ClusterAnalysis.R

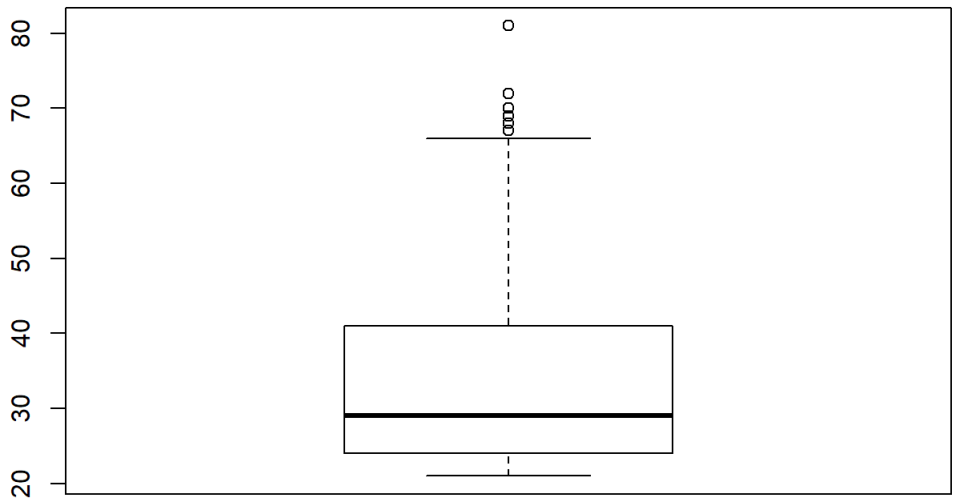
Mr.Perfectionist

Fri Apr 12 18:45:46 2019

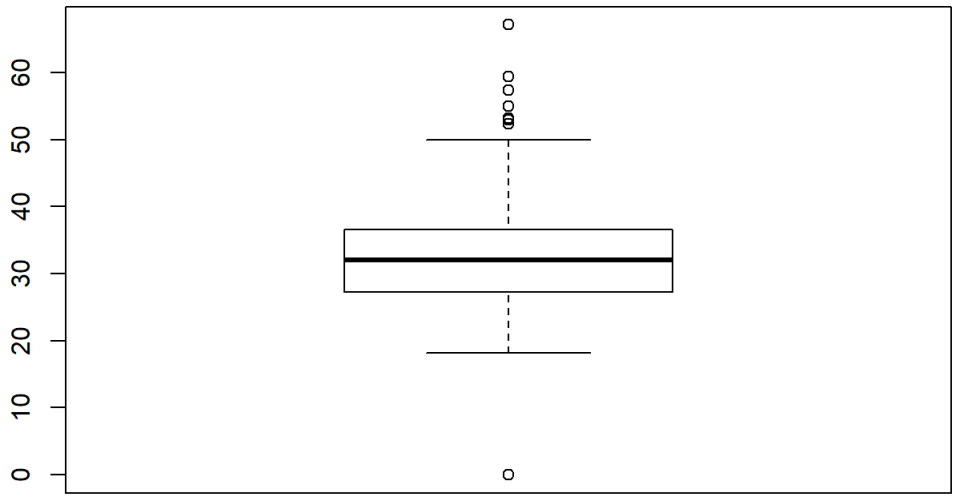
```
diabetes <- read.csv("F:/Courses/MVA/diabetes.csv")
str(diabetes)
```

```
## 'data.frame':    768 obs. of  9 variables:
##  $ Pregnancies             : int  6 1 8 1 0 5 3 10 2 8 ...
##  $ Glucose                 : int  148 85 183 89 137 116 78 115 197 125 ...
##  $ BloodPressure           : int  72 66 64 66 40 74 50 0 70 96 ...
##  $ SkinThickness           : int  35 29 0 23 35 0 32 0 45 0 ...
##  $ Insulin                 : int  0 0 0 94 168 0 88 0 543 0 ...
##  $ BMI                     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
##  $ Age                     : int  50 31 32 21 33 30 26 29 53 54 ...
##  $ Outcome                 : int  1 0 1 0 1 0 1 0 1 1 ...
```

```
attach(diabetes)
boxplot(Age)
```

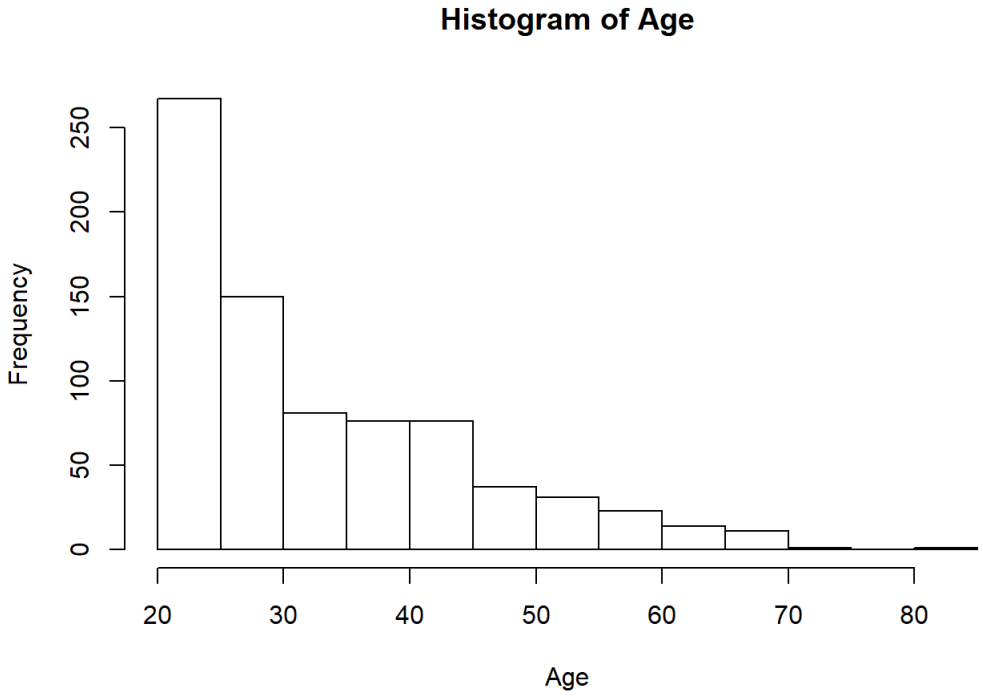



```
boxplot(BMI)
```




```
hist(Age)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

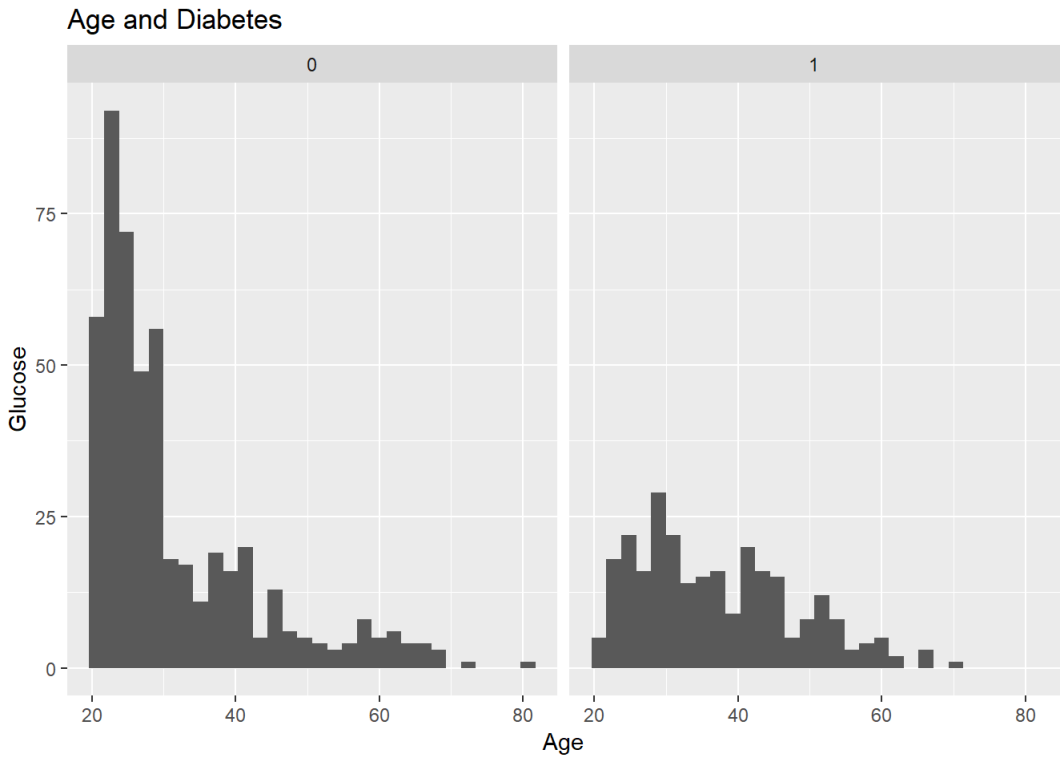



```
ggplot(diabetes, aes(Age, fill=Outcome)) + geom_histogram() + facet_wrap(~BMI) +labs(title = "Age and BMI" , x="Age", y="BMI")
```

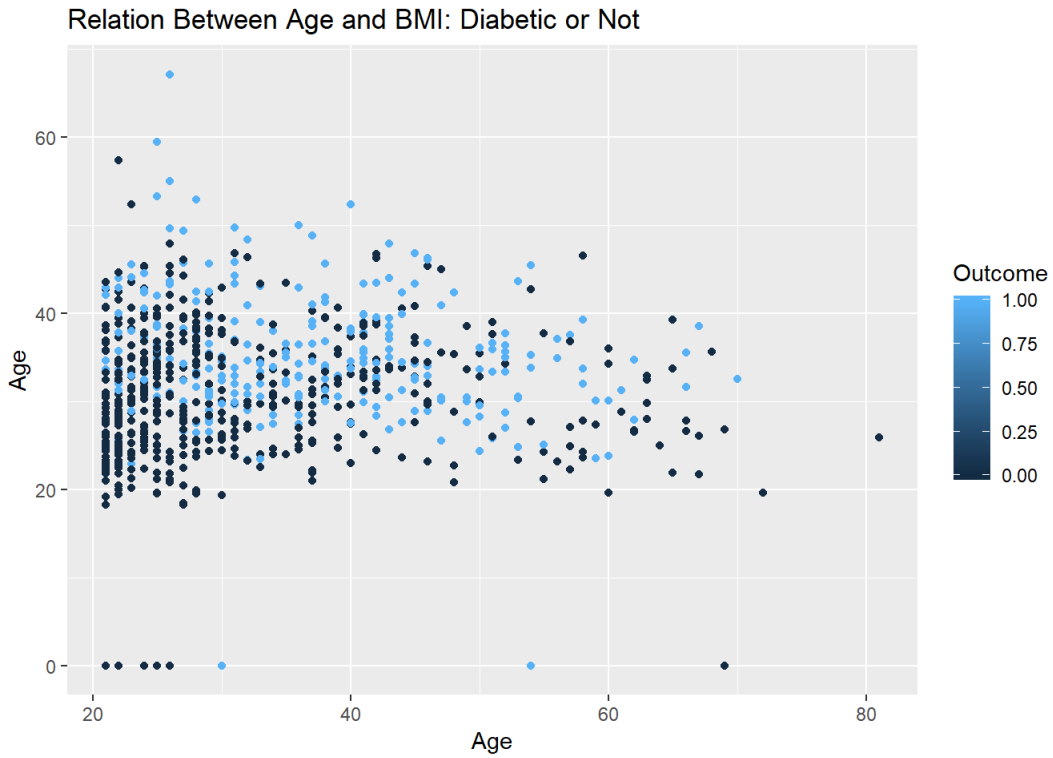
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


```
ggplot(diabetes, aes(Age, fill=Glucose)) + geom_histogram() + facet_wrap(~Outcome)+   labs(title = "Age and Diabetes" ,x="Age", y="Glucose")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



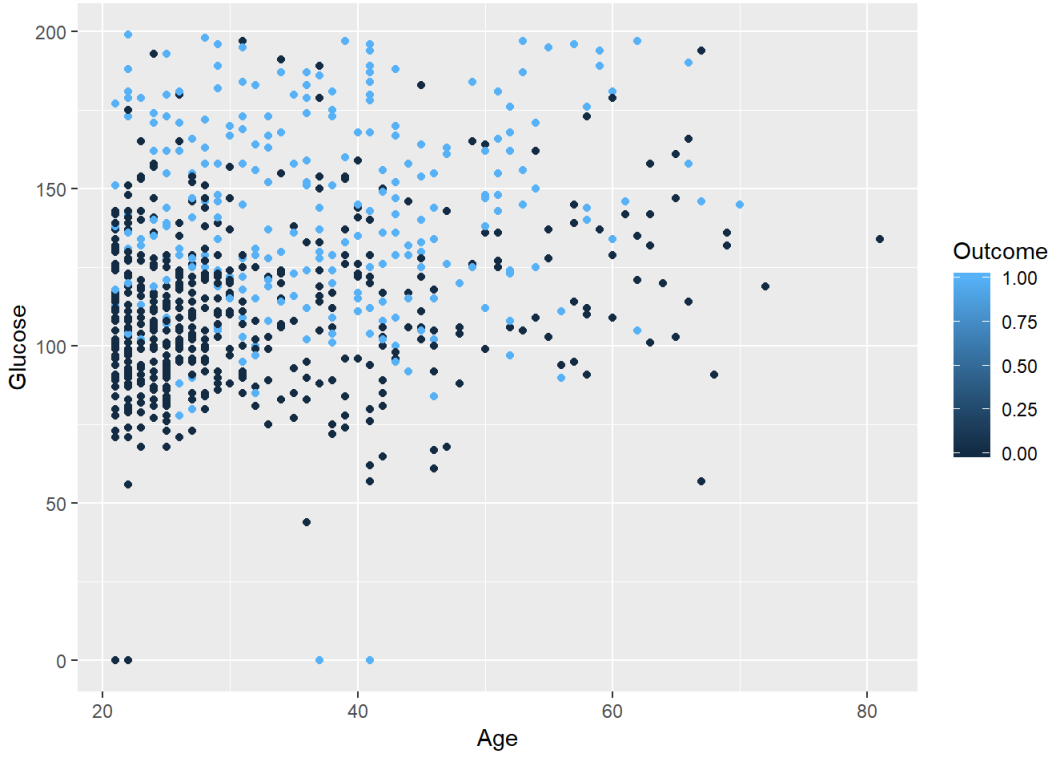

```
ggplot(diabetes, aes(Age,BMI,color=Outcome))  + geom_point() +
  labs(title = "Relation Between Age and BMI: Diabetic or Not" ,x="Age", y="Age")
```




```
summary(BMI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   27.30   32.00   31.99   36.60   67.10
```

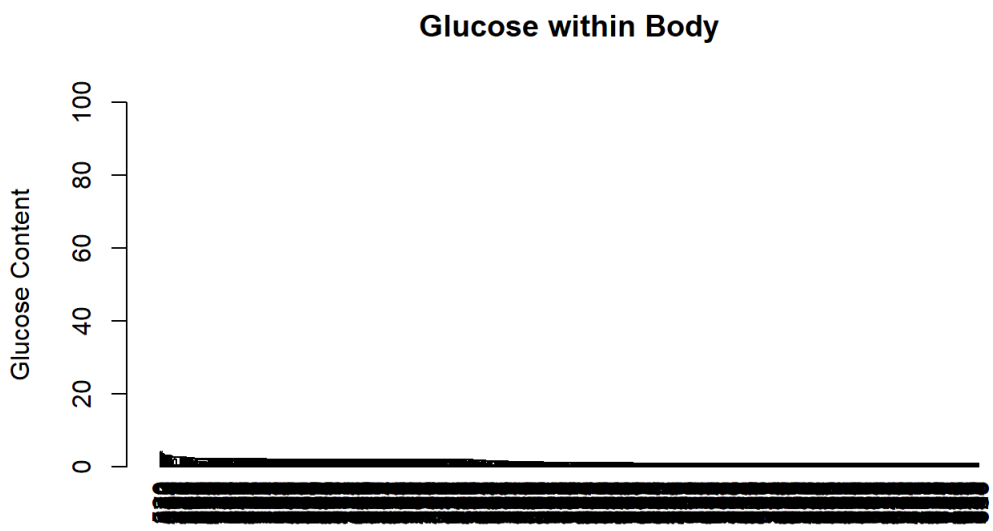
```
ggplot(diabetes, aes(Age,Glucose,color=Outcome))  + geom_point()
```




```
#Clustering

# Standardizing the data with scale()
# Clustering

#install.packages("cluster", lib="/Library/Frameworks/R.framework/Versions/3.5/Resources/library")
library(cluster)
```

```
## Warning: package 'cluster' was built under R version 3.5.3
```

```
require(graphics)

# Hierarchical cluster analysis

# Standardizing the data with scale() function
matstd.diabetes = scale(diabetes[,2:9])

# Creating a (Euclidean) distance matrix of the standardized data
dist.diabetes = dist(matstd.diabetes, method = "euclidean")

# Invoking hclust i.e cluster analysis by single linkage method
clusdiabetes.nn = hclust(dist.diabetes, method = "single")


# Plotting
# Create extra margin room in the dendrogram
par(mar=c(8, 4, 4, 2) + 0.1)
# "clusdiabetes.nn" is converted into a object of class "dendrogram"
# In order to allow better flexibility in the (vertical) dendrogram plotting.
plot(as.dendrogram(clusdiabetes.nn),ylab="Glucose Content",ylim=c(0,100),
     main="Glucose within Body")
```




```
#Horizontal Dendrogram

dev.new()
par(mar=c(5, 4, 4, 7) +0.1)
plot(as.dendrogram(clusdiabetes.nn), xlab= "Glucose Content", xlim=c(100,0),
     horiz = TRUE,main="Glucose within Body")



#K-Means Clustering

#install.packages("cluster", lib="/Library/Frameworks/R.framework/Versions/3.5/Resources/library")
library(cluster)
