# Clustering

2024-12-21

## Contents



Data mining = Process of gathering insight and detecting pattern from large data set.

Partition = determine the number of group first then splits the set

Cluster = build group based on similarity

# K-means Clustering

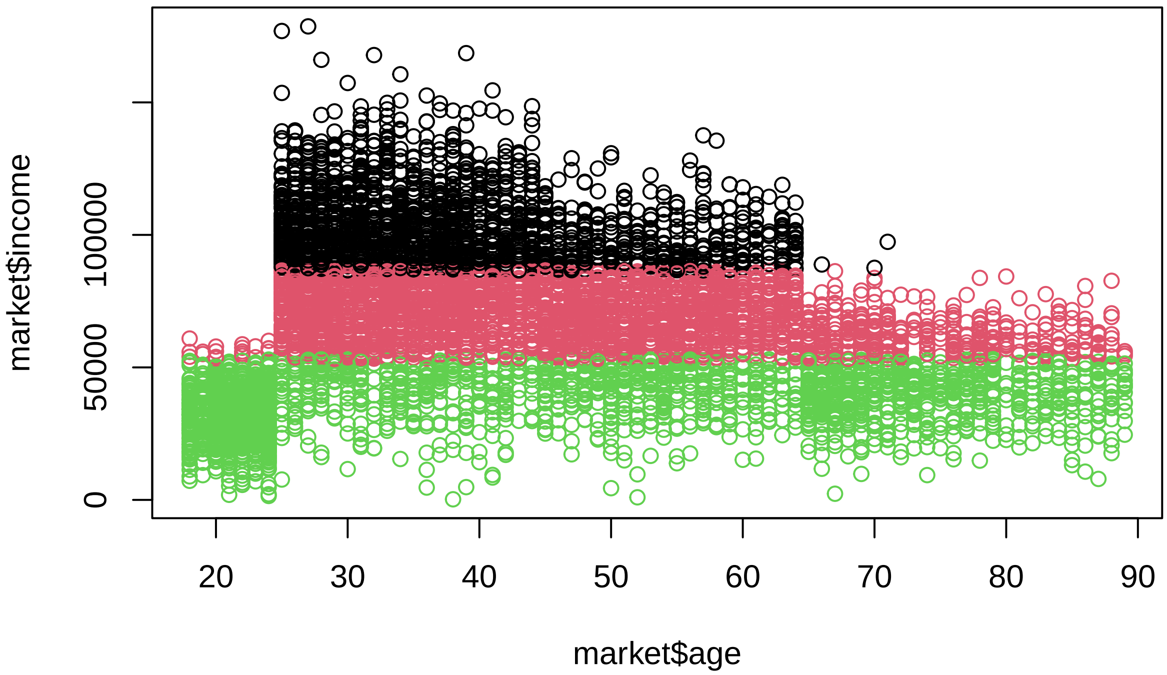
```
stations = read.csv('./Data/Ch5_bike_station_locations.csv')
two = kmeans(stations, 2)
two
```

```
## K-means clustering with 2 clusters of sizes 118, 126
##
## Cluster means:
##   latitude longitude
## 1 38.88838 -76.97846
## 2 38.93855 -77.03975
##
## Clustering vector:
##   [1] 2 1 2 1 2 2 1 1 1 1 2 1 1 1 2 2 2 2 2 1 1 2 2 1 2 1 2 2 2 1 1 1 1 2 2 2 2
##  [38] 1 2 2 1 2 2 2 1 2 1 2 1 2 1 2 1 1 1 1 1 1 2 1 2 2 2 2 1 1 1 2 1 2 1 2 2 2
##  [75] 1 2 1 2 1 2 1 2 2 2 1 2 1 2 1 1 2 1 2 1 1 2 2 1 2 2 1 1 2 2 1 2 2 2 2 2 2
## [112] 2 1 2 2 1 2 2 1 1 1 1 2 2 2 1 1 1 1 1 1 2 2 2 2 1 1 1 2 1 2 2 2 2 2 2 2 1
## [149] 2 1 2 2 1 2 2 1 2 1 1 1 1 1 1 1 2 1 2 2 2 1 2 2 2 1 2 1 1 2 2 2 1 1 1 2 1
## [186] 1 1 2 1 2 1 1 2 2 1 1 1 2 1 2 2 2 2 2 2 1 1 2 2 2 2 1 1 2 2 1 2 1 1 2 1 2
## [223] 1 1 2 1 1 2 1 1 2 1 2 2 1 1 1 2 1 2 1 1 1 1
##
## Within cluster sum of squares by cluster:
## [1] 0.1754263 0.1575802
##  (between_SS / total_SS =  53.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
market = read.csv('./Data/Ch5_age_income_data.csv')
str(market)
```

