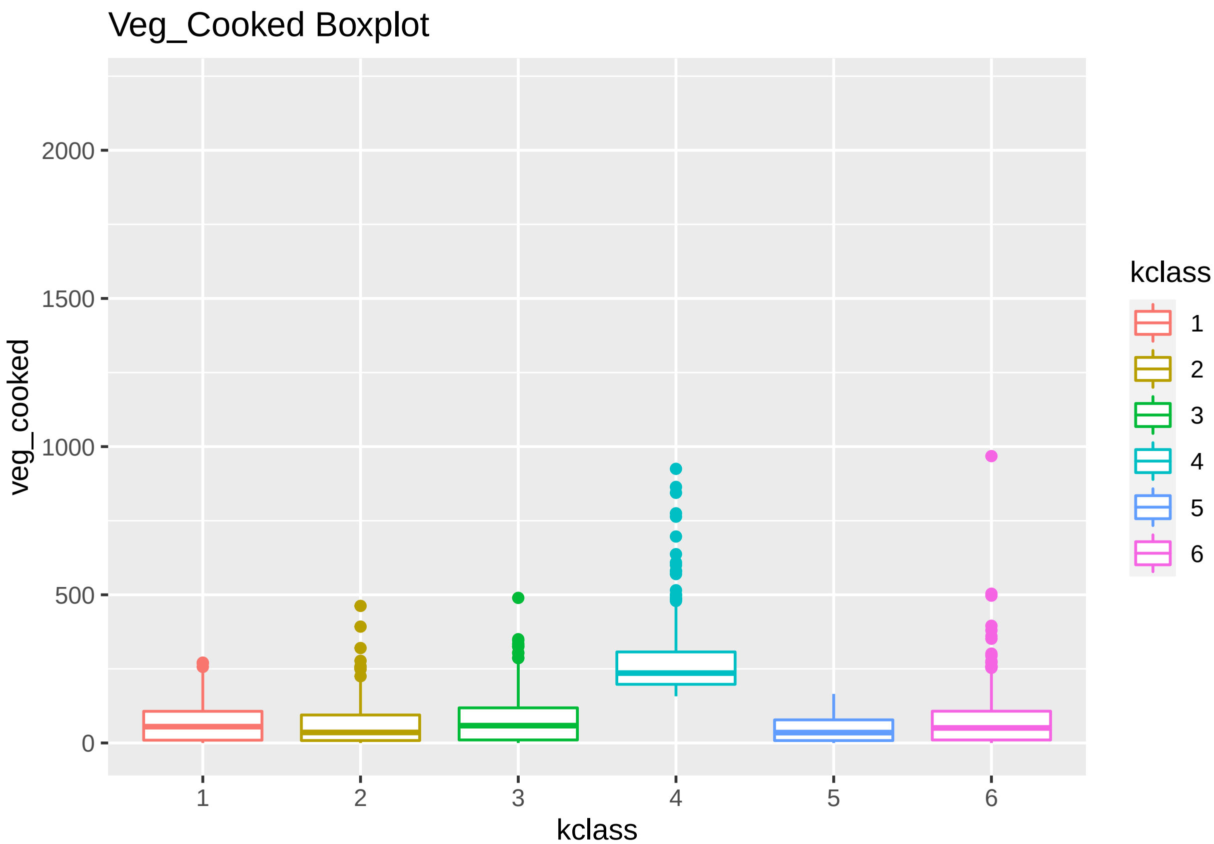
NOTE: You are free to code this exercise anyway you like. There are more elegant ways. Extra credit of 1 point if you figure a different way to plot these figures.

Put your answers here

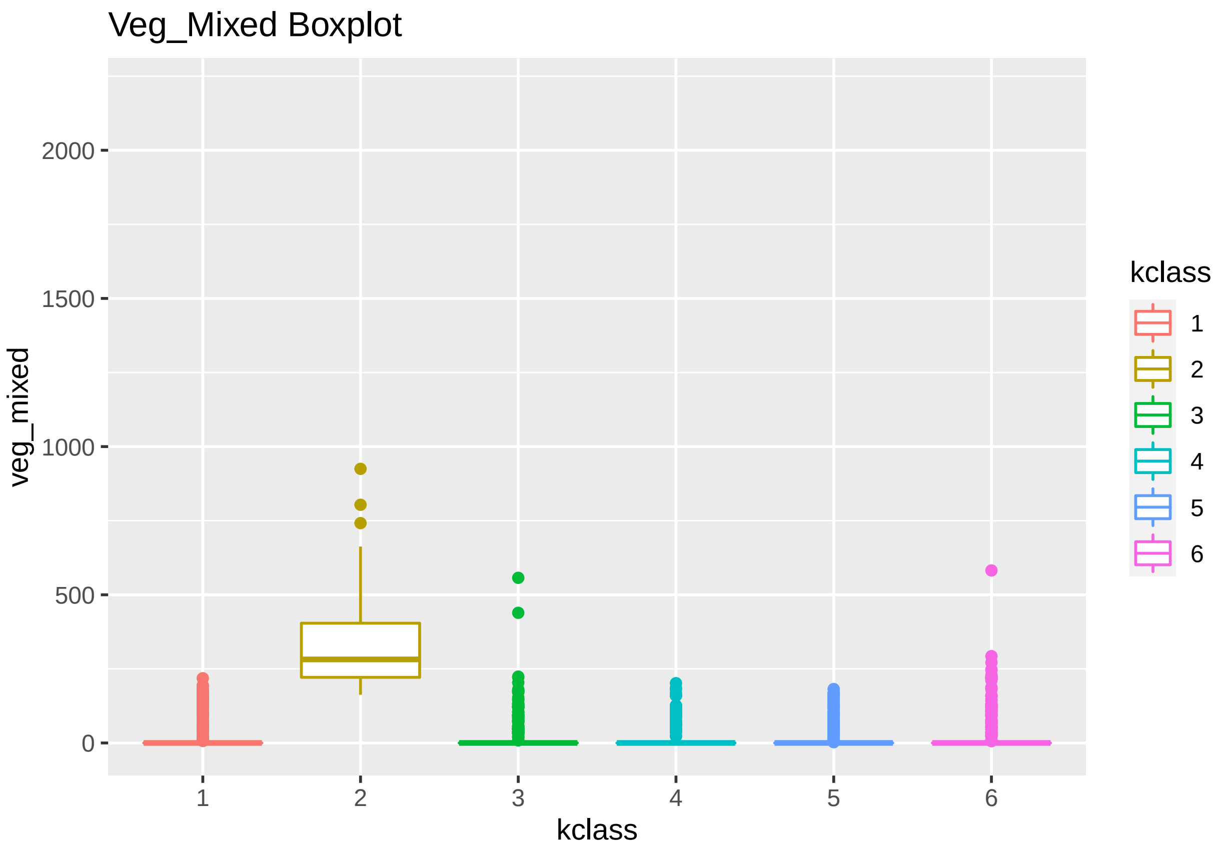
```
kclass=as.factor(D.km$cluster)

ggplot(Dresults.df, aes(x = kclass, y = veg_raw, color = kclass)) +
  geom_boxplot() + scale_y_continuous(limits=c(0,2200)) + ggtitle("Veg_Raw Boxplot")
```



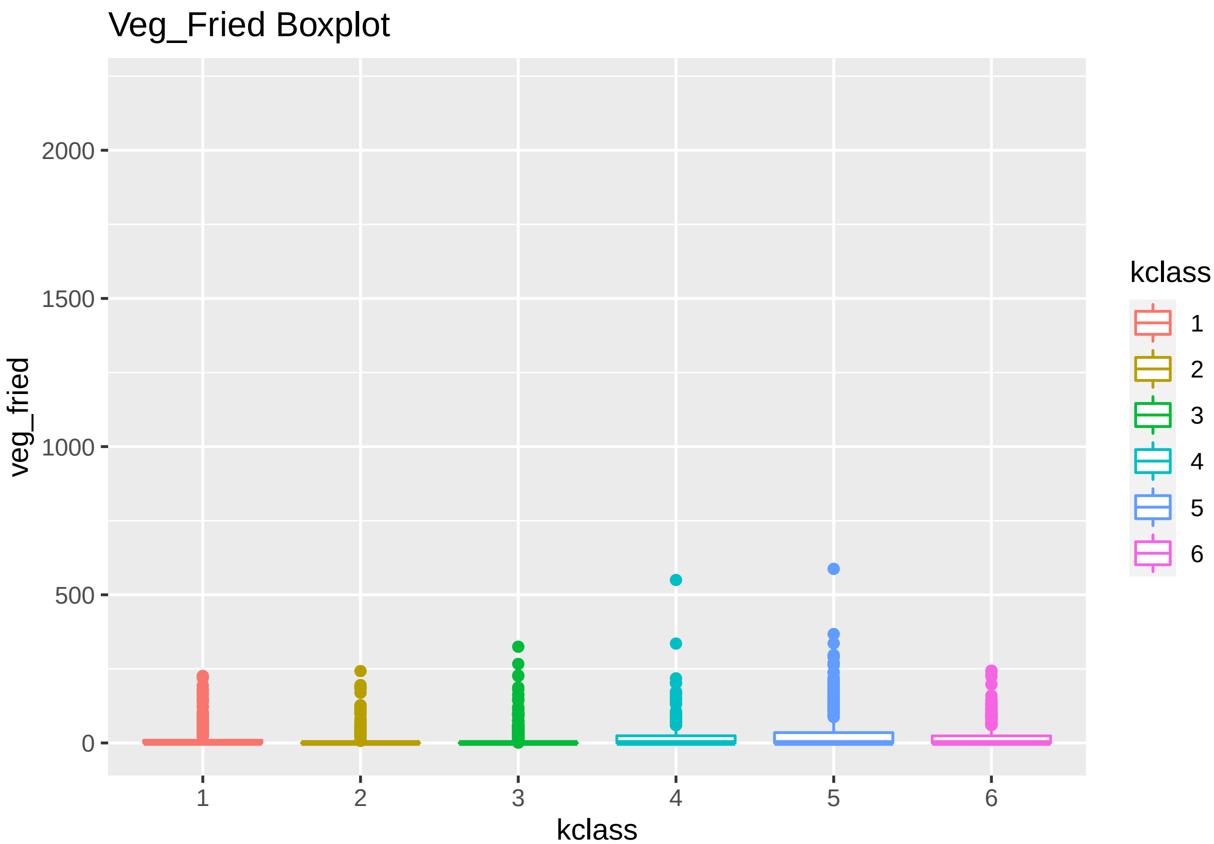

```r
ggplot(Dresults.df, aes(x = kclass, y = veg_cooked, color = kclass)) +
  geom_boxplot() + scale_y_continuous(limits=c(0,2200)) + ggtitle("Veg_Cooked Boxplot")
```