require(graphics)

attach(diabetes)
```

```
## The following objects are masked from diabetes (pos = 5):
##
##     Age, BloodPressure, BMI, DiabetesPedigreeFunction, Glucose,
##     Insulin, Outcome, Pregnancies, SkinThickness
```

```
# Standardizing the data with scale()
matstd.diabetes = scale(diabetes[,2:9])
# K-means, k=2, 3, 4, 5, 6
# Centers are numbers thus, 10 random sets are chosen

(kmeans2.diabetes = kmeans(matstd.diabetes,2,nstart = 10))
```

```
## K-means clustering with 2 clusters of sizes 485, 283
##
## Cluster means:
##      Glucose BloodPressure SkinThickness    Insulin        BMI
## 1 -0.4544383    -0.1870722    -0.2007519 -0.2754487 -0.3122796
## 2  0.7788077     0.3206007     0.3440447  0.4720587  0.5351789
##   DiabetesPedigreeFunction        Age    Outcome
## 1               -0.2263116 -0.2212295 -0.5760158
## 2                0.3878486  0.3791389  0.9871649
##
## Clustering vector:
##   [1] 2 1 2 1 2 1 1 1 2 1 1 2 2 2 2 1 2 1 1 2 2 1 2 2 2 2 2 1 1 1 1 2 1 1 1
##  [36] 1 1 2 2 2 1 1 1 2 1 2 1 1 2 1 1 1 1 2 2 1 2 2 2 1 1 2 1 1 1 1 2 1 1 1
##  [71] 2 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 2 1 2 1 1 1 2 2 1 1 1 1
## [106] 1 1 1 1 2 2 2 1 1 2 2 2 1 1 1 2 1 1 1 1 2 1 1 2 2 2 2 2 1 1 1 1 1 1 1
## [141] 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2 2 1 1 1 2 1 1 2 1 2 2 1 1 1 1 1 2 1 1 1
## [176] 2 1 2 1 2 1 1 1 1 1 2 2 2 2 2 1 1 2 2 1 2 1 1 2 2 1 1 1 1 1 1 2 2 1 2
## [211] 1 2 2 2 2 2 2 1 2 2 2 2 1 2 1 1 1 2 2 1 2 2 1 1 1 2 2 2 2 1 1 1 1 2 2
## [246] 2 1 2 2 1 1 1 1 1 2 2 1 1 2 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1
## [281] 2 1 1 2 1 1 2 2 1 1 1 2 2 2 1 2 2 1 2 1 2 2 1 2 1 1 2 1 2 2 1 1 2 1 2
## [316] 1 1 2 1 2 1 1 2 2 1 1 2 1 2 1 1 1 2 1 1 2 1 2 2 2 1 1 1 1 1 2 1 1 1 1
## [351] 1 1 1 1 1 2 2 2 1 2 2 1 2 2 2 1 1 1 1 2 2 1 1 1 1 2 1 1 2 2 1 1 1 1 1
## [386] 1 2 2 2 1 1 2 1 1 2 2 1 2 1 2 1 1 2 1 2 1 2 1 2 2 1 1 2 1 2 2 1 2 1 2
## [421] 2 1 1 1 2 2 1 2 2 2 1 1 1 1 1 1 1 1 1 2 1 1 2 1 2 1 1 2 1 1 1 1 1 1 1
## [456] 2 1 1 2 2 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 2 1 1 1 2 1 1 1 2 2 2 2 1 2 2
## [491] 1 1 1 2 1 1 1 1 2 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 2 1 1 2 1 1 1 2 1
## [526] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 1 1 2 2 1 2 2 1 1 1 1 1 1 1 1 1
## [561] 2 2 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 2 1 2 2 1 1 1 2 1 2 1 2 1 2 1 2 1 2
## [596] 2 1 1 2 1 1 1 1 1 2 1 1 2 1 2 1 1 2 2 1 2 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1
## [631] 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 2 2 2 2 1 1 1 1 1 1 2 1 2 1 2 1 2 2 2 2
## [666] 1 2 1 1 1 2 1 1 2 1 2 2 1 1 1 1 2 1 2 1 1 1 1 1 2 1 2 1 2 1 2 2 1 1 1
## [701] 1 2 2 1 1 1 1 1 2 2 2 1 2 1 1 2 2 1 1 2 1 1 2 1 1 1 1 1 1 1 2 1 2 1 1
## [736] 1 1 1 1 2 2 1 1 2 2 1 2 1 2 2 2 1 1 2 2 2 1 2 1 2 1 2 1 2 1 1 2 1
##
## Within cluster sum of squares by cluster:
## [1] 2649.553 2318.787
##  (between_SS / total_SS =  19.0 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
# Calculating the percentage of variation for 2 clusters
perc.var.2 = round(100*(1 - kmeans2.diabetes$betweenss/kmeans2.diabetes$totss),1)
names(perc.var.2) <- "Perc. 2 clus"
perc.var.2
```

```
## Perc. 2 clus
##           81
```

```