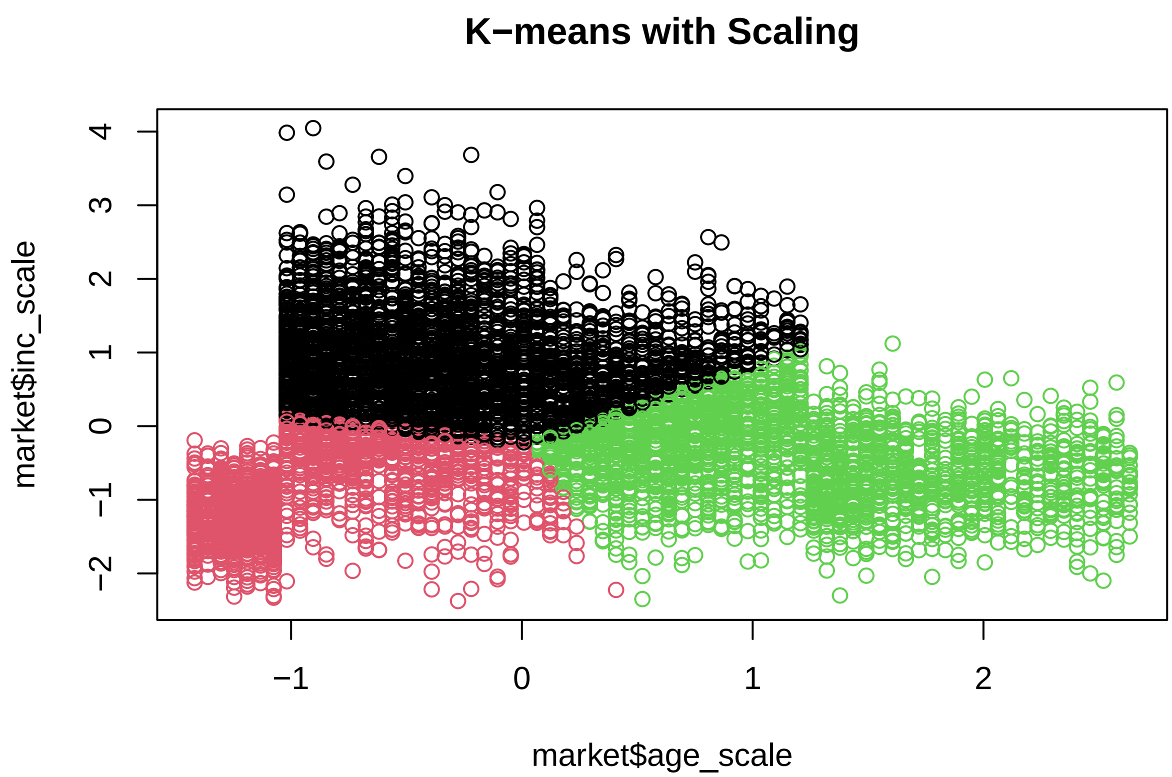
```
## 'data.frame':    8105 obs. of  3 variables:
##  $ bin   : chr  "60-69" "30-39" "20-29" "30-39" ...
##  $ age   : int  64 33 24 33 78 62 88 54 54 31 ...
##  $ income: num  87083 76808 12044 61972 60120 ...
```

```r
three = kmeans(market[,c(2,3)], 3)
plot(market$age, market$income, col=three$cluster)
```




```r
market$age_scale = as.numeric(scale(market$age))
market$inc_scale = as.numeric(scale(market$income))
```

```r
three_scale = kmeans(market[, c(4,5)],3)
plot(market$age_scale, market$inc_scale, col=three_scale$cluster,
     main='K-means with Scaling')
```