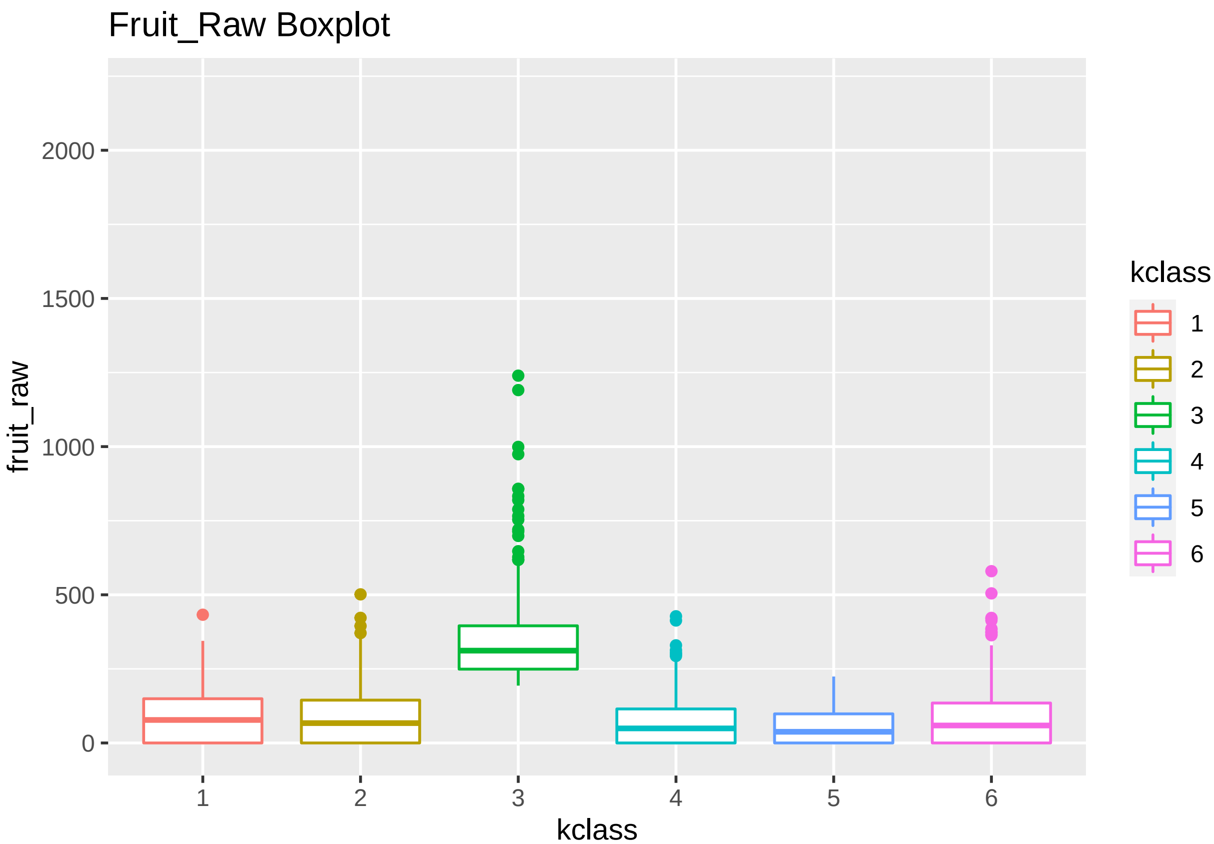

```r
ggplot(Dresults.df, aes(x = kclass, y = veg_mixed, color = kclass)) +
  geom_boxplot() + scale_y_continuous(limits=c(0,2200)) + ggtitle("Veg_Mixed Boxplot")
```



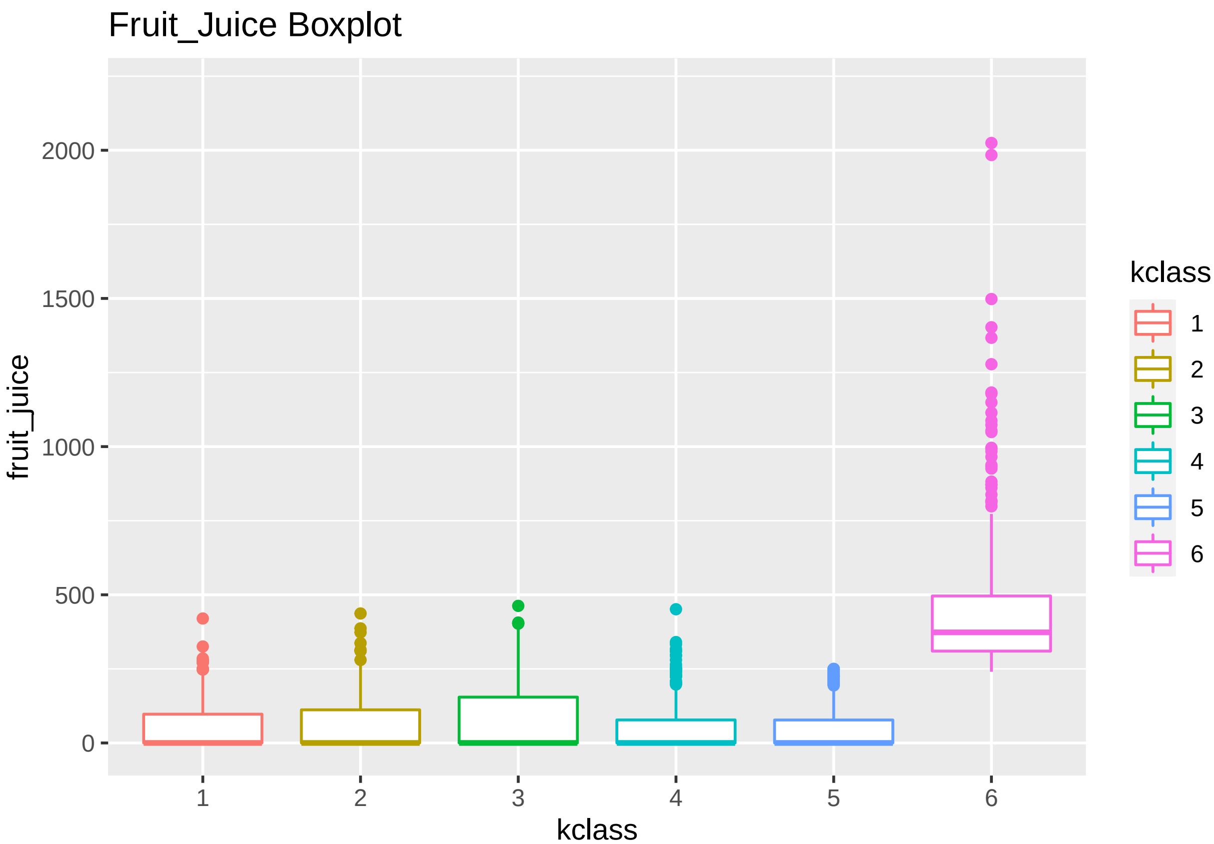

```
ggplot(Dresults.df, aes(x = kclass, y = veg_fried, color = kclass)) +
  geom_boxplot() + scale_y_continuous(limits=c(0,2200)) + ggtitle("Veg_Fried Boxplot")
```