# Calculating the percentage of variation for 3 clusters
(kmeans3.diabetes = kmeans(matstd.diabetes,3,nstart = 10))
```

```
## K-means clustering with 3 clusters of sizes 206, 177, 385
##
## Cluster means:
##      Glucose BloodPressure SkinThickness    Insulin         BMI
## 1  0.7584959    0.32760882    0.87411823  0.9051419  0.61006485
## 2  0.4187319    0.02403169   -1.05397851 -0.6437585 -0.01410704
## 3 -0.5983525   -0.18634033    0.01684634 -0.1883480 -0.31993873
##   DiabetesPedigreeFunction        Age    Outcome
## 1                0.4892260  0.1764122  0.9070003
## 2               -0.1053984  0.9292849  0.4410590
## 3               -0.2133118 -0.5216217 -0.6880767
##
## Clustering vector:
##   [1] 1 3 2 3 1 3 3 3 1 2 3 2 2 1 1 2 1 2 3 1 1 2 2 1 1 1 2 3 2 2 2 1 3 3 3
##  [36] 3 2 1 1 1 3 2 2 1 2 1 2 3 1 3 3 3 3 1 1 3 1 1 2 3 3 2 3 3 2 3 1 2 3 3
##  [71] 1 3 2 3 3 3 3 3 2 3 3 3 3 3 2 3 3 3 1 3 3 3 3 2 3 1 3 3 3 1 2 3 3 3 3
## [106] 3 3 3 3 3 1 1 3 3 1 2 2 3 3 3 1 3 3 2 2 1 3 3 1 2 1 2 1 3 3 3 3 3 2 3
## [141] 2 3 3 2 1 3 3 3 2 3 1 3 1 1 2 1 3 3 3 1 3 3 1 3 2 1 3 2 3 3 2 1 3 3 3
## [176] 1 2 1 2 2 3 3 3 3 2 1 1 1 1 1 3 3 2 2 3 1 3 3 1 1 3 2 3 3 1 3 1 2 3 1
## [211] 3 1 2 1 1 1 1 3 1 2 1 2 3 1 3 3 3 1 1 3 2 1 3 3 3 2 1 1 1 3 3 3 2 1 1
## [246] 2 2 1 1 3 2 3 3 3 1 1 3 3 1 1 2 2 3 2 2 3 2 3 3 2 1 3 2 3 2 3 3 3 2 3
## [281] 2 3 3 2 2 2 1 1 3 3 3 1 1 1 2 1 1 3 1 2 2 1 3 2 2 3 2 3 1 1 3 3 1 3 1
## [316] 3 3 2 3 2 3 3 2 1 3 3 1 2 1 3 2 3 2 2 3 1 2 2 1 2 3 3 3 2 2 1 3 3 3 3
## [351] 3 2 3 3 3 2 1 2 3 1 1 2 2 2 1 3 2 3 3 1 1 3 3 3 1 1 3 3 2 1 3 3 3 3 3
## [386] 3 1 1 1 3 3 2 3 3 2 1 3 1 3 1 2 2 1 3 2 3 2 3 2 1 3 3 1 3 1 1 3 1 3 1
## [421] 1 3 3 3 1 1 3 1 1 1 3 3 3 3 3 2 3 2 3 3 1 3 3 2 2 1 3 3 1 3 3 2 3 2 3
## [456] 1 2 3 1 2 3 3 3 3 2 3 3 3 2 1 1 3 3 2 3 2 1 3 3 2 1 3 3 3 2 1 1 1 3 2
## [491] 3 3 3 1 3 2 3 3 1 1 3 3 3 3 3 3 1 3 3 2 2 3 2 3 3 1 1 2 3 2 3 3 3 2 3
## [526] 3 3 3 3 3 3 3 3 3 3 2 2 2 1 1 1 1 2 3 3 1 1 3 1 1 3 3 2 3 3 3 3 2 3 3
## [561] 2 1 3 3 3 3 3 3 3 3 1 3 3 3 3 1 3 3 2 2 1 1 3 2 2 1 3 2 3 1 3 1 3 2 3 1
## [596] 1 2 3 2 3 3 3 3 3 1 2 3 1 3 1 3 3 1 1 3 1 3 3 3 2 2 3 3 2 3 3 3 3 3 2 3
## [631] 2 3 3 3 3 3 2 2 3 1 3 3 3 2 3 3 1 1 1 1 3 3 3 3 3 3 1 3 1 2 1 2 1 1 1 1
## [666] 3 2 2 3 2 1 3 3 1 2 2 2 3 2 3 3 1 3 2 2 3 3 3 3 1 3 2 3 1 3 1 1 3 3 3
## [701] 3 2 1 2 3 3 3 3 2 1 1 3 1 3 3 1 1 2 3 2 3 3 1 3 2 3 3 3 2 3 1 2 1 3 2
## [736] 3 3 3 3 2 1 3 3 2 1 3 1 3 1 2 2 3 3 1 1 1 3 2 3 2 3 1 3 1 3 3 2 3
##
## Within cluster sum of squares by cluster:
## [1] 1470.482 1231.893 1665.643
##  (between_SS / total_SS =  28.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
perc.var.3 = round(100*(1 - kmeans3.diabetes$betweenss/kmeans3.diabetes$totss),1)
names(perc.var.3) <- "Perc. 3 clus"
perc.var.3
```

```
## Perc. 3 clus
##         71.2
```

```
# Calculating the percentage of variation for 4 clusters
(kmeans4.diabetes = kmeans(matstd.diabetes,4,nstart = 10))
```

```