## Hierachical Clustering

### ETL

```
market = read.csv('./data/Ch5_age_income_data.csv')
```

### EDA

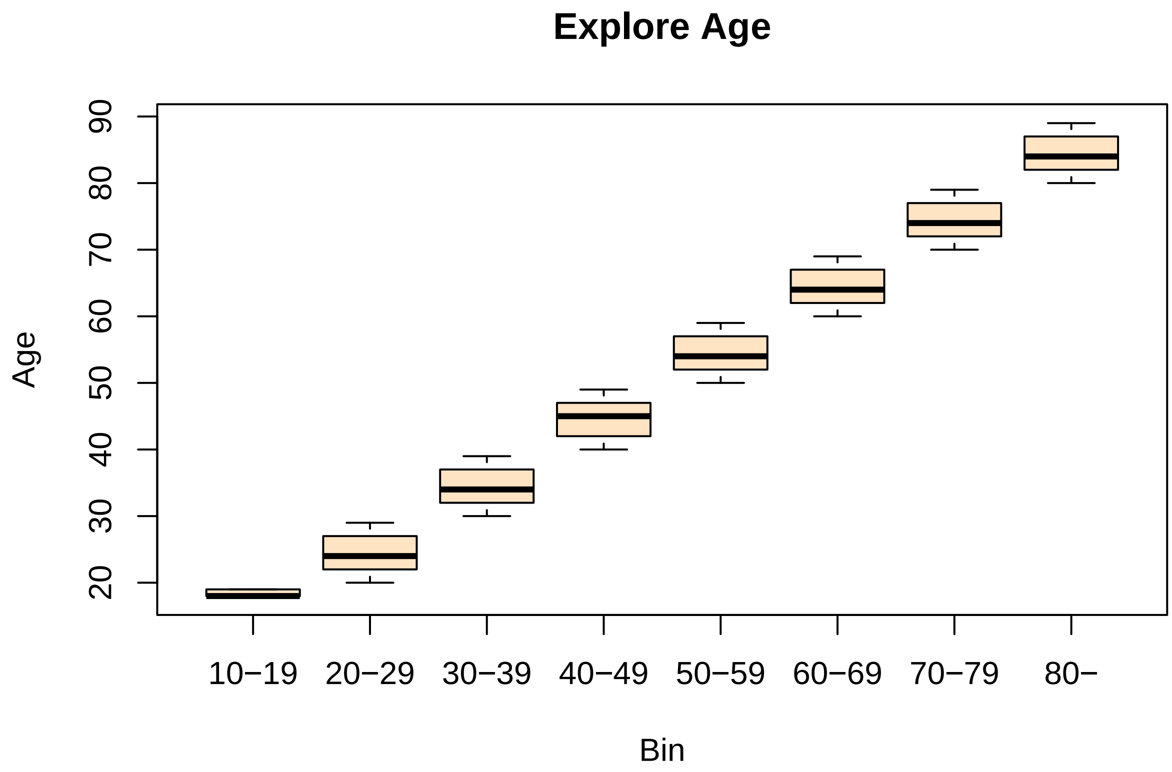
```
str(market)
```

```
## 'data.frame':    8105 obs. of  3 variables:
##  $ bin   : chr  "60-69" "30-39" "20-29" "30-39" ...
##  $ age   : int  64 33 24 33 78 62 88 54 54 31 ...
##  $ income: num  87083 76808 12044 61972 60120 ...
```

```
summary(market)
```

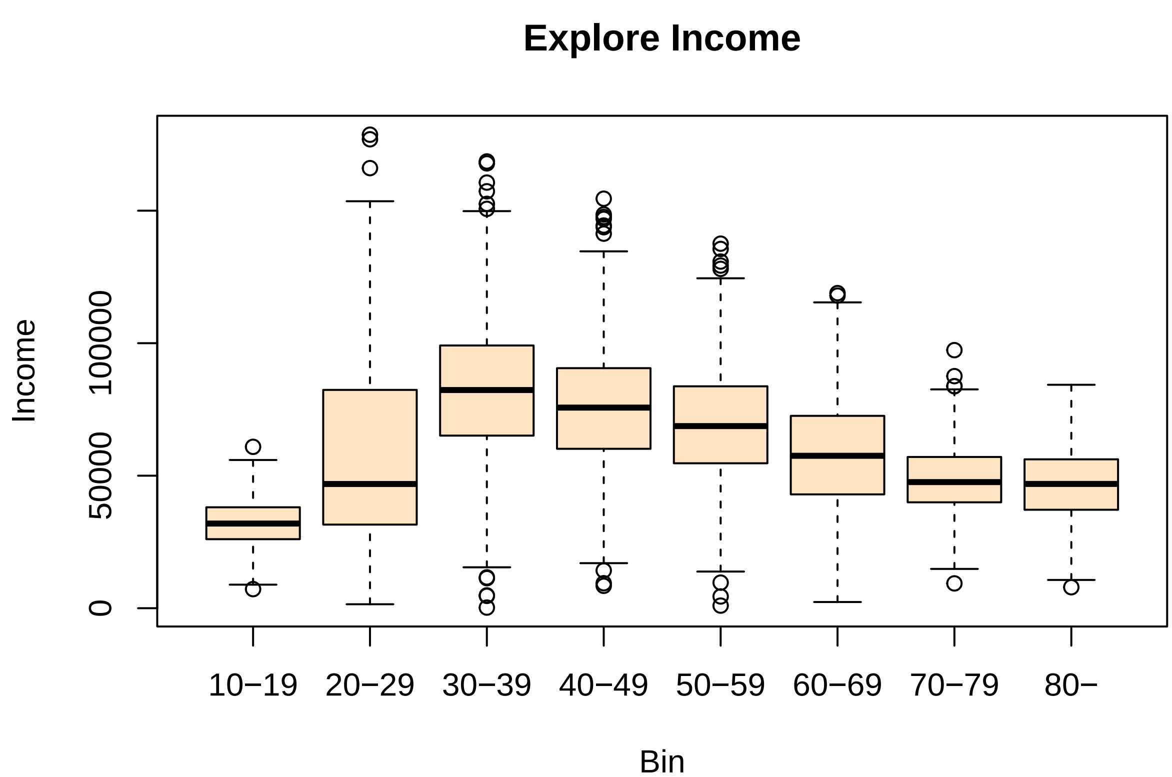
```
##      bin                 age            income
##  Length:8105        Min.   :18.00   Min.   :   233.6
##  Class :character   1st Qu.:28.00   1st Qu.: 43792.7
##  Mode  :character   Median :39.00   Median : 65060.0
##                     Mean   :42.85   Mean   : 66223.6
##                     3rd Qu.:55.00   3rd Qu.: 85944.7
##                     Max.   :89.00   Max.   :178676.4
```

```
boxplot(market$age~market$bin, main = 'Explore Age', col = 'bisque',
        xlab = 'Bin', ylab = 'Age')
```