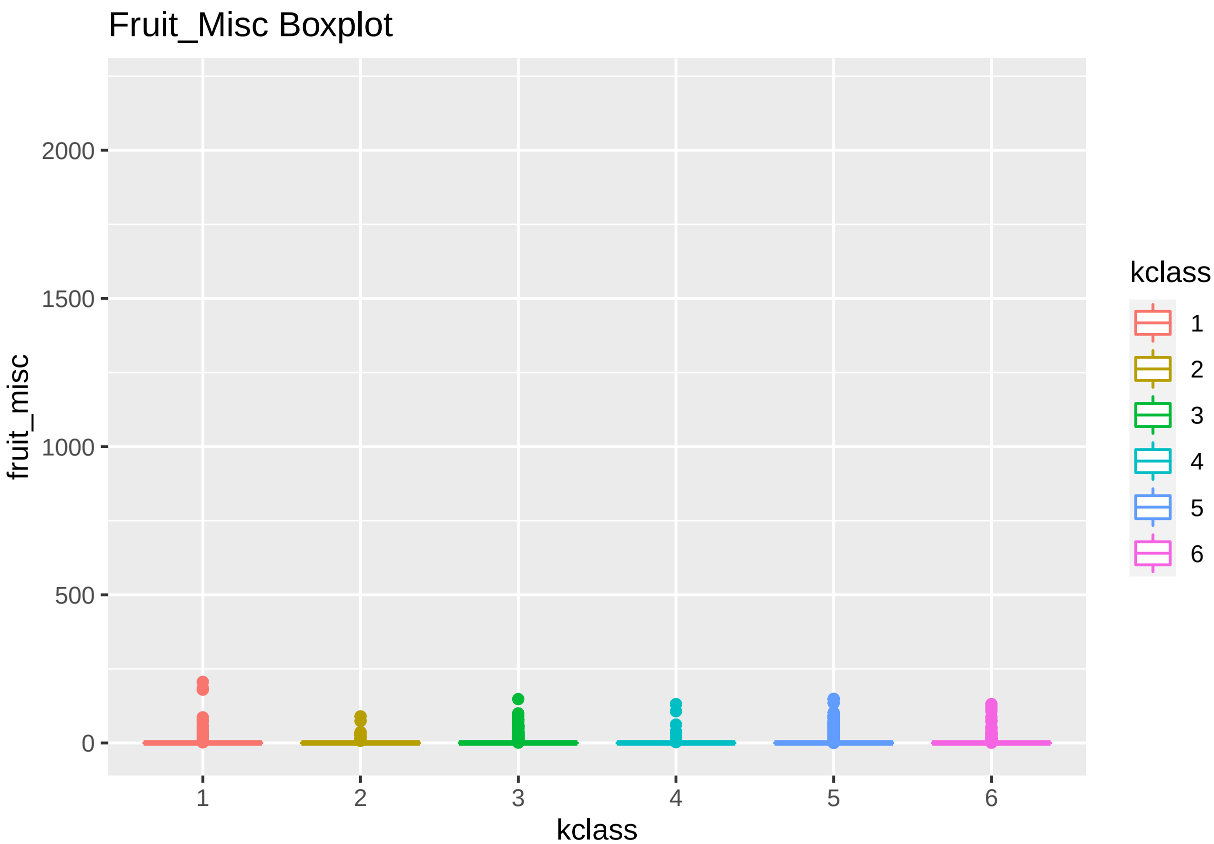

```
ggplot(Dresults.df, aes(x = kclass, y = fruit_raw, color = kclass)) +
  geom_boxplot() + scale_y_continuous(limits=c(0,2200)) + ggtitle("Fruit_Raw Boxplot")
```



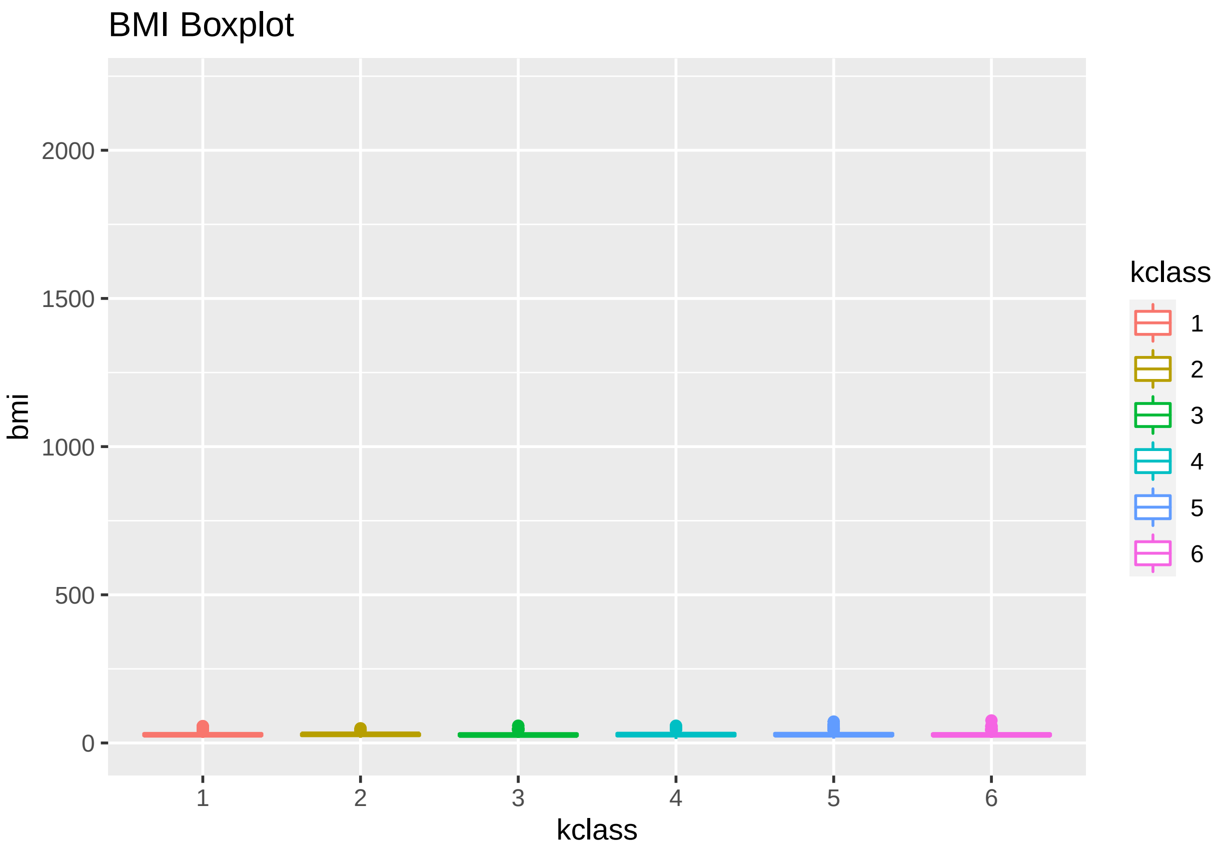

```
ggplot(Dresults.df, aes(x = kclass, y = fruit_juice, color = kclass)) +
  geom_boxplot() + scale_y_continuous(limits=c(0,2200)) + ggtitle("Fruit_Juice Boxplot")
```



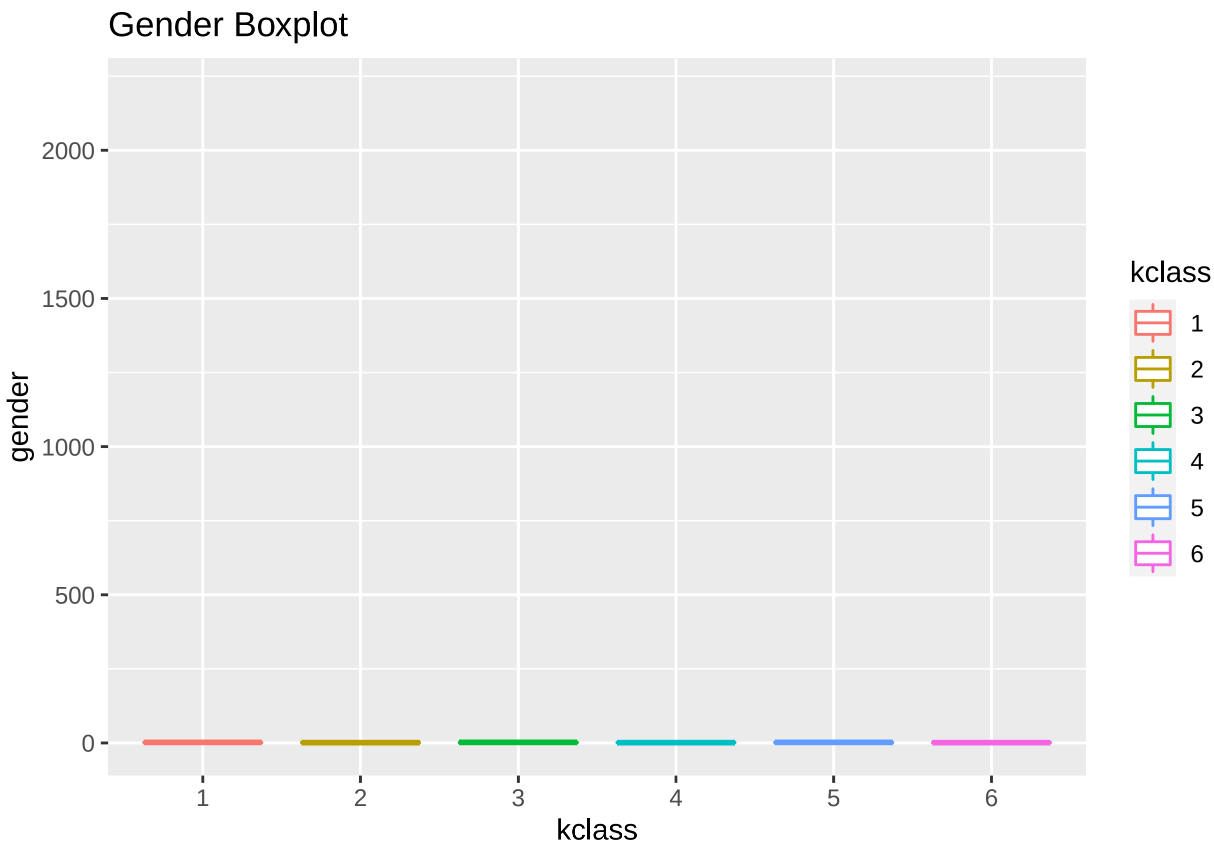

```r
ggplot(Dresults.df, aes(x = kclass, y = fruit_misc, color = kclass)) +
  geom_boxplot() + scale_y_continuous(limits=c(0,2200)) + ggtitle("Fruit_Misc Boxplot")
```