## K-means clustering with 4 clusters of sizes 200, 167, 36, 365
##
## Cluster means:
##      Glucose BloodPressure SkinThickness    Insulin         BMI
## 1  0.7695794     0.3159016     0.8737683  0.9519413  0.61920632
## 2  0.4003451     0.4601465    -1.0306185 -0.6447924 -0.07857834
## 3 -0.1218083    -3.5358279    -1.1619988 -0.6864135 -0.79002569
## 4 -0.5928449    -0.0348904     0.1073742 -0.1588960 -0.22541851
##   DiabetesPedigreeFunction        Age    Outcome
## 1                0.4826136  0.1946488  0.9247099
## 2               -0.1103766  0.9703728  0.3229589
## 3               -0.2375576 -0.2377879  0.2002009
## 4               -0.1905144 -0.5271826 -0.6742010
##
## Clustering vector:
##   [1] 1 4 2 4 1 4 4 3 1 2 2 2 2 1 1 3 1 2 4 1 1 2 2 4 1 1 2 4 2 2 2 1 4 4 4
##  [36] 4 2 1 1 1 4 2 2 1 2 1 2 4 1 3 4 4 4 1 1 4 1 1 2 4 3 2 4 4 2 4 1 2 4 4
##  [71] 1 4 2 4 4 4 4 4 3 4 4 3 4 4 2 4 4 4 1 4 4 4 4 2 4 1 4 4 4 1 2 4 4 4 4
## [106] 4 4 4 4 4 4 1 1 4 4 1 2 2 4 4 4 1 4 4 2 2 1 4 4 1 2 1 2 1 4 4 4 4 4 2 4
## [141] 2 4 4 2 1 4 4 4 4 2 4 1 2 1 1 2 1 4 4 4 1 4 4 1 4 2 1 4 2 4 4 2 1 3 4 4
## [176] 1 2 1 2 2 4 4 4 4 4 2 1 1 1 1 1 4 4 2 3 4 1 4 4 1 1 4 2 4 4 1 4 1 2 4 1
## [211] 4 1 2 1 1 1 1 4 4 2 1 2 3 1 4 4 4 1 1 4 2 1 4 4 4 2 1 1 1 4 4 4 2 1 1
## [246] 2 2 1 1 4 2 4 4 4 4 1 4 4 4 1 1 2 3 4 2 2 4 3 4 4 3 1 4 2 4 2 4 4 4 2 4
## [281] 2 4 4 2 2 2 1 1 4 4 4 1 1 1 2 1 1 4 1 2 3 1 4 2 2 4 1 4 1 1 4 4 1 4 1
## [316] 4 4 2 4 2 4 4 2 1 4 4 1 2 1 4 2 4 3 2 4 1 3 2 1 2 4 4 4 2 2 1 4 3 4 4
## [351] 4 2 4 4 4 4 2 1 3 4 1 1 2 2 2 1 4 2 4 4 1 1 4 4 4 4 1 4 4 2 1 4 4 4 4 4
## [386] 4 1 1 1 4 4 2 4 4 2 1 4 1 4 1 2 2 1 4 2 4 2 4 2 1 4 4 1 4 1 1 4 1 4 1
## [421] 1 4 4 4 4 1 1 3 1 1 1 3 4 4 2 4 3 4 2 4 2 2 4 4 2 2 1 4 4 1 4 4 2 4 3 4
## [456] 1 2 4 1 2 2 4 4 4 2 4 4 4 3 1 4 4 4 2 4 2 1 4 4 2 1 4 4 4 3 1 1 1 4 2
## [491] 4 4 4 1 3 2 4 4 1 1 4 4 4 4 4 4 1 4 4 2 2 4 2 4 4 1 1 2 4 2 4 4 3 2 4
## [526] 4 4 4 4 4 4 4 4 3 4 3 2 2 1 1 1 1 2 4 4 1 1 4 1 2 4 4 2 4 4 4 4 2 4 4
## [561] 2 1 4 4 4 4 4 4 4 4 1 4 4 4 4 1 4 4 2 2 1 1 4 2 2 1 4 2 4 1 3 1 4 2 4 1
## [596] 1 2 3 2 4 4 3 4 1 3 4 1 4 1 4 4 1 1 4 1 4 2 4 2 3 4 4 2 4 4 4 4 4 2 4
## [631] 2 4 4 4 4 4 2 2 4 1 4 4 4 2 3 4 1 1 1 1 4 4 4 4 4 4 1 4 1 2 1 2 1 1 1 1
## [666] 4 2 2 4 2 1 4 4 1 2 2 2 4 2 4 4 1 4 2 2 4 4 4 4 1 2 2 4 1 4 1 1 3 4 4
## [701] 4 2 1 3 4 4 3 4 2 1 1 4 1 4 4 1 1 2 4 2 4 4 1 4 2 4 4 4 2 4 1 2 1 4 2
## [736] 4 4 4 4 2 1 4 4 2 1 4 1 4 1 2 2 4 4 1 1 1 4 2 4 2 4 1 4 1 4 4 2 4
##
## Within cluster sum of squares by cluster:
## [1] 1420.7967  929.2251  238.7477 1300.8939
##  (between_SS / total_SS =  36.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
perc.var.4 = round(100*(1 - kmeans4.diabetes$betweenss/kmeans4.diabetes$totss),1)
names(perc.var.4) <- "Perc. 4 clus"
perc.var.4
```

```
## Perc. 4 clus
##         63.4
```

```
# Calculating the percentage of variation for 5 clusters
(kmeans5.diabetes = kmeans(matstd.diabetes,5,nstart = 10))
```

```
## K-means clustering with 5 clusters of sizes 167, 36, 150, 193, 222
##
## Cluster means:
##      Glucose BloodPressure SkinThickness    Insulin         BMI
## 1  0.8801718    0.29772950     0.8192173  1.0172889  0.54876910
## 2 -0.1218083   -3.53582785    -1.1619988 -0.6864135 -0.79002569
## 3  0.4294967    0.47984899    -0.9852209 -0.6438468 -0.01597018
## 4 -0.3657023    0.13803265     0.8241788  0.1215129  0.46865950
## 5 -0.6146287   -0.09481384    -0.4786518 -0.3245556 -0.68134807
##   DiabetesPedigreeFunction        Age    Outcome
## 1                0.6153715  0.2503984  1.1390202
## 2               -0.2375576 -0.2377879  0.2002009
## 3               -0.1243568  1.0900393  0.4704358
## 4               -0.1718490 -0.3108266 -0.5795548
## 5               -0.1909666 -0.6160925 -0.7033103
##
## Clustering vector:
##   [1] 1 5 3 5 1 5 4 2 1 3 5 3 3 1 1 2 1 3 4 1 4 3 3 4 1 1 3 5 3 3 3 1 5 5 4
##  [36] 4 3 1 4 1 4 3 3 1 3 1 5 5 4 2 5 5 5 1 1 5 1 4 3 4 2 3 5 4 3 5 1 3 5 4
##  [71] 1 4 3 4 4 5 5 4 2 5 5 2 4 5 3 4 4 4 1 5 5 4 4 3 5 4 5 5 5 1 3 5 5 5 5
## [106] 4 5 4 4 4 1 1 4 5 1 3 3 5 5 5 1 4 4 3 3 4 4 4 1 3 1 3 1 4 5 4 5 5 5 4
## [141] 3 4 4 3 4 5 4 4 3 5 4 5 1 1 3 1 5 5 5 1 4 4 4 5 3 1 4 5 5 5 3 1 2 4 5
## [176] 1 5 1 3 3 5 4 5 5 3 1 1 1 1 1 5 4 3 2 5 1 5 5 1 1 5 3 5 5 4 5 1 3 4 1
## [211] 5 4 3 1 1 1 1 4 1 3 1 3 2 1 5 4 5 1 1 4 3 1 5 5 5 3 1 1 1 5 5 4 5 1 4
## [246] 3 3 1 4 5 5 5 5 4 1 4 4 5 1 1 3 2 4 3 3 4 2 4 5 2 1 5 5 4 3 4 5 5 3 5
## [281] 3 4 4 3 3 3 1 1 5 4 4 1 1 1 3 4 1 4 1 3 2 1 4 3 3 4 1 5 1 1 5 4 1 5 1
## [316] 4 5 3 4 3 5 4 3 1 4 5 1 3 4 4 3 5 2 3 5 4 2 3 1 3 5 5 5 3 3 4 5 2 5 4
## [351] 5 5 4 5 5 3 1 2 4 1 1 3 3 3 4 4 3 5 5 1 1 5 4 4 4 1 5 4 3 4 4 5 5 5 5
## [386] 5 1 3 1 4 4 3 5 5 3 1 4 4 5 1 3 3 1 4 3 4 3 5 3 1 4 4 1 5 1 1 5 1 5 1
## [421] 4 5 4 5 1 1 2 1 4 1 2 5 5 5 5 2 4 5 5 5 3 5 4 3 3 1 5 4 4 5 5 3 4 2 4
## [456] 1 3 5 1 3 5 5 4 5 5 5 5 4 2 4 4 4 4 3 5 3 1 5 4 3 1 4 5 4 2 1 1 1 5 3
## [491] 4 4 4 1 2 3 5 5 1 1 5 4 4 4 4 5 1 5 5 3 3 5 3 5 5 1 1 3 5 3 5 4 2 3 5
## [526] 5 5 5 4 5 5 5 4 2 4 2 3 3 4 1 1 1 3 4 4 1 1 4 4 3 5 4 3 5 4 4 4 3 4 5
## [561] 3 1 4 5 5 5 4 4 4 1 5 5 4 5 1 4 5 3 5 1 1 5 3 3 1 5 3 5 1 2 1 4 3 5 4
## [596] 1 3 2 3 5 5 2 4 1 2 4 1 5 4 5 5 1 1 4 1 5 5 5 3 2 4 5 3 4 5 4 5 5 3 5
## [631] 3 4 5 5 5 3 3 5 1 5 5 5 3 2 4 1 1 1 1 5 5 4 4 5 4 1 5 1 3 1 3 1 1 1 4
## [666] 4 3 3 4 4 1 5 4 4 3 3 3 5 3 5 5 1 4 3 3 4 5 5 5 1 5 3 4 1 5 1 1 2 4 5
## [701] 4 3 1 2 5 4 2 4 3 1 1 4 1 5 5 1 1 3 4 3 5 4 1 4 3 4 4 4 5 5 3 3 1 4 3
## [736] 4 4 5 4 3 1 5 5 3 1 4 1 4 1 3 3 4 5 1 3 1 4 3 5 3 5 1 5 4 4 5 3 4
##
## Within cluster sum of squares by cluster:
## [1] 1177.8849  238.7477  849.1396  653.1159  659.1016
##  (between_SS / total_SS =  41.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
perc.var.5 = round(100*(1 - kmeans5.diabetes$betweenss/kmeans5.diabetes$totss),1)
names(perc.var.5) <- "Perc. 5 clus"
perc.var.5
```

```
## Perc. 5 clus
##         58.3
```

```
# Calculating the percentage of variation for 6 clusters
(kmeans6.diabetes = kmeans(matstd.diabetes,6,nstart = 10))
```

```
## K-means clustering with 6 clusters of sizes 149, 100, 180, 36, 80, 223
##
## Cluster means:
##      Glucose BloodPressure SkinThickness    Insulin        BMI
## 1  0.4818962     0.3385159    -0.2473546 -0.3734779  0.2784496
## 2  1.1101834     0.2838699     1.0025905  1.6511100  0.7048844
## 3 -0.3821005     0.1392105     0.8314544  0.1762007  0.5039894
## 4 -0.1218083    -3.5358279    -1.1619988 -0.6864135 -0.7900257
## 5  0.3625905     0.5357323    -0.8383761 -0.5468786 -0.3635553
## 6 -0.6218158    -0.0872311    -0.4670985 -0.3260885 -0.6509873
##   DiabetesPedigreeFunction        Age    Outcome
## 1            -0.0009433043  0.2825517  1.3650064
## 2             0.8938368205  0.2201585  1.0295424
## 3            -0.1437266425 -0.3232927 -0.6384590
## 4            -0.2375576289 -0.2377879  0.2002009
## 5            -0.1188813845  1.7736993 -0.4433541
## 6            -0.2031827294 -0.6244796 -0.7316434
##
## Clustering vector:
##   [1] 1 6 1 6 2 6 3 4 2 5 6 1 5 2 1 4 2 1 3 1 3 5 1 1 1 1 1 6 5 5 5 2 6 6 5
##  [36] 3 5 1 3 2 3 5 5 2 5 2 6 6 1 4 6 6 6 2 2 6 2 3 5 3 4 1 6 3 1 6 1 5 6 3
##  [71] 1 3 1 3 3 6 6 3 4 6 6 4 3 6 1 3 3 3 1 6 6 3 3 5 6 3 6 6 6 2 1 6 6 6 6
## [106] 3 6 3 3 1 1 2 3 6 2 5 1 6 6 6 2 3 3 5 1 3 3 3 1 5 1 1 2 3 6 3 6 6 6 3
## [141] 5 3 6 1 3 6 3 3 5 6 3 6 2 2 1 1 6 6 6 2 3 3 3 6 1 1 3 6 6 6 1 1 4 3 6
## [176] 2 6 2 5 1 6 3 6 6 5 1 2 2 1 1 6 3 1 4 6 2 6 1 2 2 6 6 6 6 3 6 2 1 3 1
## [211] 6 3 5 1 1 2 3 3 1 1 2 5 4 5 6 3 6 1 2 3 1 2 6 6 6 1 2 1 1 6 6 3 1 2 3
## [246] 1 5 2 3 6 6 6 6 3 1 1 3 6 2 2 5 4 3 5 1 3 4 3 6 4 1 6 5 3 5 3 1 6 5 6
## [281] 1 3 3 1 5 5 2 2 6 3 3 1 2 2 5 3 2 3 1 5 4 1 3 1 5 3 1 6 2 2 6 3 1 6 1
## [316] 3 6 1 3 5 6 1 1 1 3 6 2 5 3 3 5 6 4 5 6 3 4 1 2 1 6 6 6 6 5 3 6 4 6 3
## [351] 6 6 3 6 6 1 2 4 3 2 2 5 5 5 3 3 1 6 6 1 2 6 3 3 3 2 6 3 1 3 3 6 6 6 6
## [386] 6 1 1 2 3 3 1 6 6 1 2 3 1 6 1 1 5 1 6 1 3 1 6 1 2 3 3 2 6 2 2 6 1 6 1
## [421] 3 6 3 6 2 2 4 2 3 1 4 6 6 6 6 4 3 6 6 6 1 6 3 1 1 2 6 3 1 6 6 1 3 4 3
## [456] 1 5 6 2 5 5 6 3 6 6 6 6 3 4 3 3 3 3 5 6 5 2 6 3 5 2 3 6 3 4 2 2 2 6 5
## [491] 3 3 3 1 4 5 6 6 1 2 6 3 3 3 3 6 1 6 6 5 1 6 5 6 6 1 2 5 6 5 6 3 4 1 6
## [526] 6 6 6 3 6 6 6 3 4 3 4 5 5 3 2 2 1 1 3 3 2 2 3 5 5 6 3 5 6 3 3 3 5 3 6
## [561] 1 2 3 6 6 6 3 3 3 1 6 6 6 6 2 3 6 1 6 2 1 6 5 5 2 6 1 6 2 4 1 3 1 6 3
## [596] 2 5 4 1 6 6 4 3 1 4 3 2 6 3 6 6 2 2 3 1 6 6 6 1 4 3 6 5 3 6 3 6 6 5 6
## [631] 1 3 6 6 6 1 5 6 1 6 6 6 1 4 3 2 1 2 1 6 6 3 3 6 3 2 6 2 5 1 5 2 2 2 1
## [666] 3 5 1 3 5 5 6 3 3 5 1 5 6 1 6 6 1 3 1 5 3 6 6 6 2 6 1 3 2 6 2 1 4 3 6
## [701] 3 1 1 4 6 3 4 3 1 3 2 3 1 6 6 2 2 5 3 1 6 3 1 3 5 3 3 3 6 6 1 1 2 3 5
## [736] 3 3 6 3 1 2 6 6 1 2 3 1 3 2 1 1 3 6 2 1 2 3 1 6 5 6 1 6 3 3 6 1 6
##
## Within cluster sum of squares by cluster:
## [1] 639.1144 775.4007 593.6084 238.7477 402.1888 650.7680
##  (between_SS / total_SS =  46.2 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
perc.var.6 = round(100*(1 - kmeans6.diabetes$betweenss/kmeans6.diabetes$totss),1)
names(perc.var.6) <- "Perc. 6 clus"
perc.var.6
```

```
## Perc. 6 clus
##         53.8
```

```
# Calculating the percentage of variation for 6 clusters
(kmeans7.diabetes = kmeans(matstd.diabetes,7,nstart = 10))
```

```
## K-means clustering with 7 clusters of sizes 36, 181, 82, 132, 213, 74, 50
##
## Cluster means:
##      Glucose BloodPressure SkinThickness     Insulin        BMI
## 1 -0.1218083   -3.53582785    -1.1619988 -0.68641346 -0.7900257
## 2 -0.4471903    0.13241708     0.8200945  0.11281824  0.4663826
## 3  1.2230925    0.30705574     0.9869493  1.93316686  0.7399965
## 4  0.5781277    0.33858687    -0.2758297 -0.36244121  0.2918007
## 5 -0.5850862   -0.08379401    -0.5165855 -0.32489909 -0.6503318
## 6  0.3118398    0.53911840    -0.8307732 -0.55032019 -0.3992795
## 7  0.2053466    0.22807270     0.4076889  0.07081081  0.2579123
##   DiabetesPedigreeFunction        Age    Outcome
## 1               -0.2375576 -0.2377879  0.2002009
## 2               -0.1921714 -0.3324249 -0.6273901
## 3                0.3473872  0.2418728  0.9303351
## 4               -0.2427708  0.2739092  1.3491227
## 5               -0.2627868 -0.6292827 -0.7218000
## 6               -0.2935702  1.8467812 -0.4199793
## 7                2.4918577  0.2023017  0.7360114
##
## Clustering vector:
##   [1] 4 5 4 5 7 5 2 1 3 6 5 4 7 3 4 1 3 4 2 4 2 6 4 4 4 4 4 5 6 6 6 3 5 5 6
##  [36] 2 5 4 2 7 2 6 6 3 6 7 5 5 4 1 5 5 5 3 3 5 3 2 7 2 1 4 5 2 4 5 7 6 5 2
##  [71] 7 2 4 2 2 5 5 2 1 5 5 1 2 5 4 2 2 2 4 5 5 2 2 6 5 2 5 5 5 3 7 5 5 5 5
## [106] 2 5 2 2 4 4 3 2 5 3 6 4 5 5 5 3 2 2 6 4 2 2 2 4 6 4 7 3 2 5 2 5 5 5 2
## [141] 6 2 2 4 2 5 2 7 6 5 2 5 7 3 4 4 5 5 5 3 2 2 2 5 4 4 2 5 5 5 4 4 1 2 5
## [176] 3 5 3 6 4 5 2 5 5 6 4 3 7 4 4 5 2 4 1 5 3 5 5 7 3 5 5 5 5 2 5 3 4 2 4
## [211] 5 2 6 4 4 3 2 2 7 4 3 6 1 6 5 2 5 4 3 2 4 3 5 5 5 4 3 4 4 5 5 2 4 7 2
## [246] 7 6 3 3 5 5 5 5 2 7 4 2 5 3 7 4 1 2 6 4 7 1 7 5 1 7 5 6 2 6 2 4 5 6 5
## [281] 4 2 2 4 6 6 3 3 5 2 2 4 7 3 6 2 3 2 4 6 1 4 2 4 6 2 4 5 7 7 5 2 4 5 7
## [316] 2 5 4 2 6 5 4 4 4 2 5 4 6 2 2 7 5 1 6 5 3 1 4 3 4 5 5 2 5 6 2 5 1 5 2
## [351] 5 5 2 5 5 4 7 1 2 3 3 6 6 6 3 2 4 5 5 4 7 7 2 2 2 3 5 2 4 2 2 5 5 7 5
## [386] 5 4 4 3 2 2 4 5 5 4 7 2 4 5 4 4 6 4 2 4 2 4 5 7 3 2 2 3 5 3 3 5 4 5 4
## [421] 2 5 2 5 3 3 1 3 2 4 1 5 5 5 5 1 2 5 5 5 4 5 2 4 4 7 5 2 2 5 5 4 2 1 2
## [456] 4 6 2 3 6 6 5 2 2 5 5 5 2 1 2 2 2 2 6 5 6 3 5 2 6 3 2 5 2 1 3 3 3 5 6
## [491] 2 2 2 7 1 6 5 5 4 3 5 2 2 2 2 5 4 5 5 6 4 5 6 5 5 4 3 6 5 6 5 2 1 4 5
## [526] 5 5 5 2 5 5 5 2 1 7 1 6 6 2 7 3 4 7 2 2 3 3 2 6 6 5 2 6 5 2 2 2 6 2 5
## [561] 4 3 2 5 5 5 2 2 2 4 5 5 2 5 3 2 5 4 5 3 4 5 6 6 3 5 4 5 7 1 7 2 4 7 2
## [596] 3 6 1 4 5 5 1 2 4 1 2 3 5 3 5 5 3 3 2 4 5 5 5 7 1 2 7 7 2 5 2 5 5 6 5
## [631] 4 2 5 5 5 4 6 5 7 5 5 5 4 1 2 3 4 3 4 5 5 2 2 5 2 3 5 7 6 7 6 7 3 3 4
## [666] 2 6 4 2 6 6 5 2 2 6 4 6 5 4 5 5 4 2 4 6 2 5 5 5 3 5 4 2 3 5 3 4 1 2 5
## [701] 2 4 4 1 5 2 1 2 4 2 3 2 4 5 5 3 3 6 2 4 5 2 4 2 6 2 2 2 5 5 4 4 3 2 6
## [736] 2 2 2 2 4 3 5 5 4 7 2 4 2 3 4 7 2 5 3 4 7 2 4 5 6 5 4 5 2 2 5 4 2
##
## Within cluster sum of squares by cluster:
## [1] 238.7477 573.1655 540.8002 504.8831 559.1398 330.4007 345.0511
##  (between_SS / total_SS =  49.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
perc.var.7 = round(100*(1 - kmeans6.diabetes$betweenss/kmeans6.diabetes$totss),1)
names(perc.var.7) <- "Perc. 7 clus"
perc.var.7
```

```
## Perc. 7 clus
##         53.8
```

```r
k.max <- 15 # Maximal number of clusters
wss <- sapply(1:k.max,function(k){kmeans(matstd.diabetes, k, nstart=50 )$tot.withinss})

plot(1:k.max, wss,
     type="b", pch = 19, frame = FALSE,       xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")

abline(v = 3, lty =2)

(kmeans8.diabetes = kmeans(matstd.diabetes,8,nstart = 10))
```

```
## K-means clustering with 8 clusters of sizes 176, 123, 70, 207, 45, 41, 36, 70
##
## Cluster means:
##      Glucose BloodPressure SkinThickness     Insulin        BMI
## 1 -0.4208924    0.13780146     0.7998259  0.11143804  0.4259160
## 2  0.5899287    0.33057760     0.8057679  0.24272741  0.5174653
## 3  0.2088317    0.51709634    -0.7276660 -0.49769724 -0.3715958
## 4 -0.6018386   -0.08806156    -0.5211963 -0.34840977 -0.6567094
## 5  1.3509797    0.17480129     0.8147378  2.84228307  0.5717060
## 6  0.3221691    0.21380838     0.2981544  0.06756002  0.1986208
## 7 -0.1218083   -3.53582785    -1.1619988 -0.68641346 -0.7900257
## 8  0.5980040    