This box plot shows no improperly binned ages

```
boxplot(market$income~market$bin, main = 'Explore Income', col = 'bisque',
        xlab = 'Bin', ylab = 'Income')
```




This box plot shows that there is a non-linear relationship between age and income. However, is there a correlation between age and age and income. Let us check:

```r
cor.test(market$age, market$income)
```

```
##
##  Pearson's product-moment correlation
##
## data:  market$age and market$income
## t = -5.4055, df = 8103, p-value = 6.648e-08
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.08160633 -0.03822020
## sample estimates:
##         cor
## -0.05994158
```

The output shows that there is a correlation of -0.05994158 and p-value of 6.648e-08. This indicates that there is a significant mild correlation between age and income.

We use Ward D algorithm

```r
set.seed(456)
market$age_scale = as.numeric(scale(market$age))
market$inc_scale = as.numeric(scale(market$income))
hc_mod = hclust(dist(market[, 4:5]), method='ward.D2')
# dist() : distance matrix
hc_mod
```

```
##
## Call:
## hclust(d = dist(market[, 4:5]), method = "ward.D2")
##
## Cluster method   : ward.D2
## Distance         : euclidean
## Number of objects: 8105
```

```r
# convert hierarchical clustering into dendrogram for visualization
# dendrogram is a tree-like hierarchical clustering structure
dend = as.dendrogram(hc_mod)

# library for branches color
library(dendextend)
```

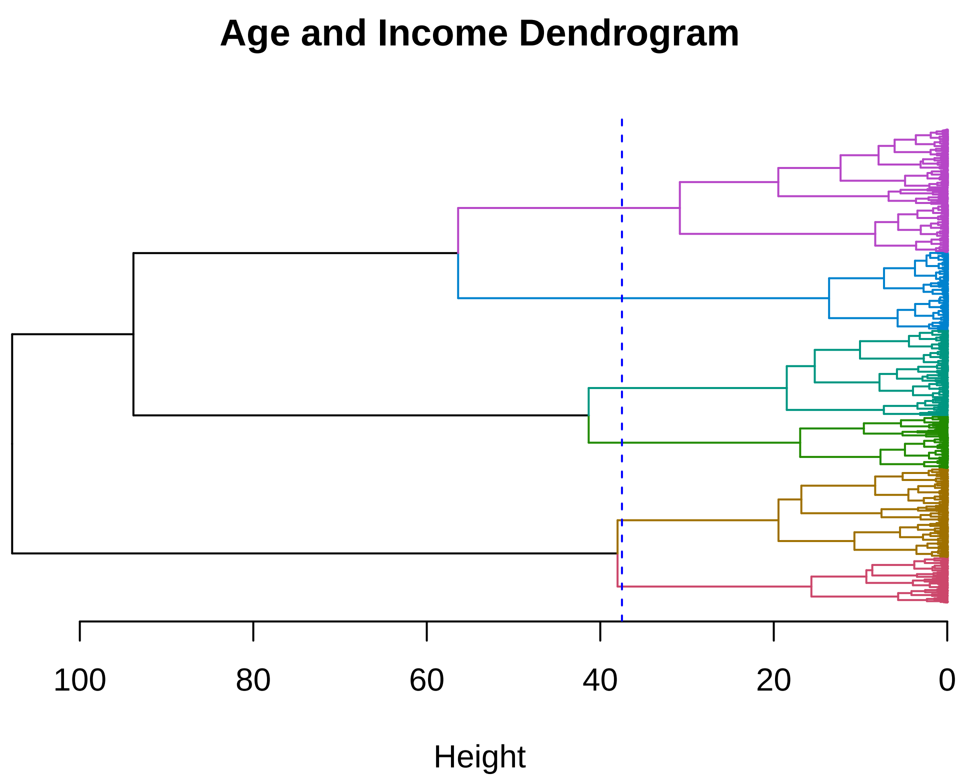
```
##
## ---------------------
## Welcome to dendextend version 1.19.0
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

```
##
## Attaching package: 'dendextend'
```

```
## The following object is masked from 'package:stats':
##
##     cutree
```

```r
# set the number of cluster
dend_six_color = color_branches(dend, k=6)
# k = 6: 6 color for 6 cluster

# plot the dendrogram
plot(dend_six_color, leaflab='none', horiz=T, main='Age and Income Dendrogram',
     xlab='Height');abline(v=37.5, lty='dashed',col='blue')
```




```r
# leaflab = 'none' : suppress numerical labes at the end of the dendrogram
# horiz = T : change the layout of the visualization
```

Height is the indicator of strength of the separation between branches