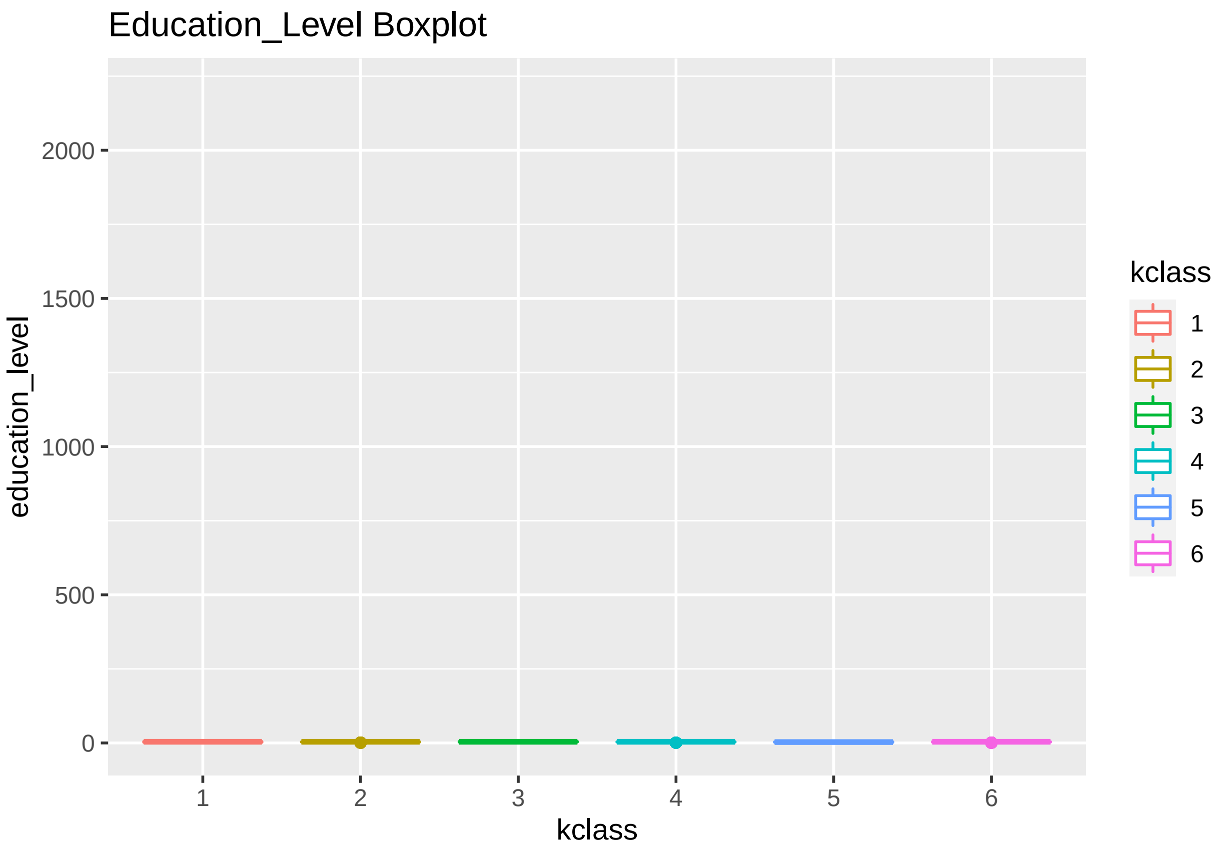

```r
ggplot(Dresults.df, aes(x = kclass, y = bmi, color = kclass)) +
  geom_boxplot() + scale_y_continuous(limits=c(0,2200)) + ggtitle("BMI Boxplot")
```




```r
ggplot(Dresults.df, aes(x = kclass, y = gender, color = kclass)) +
  geom_boxplot() + scale_y_continuous(limits=c(0,2200)) + ggtitle("Gender Boxplot")
```

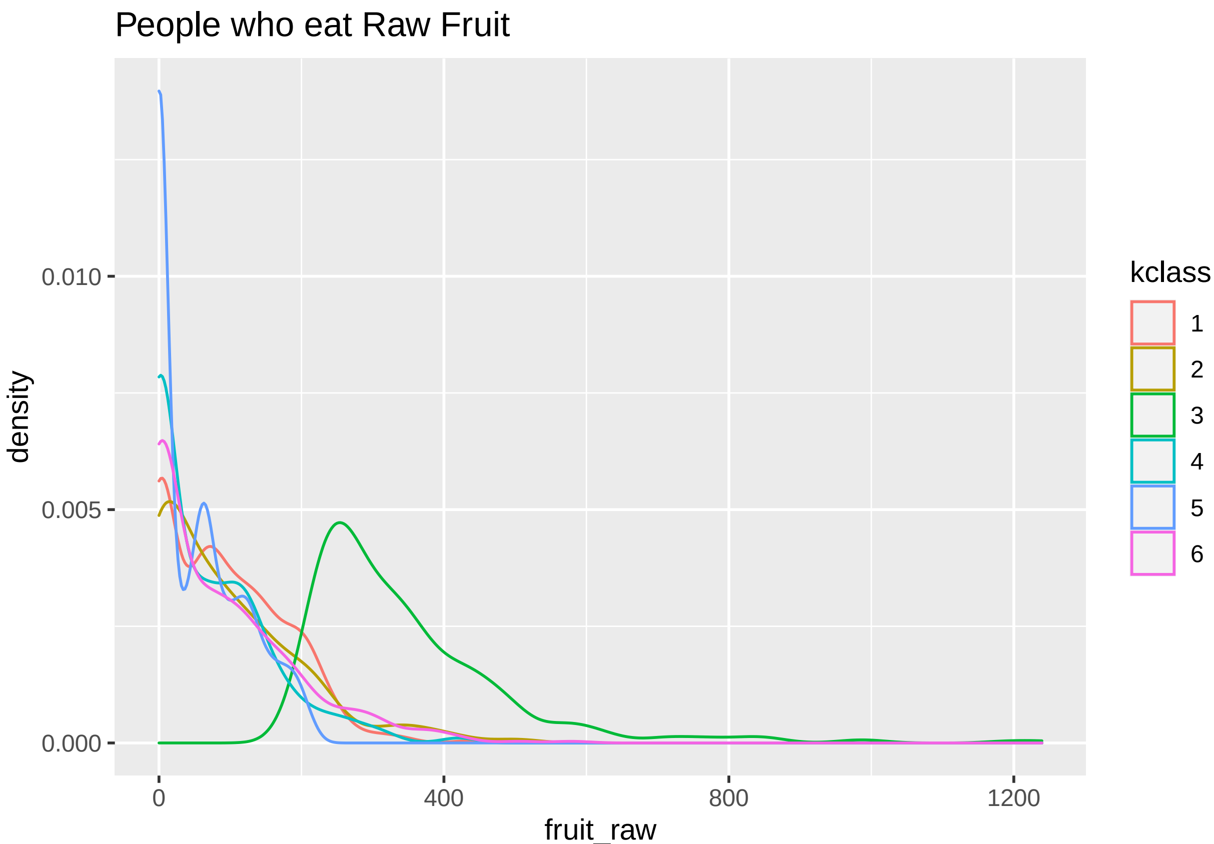



```
ggplot(Dresults.df, aes(x = kclass, y = education_level, color = kclass)) +
  geom_boxplot() + scale_y_continuous(limits=c(0,2200)) + ggtitle("Education_Level Boxplot")
```