0.39679283    -1.2587163 -0.69243932  0.2558829
##   DiabetesPedigreeFunction        Age    Outcome
## 1               -0.1667911 -0.3461183 -0.6959051
## 2                0.1139589  0.2173310  1.3650064
## 3               -0.3247247  1.7190357 -0.6717391
## 4               -0.2773549 -0.6325516 -0.7215147
## 5                0.2614099  0.1117895  0.5263465
## 6                2.7586696  0.2138744  0.5979394
## 7               -0.2375576 -0.2377879  0.2002009
## 8               -0.2976476  0.5650253  1.3650064
##
## Clustering vector:
##   [1] 2 4 8 4 6 4 1 7 5 3 4 8 6 5 2 7 2 8 1 2 1 3 8 2 2 2 8 4 3 3 3 2 4 4 3
##  [36] 1 3 2 2 6 1 3 3 2 3 6 4 4 2 7 4 4 4 5 5 4 5 1 6 1 7 8 4 1 8 4 2 3 4 1
##  [71] 2 1 8 1 1 4 4 1 7 4 4 7 1 4 8 1 1 1 2 4 4 1 1 8 4 1 4 4 4 2 6 4 4 4 4
## [106] 1 4 1 1 2 2 5 1 4 2 8 8 4 4 4 2 1 1 3 8 2 1 1 2 8 2 6 2 1 4 1 1 4 4 1
## [141] 3 1 1 8 5 4 1 6 3 4 1 4 6 5 8 2 4 4 4 2 1 1 1 4 8 2 1 4 4 4 8 2 7 1 4
## [176] 2 3 2 3 8 4 1 4 4 3 2 5 6 2 2 4 1 8 7 4 2 4 4 2 5 4 4 4 4 1 4 5 8 1 2
## [211] 4 1 3 2 2 5 2 1 6 8 5 8 7 3 4 1 4 2 5 1 8 5 4 4 4 8 2 2 2 4 4 1 8 6 1
## [246] 6 3 5 5 4 4 4 4 1 6 2 1 4 5 6 3 7 1 3 8 6 7 6 4 7 6 4 3 1 3 1 2 4 3 4
## [281] 8 1 1 8 8 3 5 2 4 1 1 2 6 2 3 1 5 1 2 3 7 2 1 8 3 1 2 4 6 2 4 1 2 4 6
## [316] 1 4 8 1 8 4 2 8 2 1 4 2 3 2 1 6 4 7 3 4 5 7 8 2 8 4 4 1 4 3 1 4 7 4 1
## [351] 4 4 1 4 4 8 2 7 1 5 5 3 3 8 5 1 8 4 4 2 6 6 1 1 1 5 4 1 8 1 1 4 4 6 4
## [386] 4 2 2 2 1 1 8 5 4 8 6 1 2 4 2 8 3 2 1 8 1 8 4 6 5 1 1 5 4 2 5 4 2 4 2
## [421] 1 4 1 4 2 5 7 2 1 2 7 4 4 4 4 7 1 4 4 4 2 4 1 8 8 6 4 1 2 4 4 8 1 7 1
## [456] 2 3 1 2 3 3 4 1 1 4 4 4 1 7 1 1 1 1 3 4 3 2 4 1 3 5 1 4 1 7 2 5 5 4 3
## [491] 1 1 1 6 7 3 4 4 2 1 4 1 1 1 1 4 2 1 4 3 2 4 3 4 4 2 2 3 4 3 4 1 7 8 4
## [526] 4 4 4 1 4 4 4 1 7 6 7 3 3 1 2 2 2 2 1 1 2 2 1 3 3 4 1 3 4 1 1 1 3 1 4
## [561] 8 5 1 4 4 4 1 1 1 2 4 4 1 4 5 1 4 8 4 2 2 4 3 3 5 4 8 4 6 7 2 1 8 6 1
## [596] 2 3 7 8 4 4 7 1 2 7 1 5 4 5 4 4 2 5 1 2 4 4 4 6 7 1 6 6 1 4 1 4 4 3 4
## [631] 8 1 4 4 4 8 3 4 2 4 4 4 8 7 1 5 2 2 2 4 4 1 1 4 1 5 4 6 3 6 3 6 2 2 2
## [666] 1 3 2 1 3 3 4 3 1 3 8 8 4 8 4 4 2 1 8 3 1 4 4 1 2 4 8 1 2 4 5 2 7 1 4
## [701] 1 2 2 7 4 1 7 1 8 2 5 1 2 4 4 5 2 3 1 2 4 1 2 1 3 1 1 1 4 4 2 8 2 1 3
## [736] 1 1 1 1 8 2 4 4 8 6 1 2 1 2 8 6 1 4 5 2 6 1 8 4 8 4 2 4 3 1 4 8 1
##
## Within cluster sum of squares by cluster:
## [1] 522.0881 507.7452 303.2555 529.4166 293.7715 297.4958 238.7477 216.6794
##  (between_SS / total_SS =  52.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
plot(diabetes$Age, diabetes$SkinThickness,col=(kmeans8.diabetes$cluster+1), main="K-Means Clustering Results
with K=8",  pch=20,cex=2)

plot(diabetes$Age, diabetes$Insulin,col=(kmeans8.diabetes$cluster+1), main="K-Means Clustering Results with
K=8",  pch=20,cex=2)



feat.scaled <- scale(diabetes[,c("Age","Insulin")])
set.seed(15555)
pclusters <- kmeans(feat.scaled, 4, nstart=20, iter.max=100)

groups <- pclusters$cluster
#clusterDF <- cbind(as.data.frame(feat.scaled), Cluster=as.factor(groups)
plot(diabetes$Age, diabetes$Insulin, col=groups)


pclusters <- kmeans(feat.scaled, 9, nstart=20, iter.max=100)
groups <- pclusters$cluster
#clusterDF <- cbind(as.data.frame(feat.scaled), Cluster=as.factor(groups)
plot(diabetes$Age, diabetes$BMI, col=groups)
```