```r
str(cut(dend, h=37.5)$upper)
```

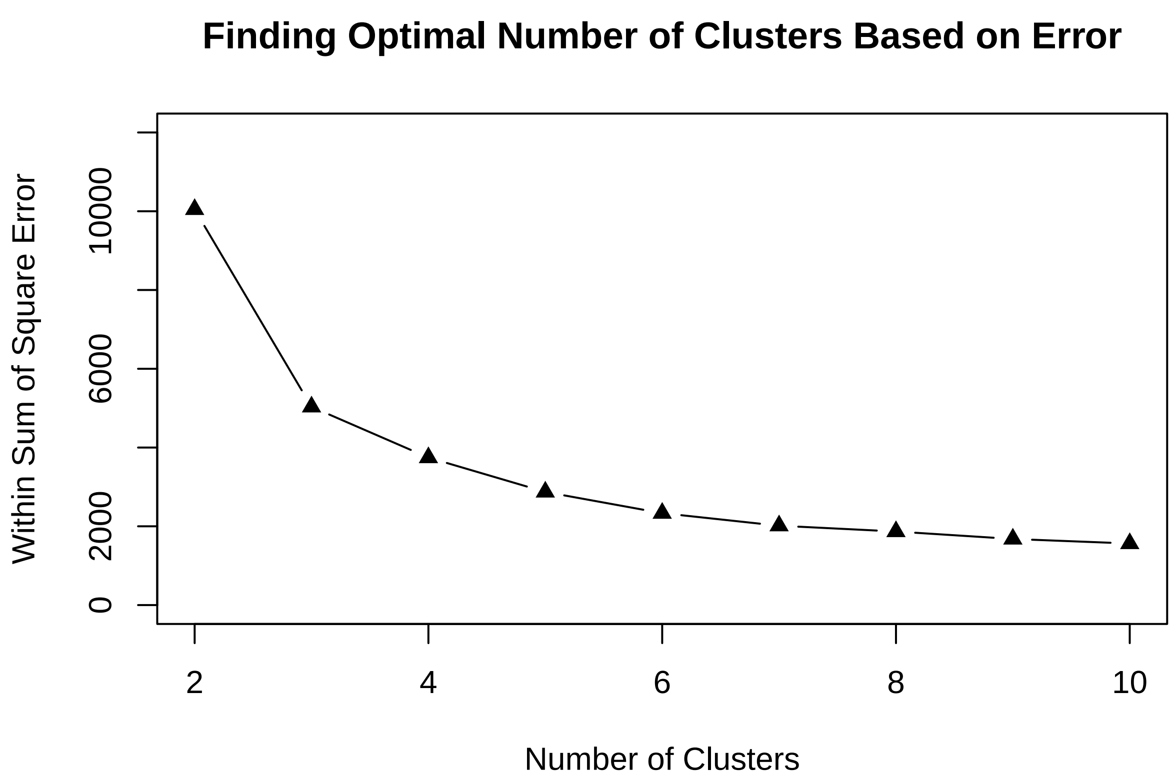
```
## --[dendrogram w/ 2 branches and 6 members at h = 108]
##   |--[dendrogram w/ 2 branches and 2 members at h = 38]
##   |  |--leaf "Branch 1" (h= 15.7 midpoint = 274, x.member = 782 )
##   |  `--leaf "Branch 2" (h= 19.5 midpoint = 628, x.member = 1526 )
##   `--[dendrogram w/ 2 branches and 4 members at h = 93.8]
##      |--[dendrogram w/ 2 branches and 2 members at h = 41.3]
##      |  |--leaf "Branch 3" (h= 17 midpoint = 431, x.member = 905 )
##      |  `--leaf "Branch 4" (h= 18.5 midpoint = 463, x.member = 1473 )
##      `--[dendrogram w/ 2 branches and 2 members at h = 56.4]
##         |--leaf "Branch 5" (h= 13.6 midpoint = 530, x.member = 1323 )
##         `--leaf "Branch 6" (h= 30.8 midpoint = 753, x.member = 2096 )
```

Interpretation :

-

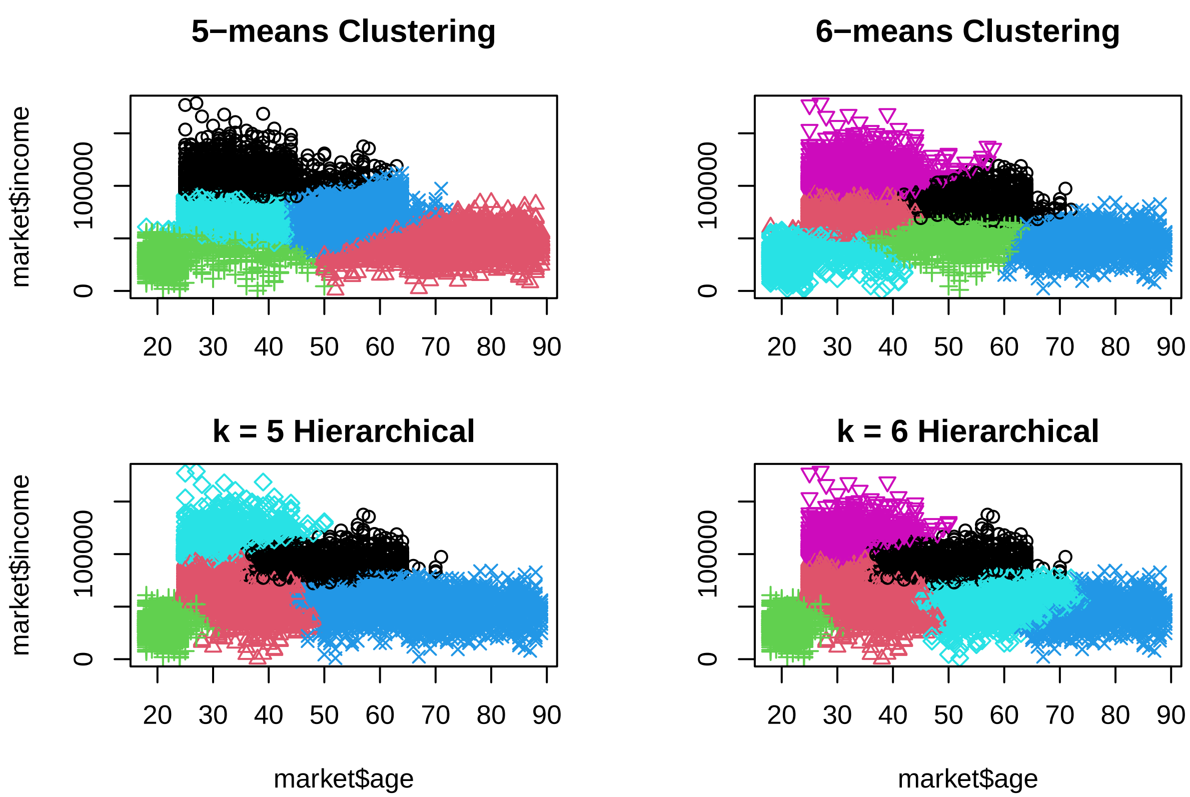
```
one = kmeans(market[,c(4,5)], 1)
two = kmeans(market[,c(4,5)], 2)
three = kmeans(market[,c(4,5)], 3)
four = kmeans(market[,c(4,5)], 4)
five = kmeans(market[,c(4,5)], 5)
six = kmeans(market[,c(4,5)], 6)
seven = kmeans(market[,c(4,5)], 7)
eight = kmeans(market[,c(4,5)], 8)
nine = kmeans(market[,c(4,5)], 9)
ten = kmeans(market[,c(4,5)], 10)
```