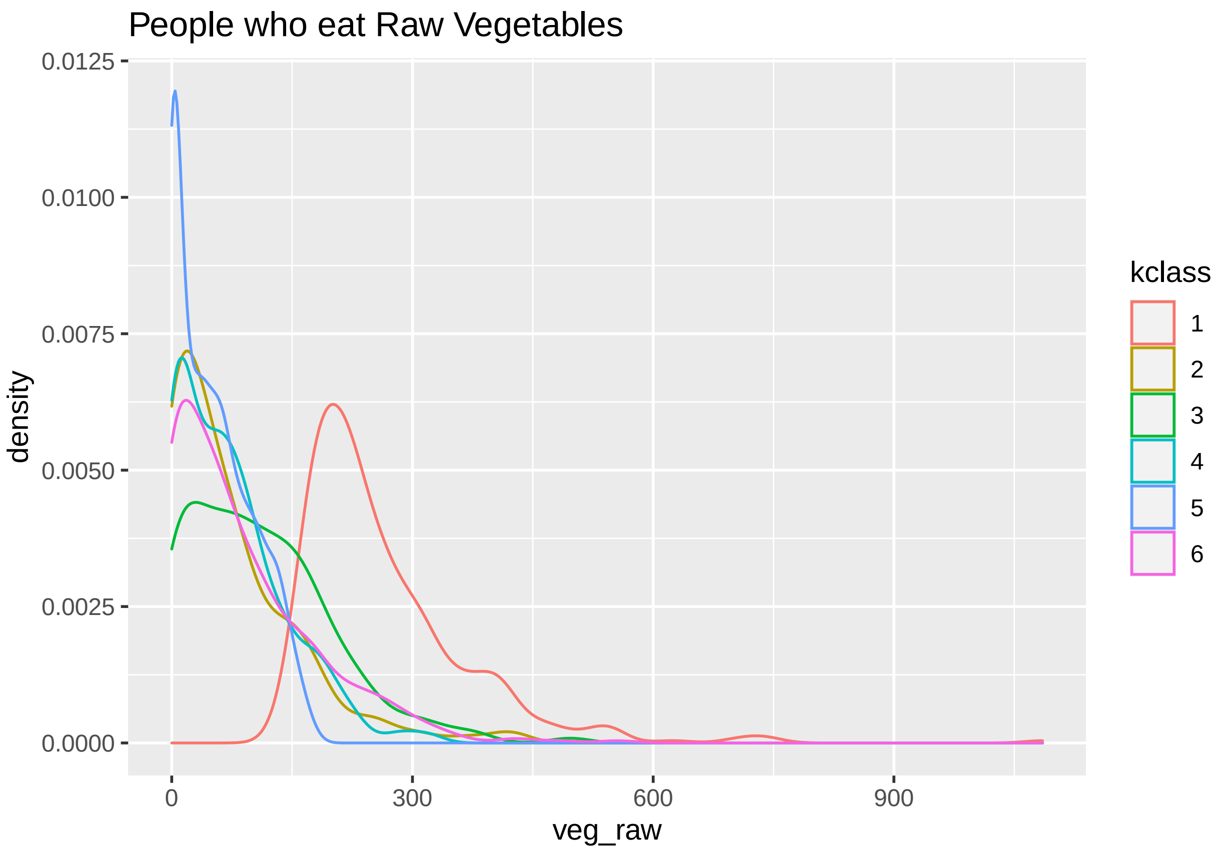


Exercise 6 : Make up a name for each of your clusters based on what you see in the cluster and describe why it fits. Make density plots for of each feature *by cluster* for the important clusters to illustrate why each names fits. For example, a good name for cluster might be "consumers who eat lots of raw vegetables."

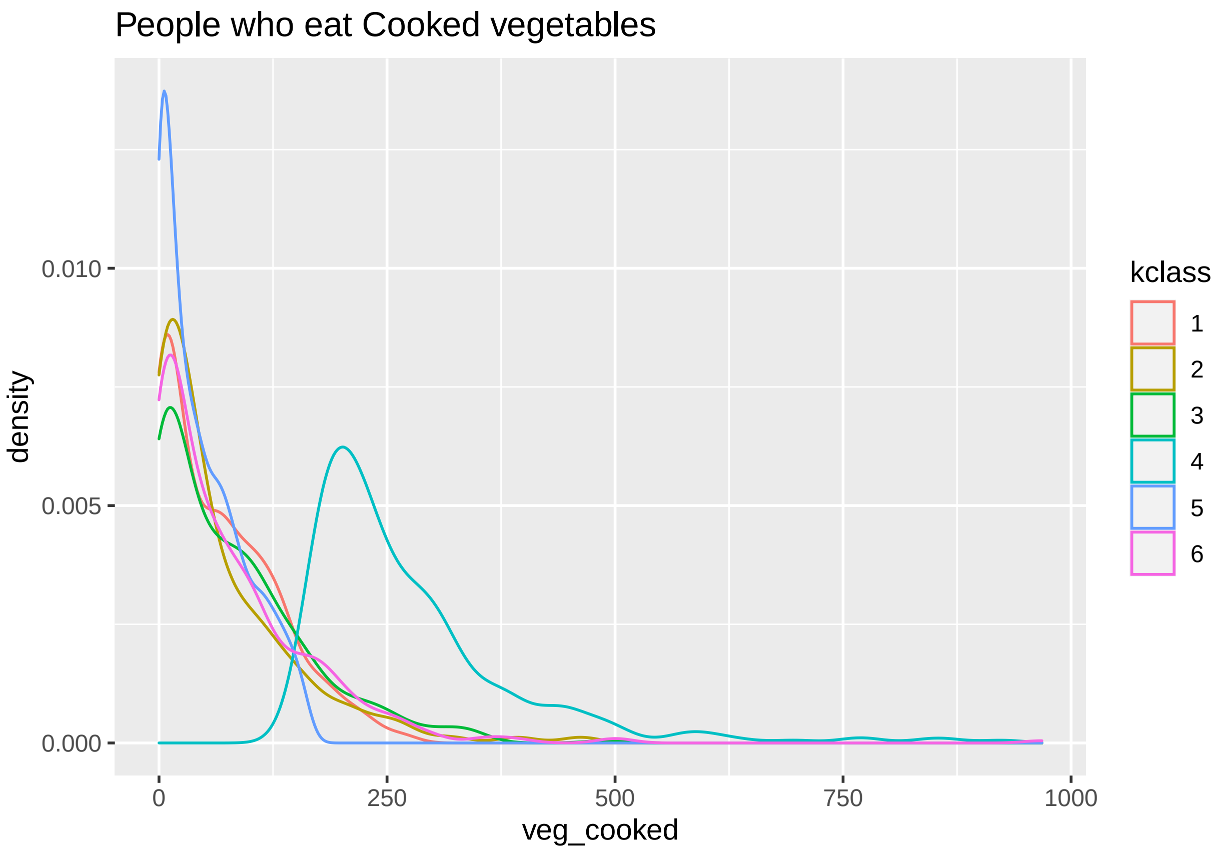
```
#Cluster 3 - People who like raw fruit. The data points in this cluster ate more raw
#fruit than other clusters.
ggplot(Dresults.df) + geom_density(aes(fruit_raw, colour= kclass)) +
  ggtitle("People who eat Raw Fruit")
```



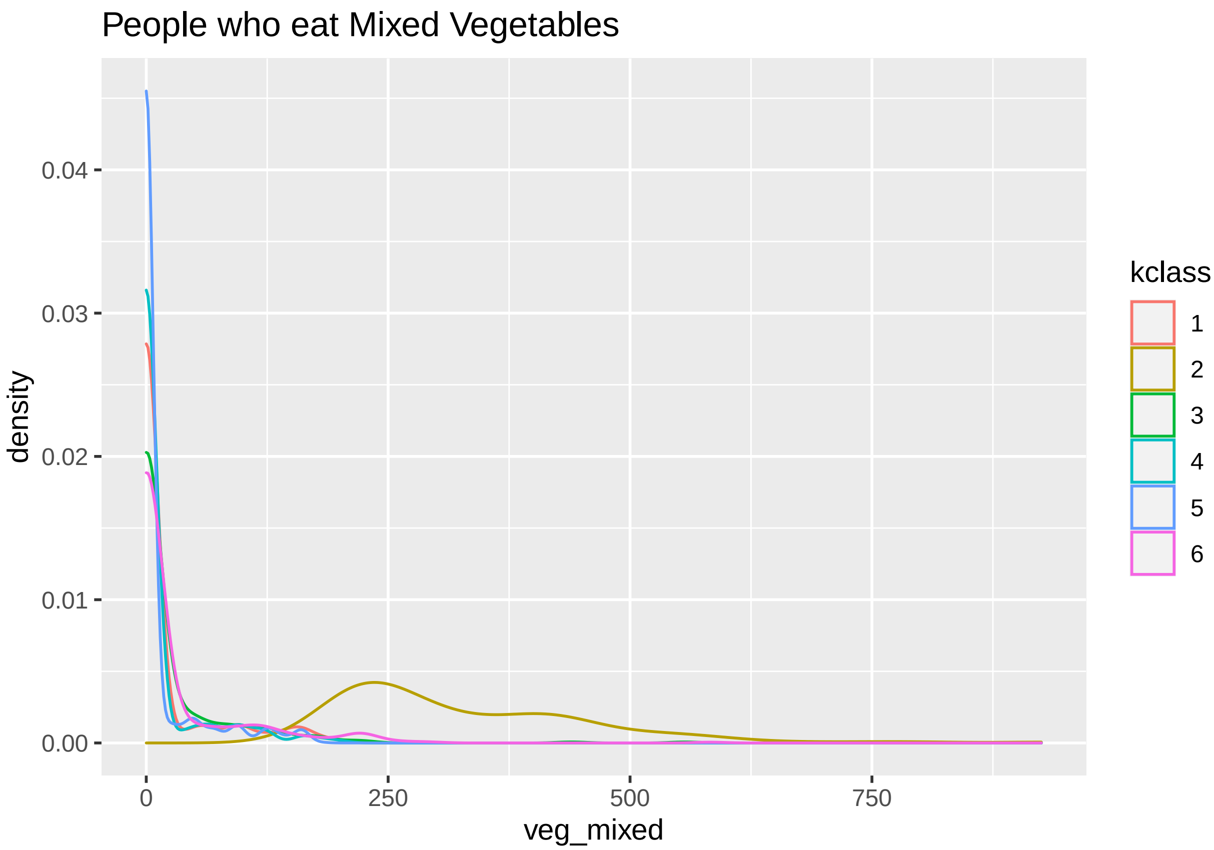

```
#Cluster 1 - People who like to eat raw vegetables. This cluster had the highest
#density of large amounts of raw vegetables.
ggplot(Dresults.df) + geom_density(aes(veg_raw, colour= kclass)) +
  ggtitle("People who eat Raw Vegetables")
```