```
optimize <- data.frame(clusters = c(2:10), wss = rep(0, 9))
 optimize[1, 2] <- as.numeric(two$tot.withinss)
 optimize[2, 2] <- as.numeric(three$tot.withinss)
 optimize[3, 2] <- as.numeric(four$tot.withinss)
 optimize[4, 2] <- as.numeric(five$tot.withinss)
 optimize[5, 2] <- as.numeric(six$tot.withinss)
 optimize[6, 2] <- as.numeric(seven$tot.withinss)
 optimize[7, 2] <- as.numeric(eight$tot.withinss)
 optimize[8, 2] <- as.numeric(nine$tot.withinss)
 optimize[9, 2] <- as.numeric(ten$tot.withinss)
 plot(optimize$wss ~ optimize$clusters, type = "b",
     ylim = c(0, 12000), ylab = 'Within Sum of Square Error',
     main = 'Finding Optimal Number of Clusters Based on Error',
     xlab = 'Number of Clusters', pch = 17, col = 'black')
```




```
market$clus5 <- five$cluster
dend_five <- cutree(dend, k = 5)
market$dend5 <- dend_five
market$clus6 <- six$cluster
dend_six <- cutree(dend, k = 6)
market$dend6 <- dend_six
```

```r
par(mfrow = c(2, 2), mar = c(3, 4, 4, 2) + 0.1)
plot(market$age, market$income, col = five$cluster,
     pch = five$cluster, xlab = '', main = '5-means Clustering')
plot(market$age, market$income, col = six$cluster, xlab = '',
     ylab = '', pch = six$cluster, main = '6-means Clustering')
par(mar = c(5, 4, 2, 2) + 0.1)
plot(market$age, market$income, col = market$dend5,
     pch = market$dend5, main = 'k = 5 Hierarchical')
plot(market$age, market$income, col = market$dend6, ylab = '',
     pch = market$dend6, main = 'k = 6 Hierarchical')
```