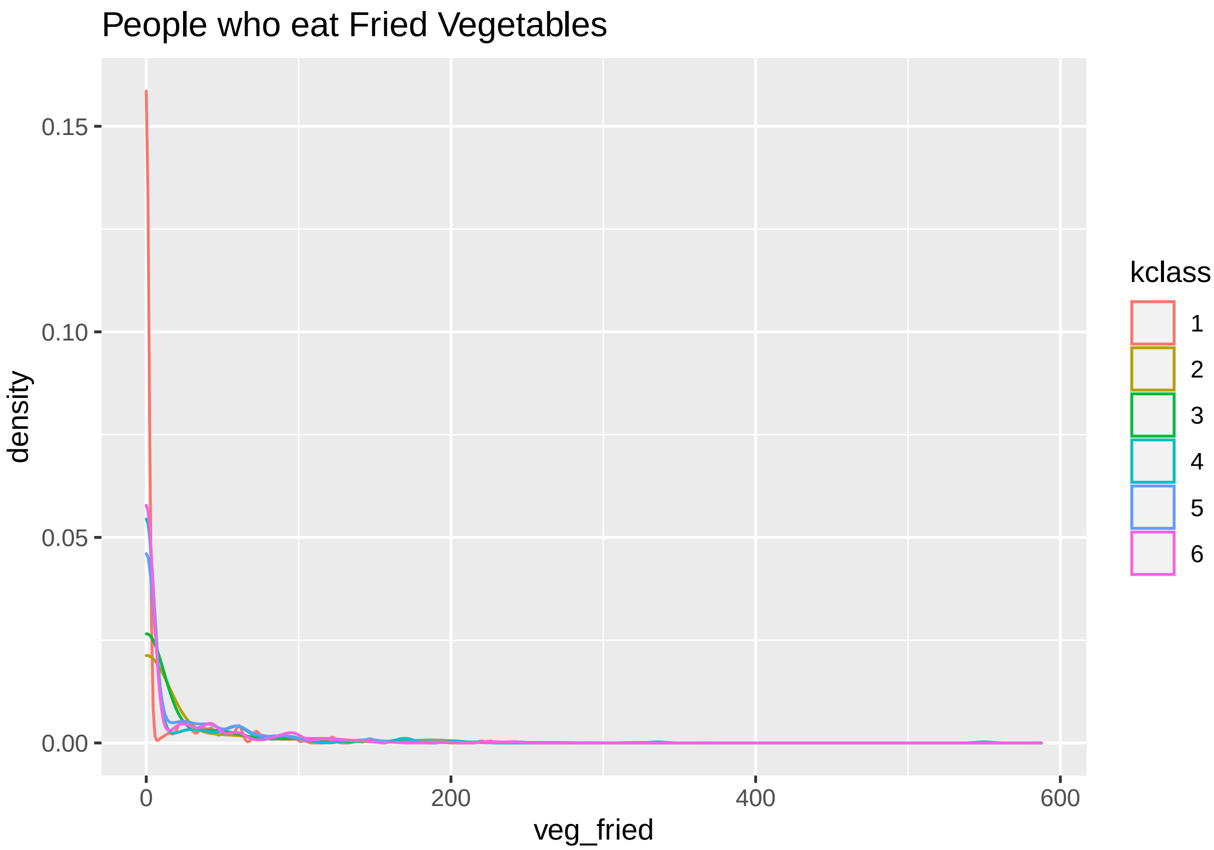

```
#Cluster 4 - People who like to each cooked vegetables. The peak was taller and
#at a higher amount of cooked veggies, indicating that this group eats more
#cooked vegetables than the rest.
ggplot(Dresults.df) + geom_density(aes(veg_cooked, colour= kclass)) +
  ggtitle("People who eat Cooked vegetables")
```



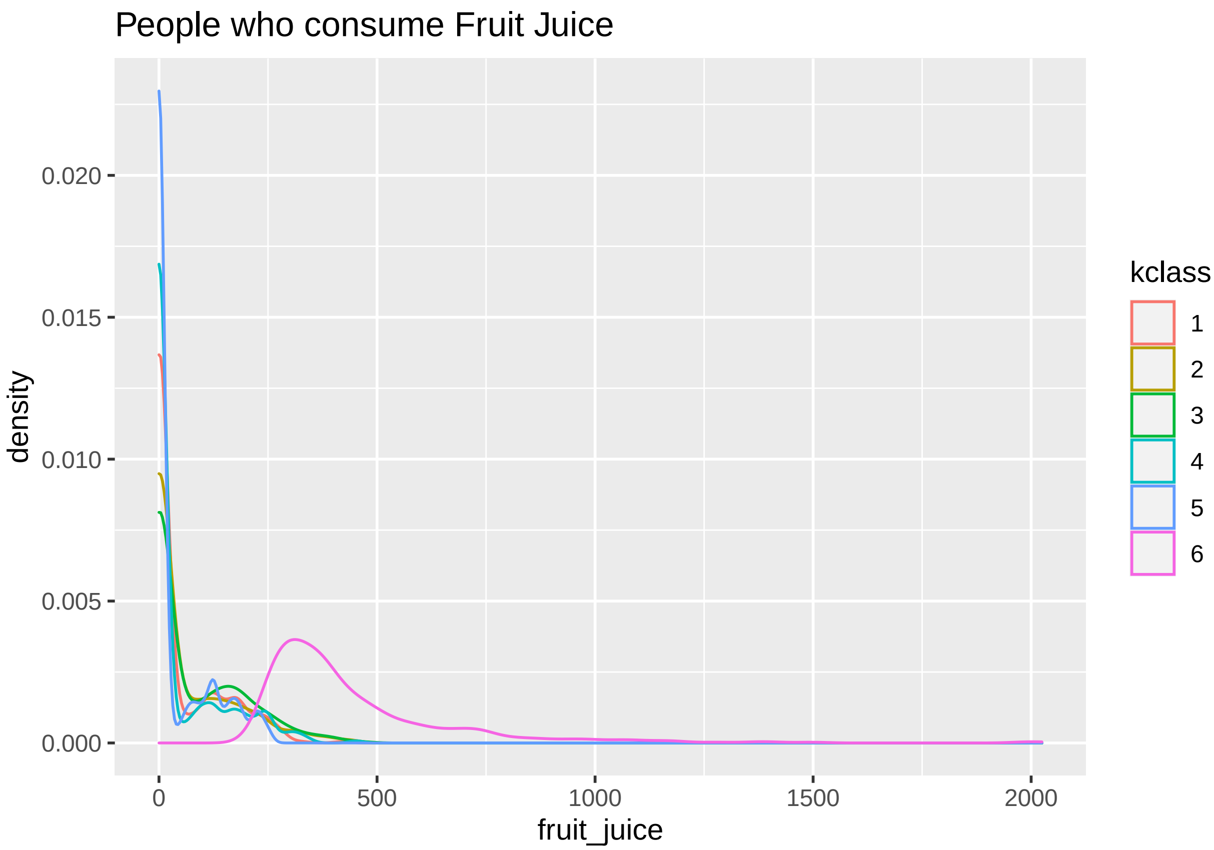

```
#Cluster 2 - People who like to eat Mixed vegetables. This cluster had a
#relatively higher density at a larger amount of cooked vegetables, indicating
#they eat more than others in the sample.
ggplot(Dresults.df) + geom_density(aes(veg_mixed, colour= kclass)) +
  ggtitle("People who eat Mixed Vegetables")
```



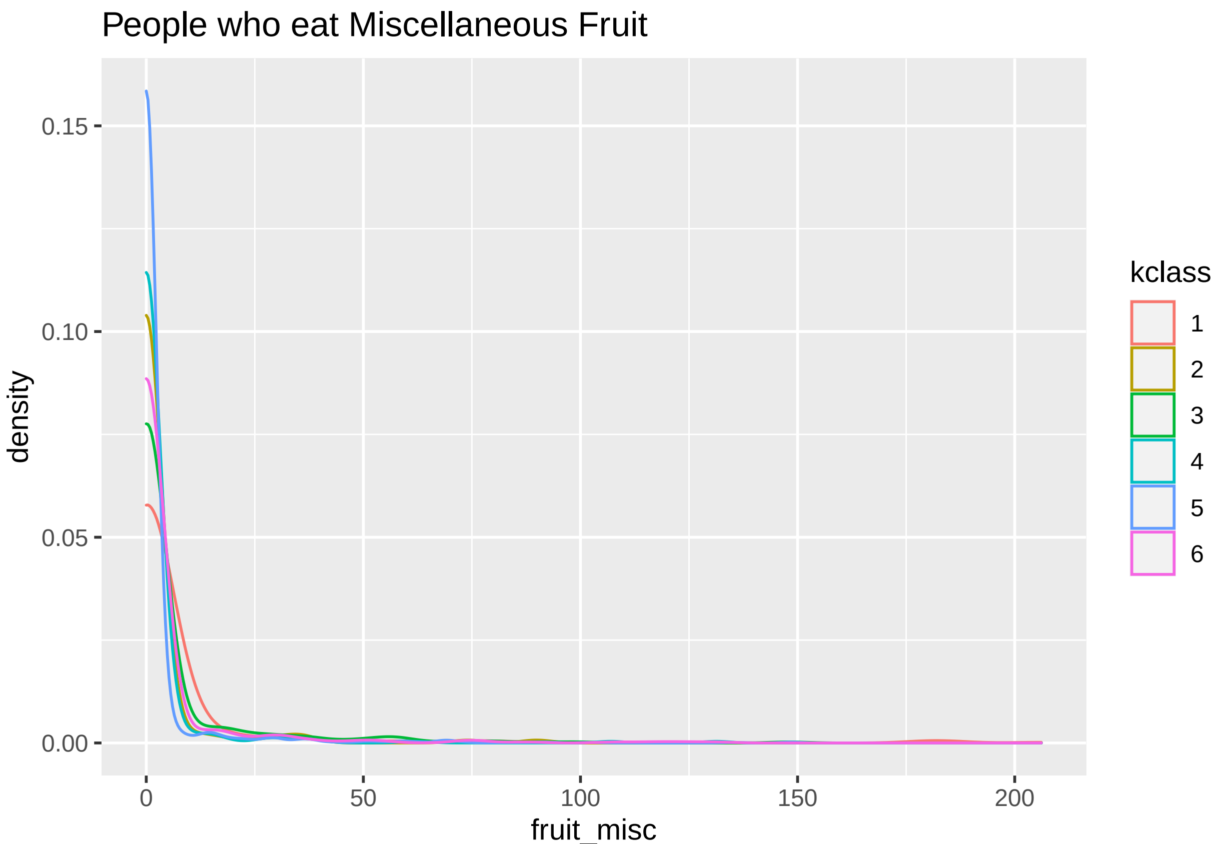

```
#Cluster 5 - People who eat a relatively fewer fruits and vegetables. While
#there was some indication for fried vegetables, there were also minor
#indications in all of the graphs, overall there were no predominant identifiers.
ggplot(Dresults.df) + geom_density(aes(veg_fried, colour= kclass)) +
  ggtitle("People who eat Fried Vegetables")
```




```r
#Cluster 6 - People who consume a lot of fruit juice. This cluster had a higher
#density at a larger amount indicting that they consume fruit juice more than
#others in the sample.
ggplot(Dresults.df) + geom_density(aes(fruit_juice, colour= kclass)) +
  ggtitle("People who consume Fruit Juice")
```



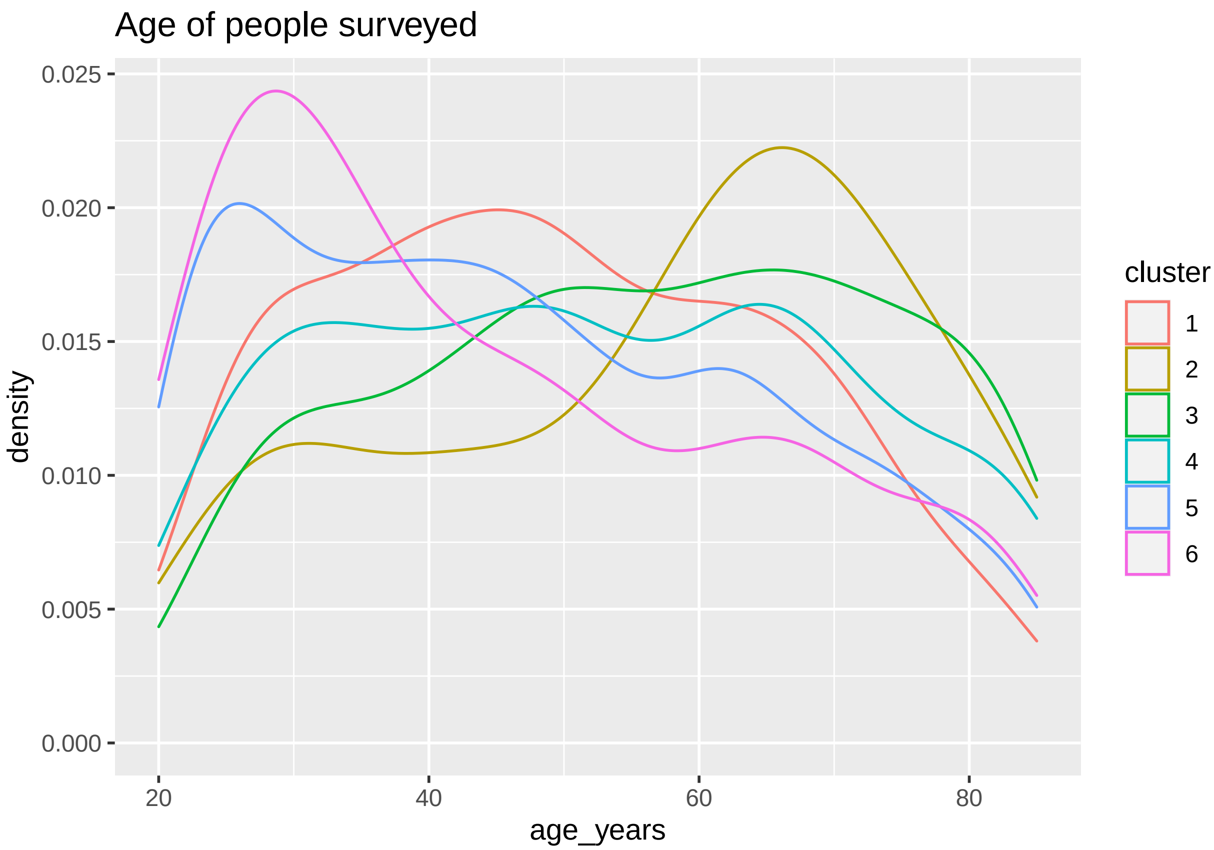

```
ggplot(Dresults.df) + geom_density(aes(fruit_misc, colour= kclass)) +
  ggtitle("People who eat Miscellaneous Fruit")
```




```
#ggplot(Dresults.df) + geom_density(aes(bmi, colour= kclass)) + ggtitle("BMI of people surveyed")
#ggplot(Dresults.df) + geom_density(aes(gender, colour= kclass)) + ggtitle("Gender in each Cluster")
#ggplot(Dresults.df) + geom_density(aes(education_level, colour= kclass)) + ggtitle("Education level in
```

Exercise 7 : Draw a density plot of`age_years` by cluster. Discuss any patterns with age that you observed by cluster. HINTS: you can create a variable 'cluster=as.factor(km$cluster)' then add color=cluster into the aesthetics of the 'geom_density()' applied to `age_years`. Which cluster tends to have the youngest people in it? Does `age_years` help explain any differences between the clusters?

```
plot.df <- raw.df %>% select(c('age_years'))
cluster=as.factor(D.km$cluster)

ggplot(plot.df) + geom_density(aes(age_years, color= cluster)) + ggtitle("Age of people surveyed")
```




```
#Cluster 6 and Cluster 5 consisted of mainly a younger population,
#Cluster 2 had the oldest.
#This helps explain why fruit juice was popular among cluster 6,
#as younger people consume more sugary/fruity drinks,
#so it makes sense that it was made up of mostly younger people
```

Exercise 8 :

Examine the summary of `D.df`.

```
summary(D.df)
```

```
##     veg_raw          veg_cooked       veg_mixed         veg_fried
##  Min.   :   0.00   Min.   :  0.00   Min.   :  0.00   Min.   :  0.0
##  1st Qu.:  18.89   1st Qu.: 11.65   1st Qu.:  0.00   1st Qu.:  0.0
##  Median :  65.90   Median : 50.00   Median :  0.00   Median :  0.0
##  Mean   :  93.81   Mean   : 79.05   Mean   : 31.51   Mean   : 21.2
##  3rd Qu.: 137.31   3rd Qu.:112.80   3rd Qu.:  0.00   3rd Qu.: 26.5
##  Max.   :1085.22   Max.   :968.00   Max.   :925.03   Max.   :587.5
##    fruit_raw        fruit_juice       fruit_misc           bmi
##  Min.   :   0.00   Min.   :   0.0   Min.   :  0.000   Min.   :13.36
##  1st Qu.:   0.00   1st Qu.:   0.0   1st Qu.:  0.000   1st Qu.:24.37
##  Median :  64.00   Median :   0.0   Median :  0.000   Median :27.70
##  Mean   :  98.96   Mean   : 100.4   Mean   :  3.157   Mean   :28.81
##  3rd Qu.: 148.50   3rd Qu.: 162.8   3rd Qu.:  0.000   3rd Qu.:31.90
##  Max.   :1239.47   Max.   :2024.7   Max.   :206.125   Max.   :76.07
##      gender      education_level
##  Min.   :1.000   Min.   :1.000
```

```
##   1st Qu.:1.000   1st Qu.:2.000
##   Median :2.000   Median :4.000
##   Mean   :1.532   Mean   :3.381
##   3rd Qu.:2.000   3rd Qu.:4.000
##   Max.   :2.000   Max.   :5.000
```

Notice that some variable ranges have large ranges of value (e.g. fruit_juice) while others have small ranges (e.g. gender). Having on variables with different ranges, can make K-means miss patterns in features with small ranges. One way to fix this problem, is to change each feature so it has mean 0 and variance 1. This is done by calculate the mean and standard deviation for the features, and a new feature vector is made by subtracting the mean (called centering) and dividing by the standard deviation (called scaling) of each entry.