```r
par(mfrow = c(1, 1), mar = c(5, 4, 4, 2) + 0.1)
```

```r
library(dplyr)
```

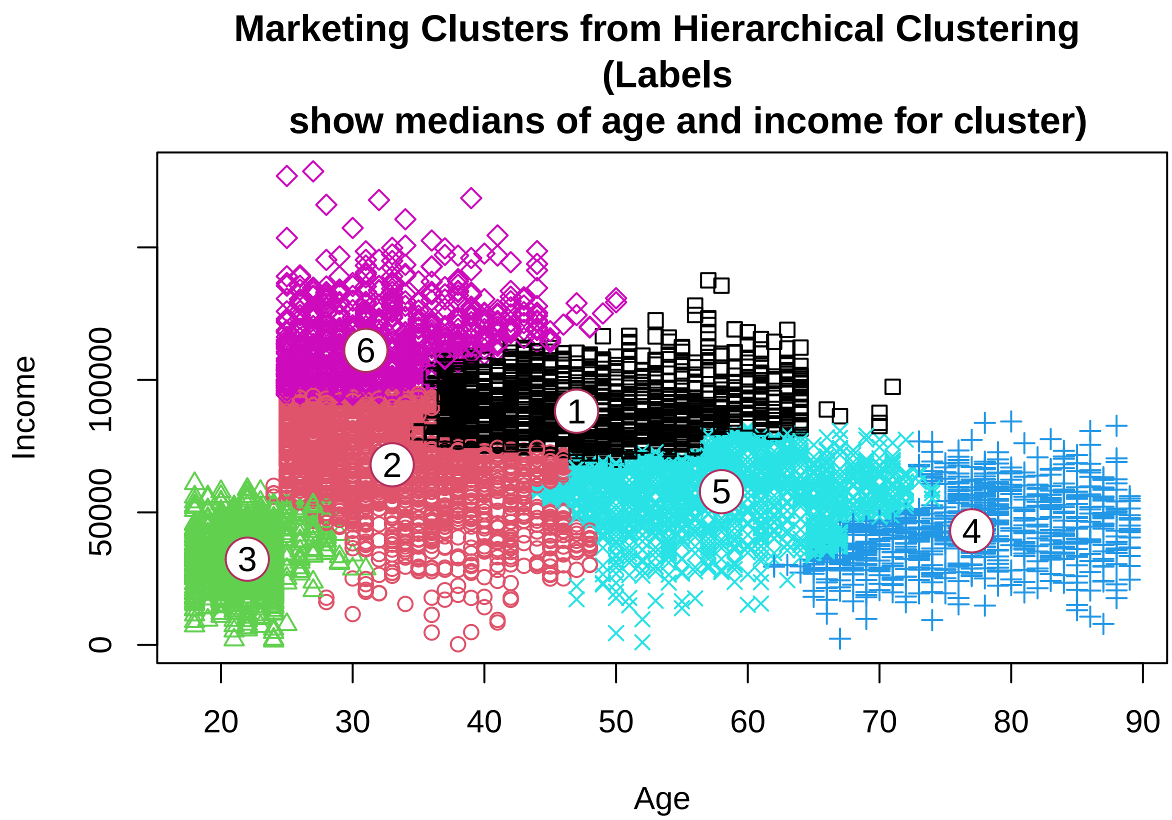
```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
labels <- as.data.frame(market %>%
                        group_by(dend6) %>%
                        summarise(avg_age = median(age),
                                  avg_inc = median(income)))
```

```r
plot(market$age, market$income, col = market$dend6,
     pch = market$dend6 - 1, xlab = "Age", ylab = "Income",
     main = 'Marketing Clusters from Hierarchical Clustering \n (Labels
     show medians of age and income for cluster)')
points(labels[ ,2], labels[ ,3], pch = 21, col = 'maroon',
       bg = 'white', cex = 3)
text(labels[ ,2], labels[ ,3], cex = 1.1, col = 'black',
     labels[ ,1])
```




```r
market %>% group_by(dend6) %>% summarise(ClusterSize=n())
```

```
## # A tibble: 6 x 2
##   dend6 ClusterSize
##   <int>       <int>
## 1     1        1473
## 2     2        2096
## 3     3        1323
## 4     4         782
## 5     5        1526
## 6     6         905
```

```r
data=market %>% group_by(dend6) %>% summarise(min_age=min(age),
                                              med_age=median(age),
                                              max_age=max(age),
                                              min_inc=min(income),
                                              med_inc=median(income),
                                              max_inc=max(income))
```

```r
label = c('old and rich','mid career with mid income', 'young and broke', 'pension', 'old and broke', 'Old Money')
data$labels = label
data
```

```
## # A tibble: 6 x 8
```

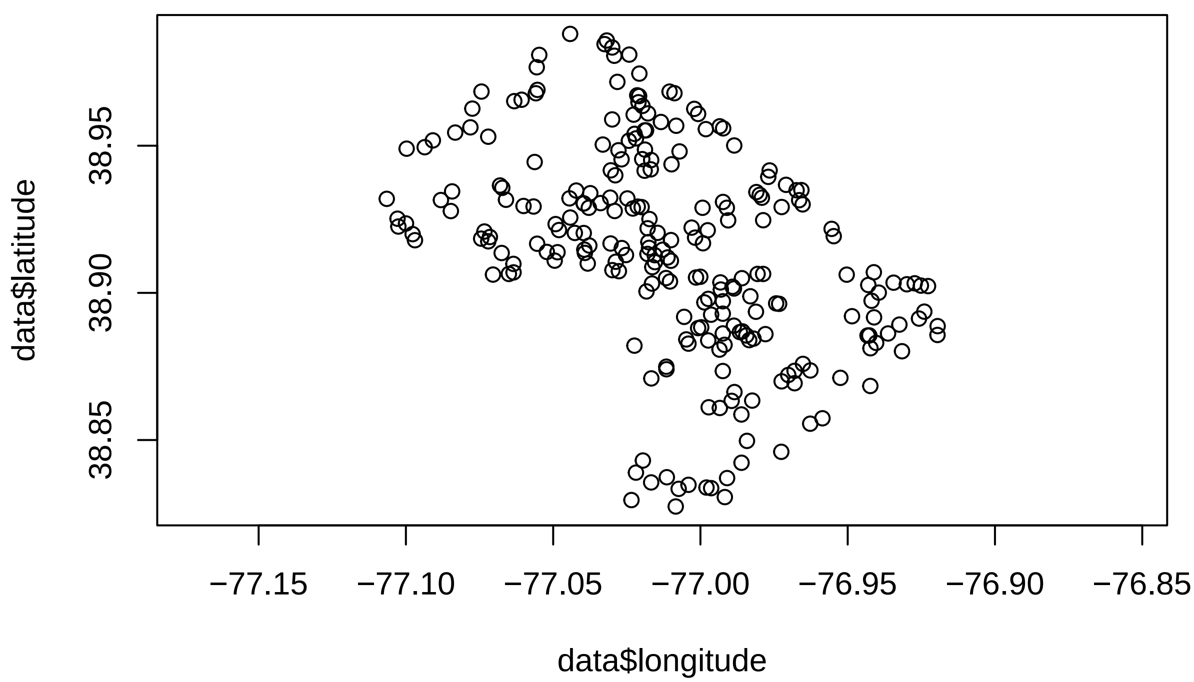
```
##   dend6 min_age med_age max_age min_inc med_inc max_inc labels
##   <int>   <int>   <dbl>   <int>   <dbl>   <dbl>   <dbl> <chr>
## 1     1      35      47      71  69492.  88170. 137557. old and rich
## 2     2      24      33      48    234.  67958.  94709. mid career with mid inc~
## 3     3      18      22      31   1485.  32329.  60887. young and broke
## 4     4      62      77      89   2319.  43044.  84301. pension
## 5     5      44      58      74    973.  57806.  81988. old and broke
## 6     6      25      31      50  93827. 111125. 178676. Old Money
```

## Self exercise

```
data = read.csv('./Data/Ch5_bike_station_locations.csv')
summary(data)
```

```
##     latitude       longitude
##  Min.   :38.83   Min.   :-77.11
##  1st Qu.:38.89   1st Qu.:-77.03
##  Median :38.92   Median :-77.01
##  Mean   :38.91   Mean   :-77.01
##  3rd Qu.:38.94   3rd Qu.:-76.99
##  Max.   :38.99   Max.   :-76.92
```

```
plot(data$longitude, data$latitude, asp = 1)
```




```
set.seed(123)
km = kmeans(data,3)
km
```

```
## K-means clustering with 3 clusters of sizes 48, 69, 127
##
## Cluster means:
```

```
##   latitude longitude
## 1 38.90753 -76.95526
## 2 38.87463 -76.99426
## 3 38.93839 -77.03945
##
## Clustering vector:
##   [1] 3 2 3 1 3 3 1 2 1 1 3 1 1 1 3 3 3 3 3 1 2 3 3 1 3 1 3 3 3 2 1 2 3 3 3 3 3
##  [38] 1 3 3 2 3 3 3 1 3 1 3 2 3 2 3 2 2 1 1 1 2 3 1 3 3 3 3 3 2 2 2 3 1 3 1 3 3 3
##  [75] 1 3 2 3 2 3 1 3 3 3 1 3 2 3 2 2 3 1 3 2 2 3 3 2 3 3 3 2 2 3 3 1 3 3 3 3 3 3
## [112] 3 2 3 3 2 3 3 2 1 1 1 3 3 3 2 2 2 2 2 2 2 3 3 3 3 2 1 2 3 1 3 3 3 3 3 3 3 2
## [149] 3 2 3 3 2 3 3 2 3 1 1 1 1 2 2 2 3 1 3 3 3 2 3 3 3 1 3 2 2 3 3 3 2 2 2 3 2
## [186] 2 2 3 2 3 2 2 3 3 2 1 2 3 1 3 3 3 3 3 3 1 2 3 3 3 3 2 2 3 3 1 3 1 2 3 2 3
## [223] 1 2 3 2 2 3 2 1 3 1 3 3 1 2 1 3 2 3 2 1 2 1
##
## Within cluster sum of squares by cluster:
## [1] 0.04361512 0.05663749 0.15939642
##  (between_SS / total_SS =  63.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
clus = cbind(data, km$cluster)
plot(clus$longitude, clus$latitude, col = km$cluster)
```