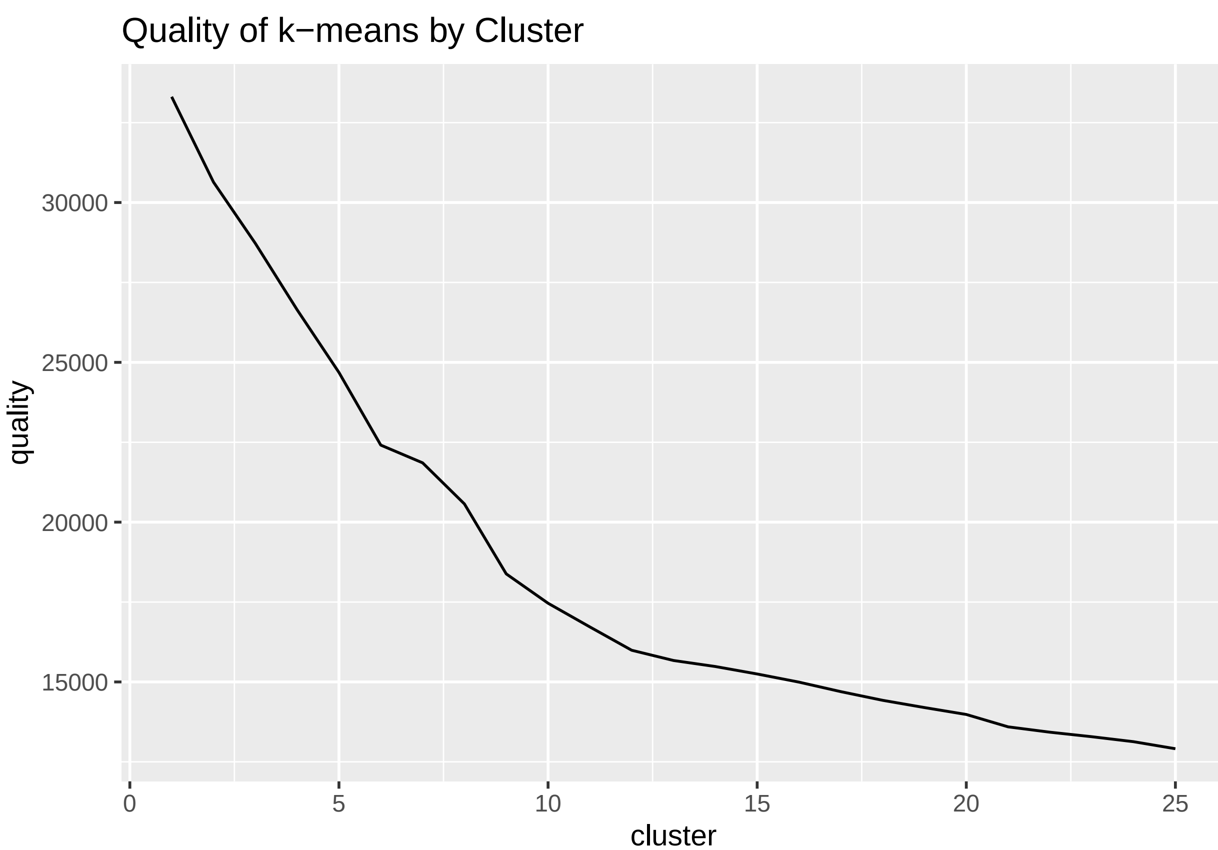
The command `scale` will do this this for you automatically. See '?scale' for more information.

Scale the matrix D.matrix and save as Dscaled.matrix. Then verify the means are 0 using the summary command.

```
Dscaled.matrix <- scale(D.matrix)
```

Redo Exercise 3 and using the scaled matrix; there is no need to give written answers to the redone 3, just provide the code. No need to describe the clusters. Look at the elbow plot.
What range of the clusters are reasonable according to the elbow plot for the scaled data?

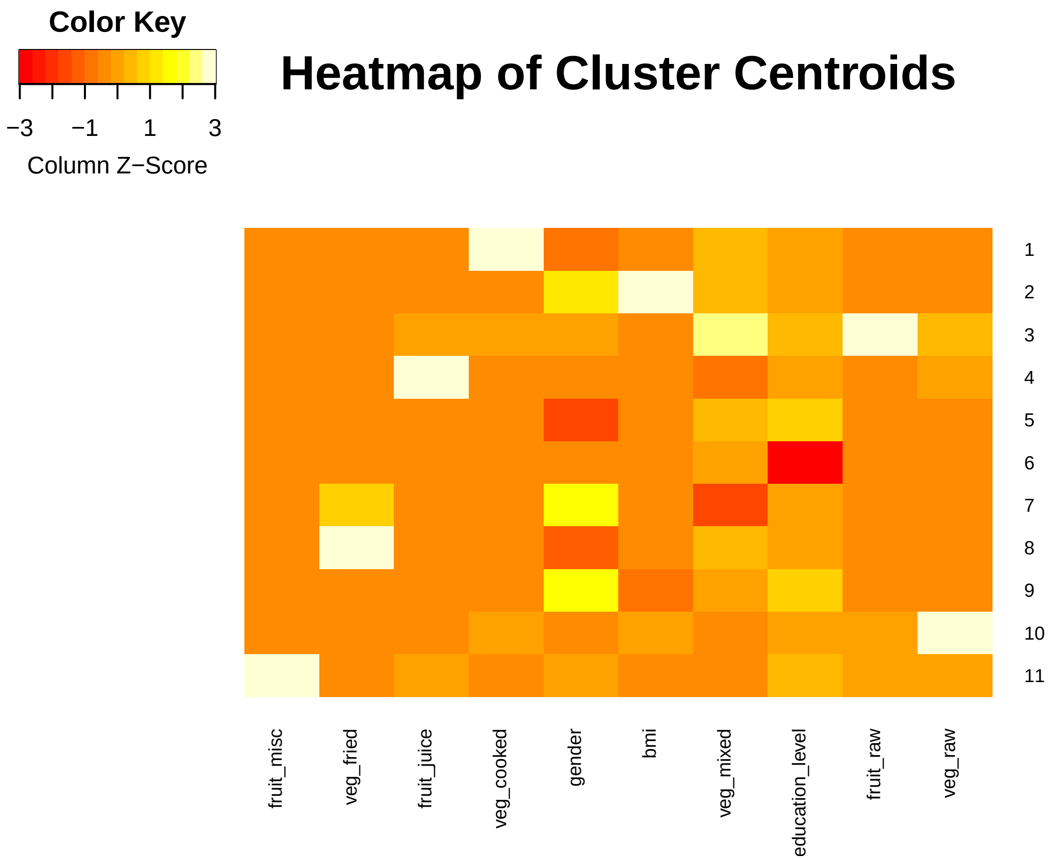
```
wssplot(Dscaled.matrix, nc=25)
```



Produce a k-means clustering of the scaled data using 11 clusters. Draw a heatmap of the cluster centers. Find a cluster that was found with scaling but was not found without scaling and that is influenced by `gender`, `bmi`, or `ed.level`. *Describe the cluster. Why does it appear only with scaling?*

```
#Creating the kmeans clustering
DS.km <- kmeans(Dscaled.matrix, centers=11)
```

```r
DSresults.df <-cbind.data.frame(Dscaled.matrix, kclass)

heatmap.2(DS.km$centers,
          dendrogram='none',
          main='Heatmap of Cluster Centroids',
          cexRow=0.75,
          cexCol=0.75,
          Rowv=FALSE,
          scale = 'col',
          tracecol=NA,
          density.info='none'
          )
```




```r
#From the heatmap, we can see that cluster 4 appears to be a grouping of a
#certain education level and gender. While their fruit/veggie consumption is
#generally average, the relationship is showing that the group with the highest
#education level is skewed towards one gender (Males). This cluster wasn't
#visible without scaling because the values were extremely low compared to the
#other numbers for fruits and vegetables eaten. The Kmeans could not recognize
#it as a cluster because it appeared that all the values were constant when
#compared to the other value fluctuations.
```

After you have completed these exercises and put your answers into this document, knit the pdf, and submit the result LMS.

APPENDIX

Reference: RStudio Cheatsheets

RStudio provides a number of convenient and useful "cheatsheets" through their web site https://www.rstudio.com/resources/cheatsheets/ and via the RStudio **Help** menu. Some cheatsheets you may want to have available this term include:

1. RStudio IDE download
2. RMarkdown download
3. Data Import download
4. Data Visualization download