ABF\_Week1\_HW

taehwan.lee

2023-12-27

#install.packages("Quandl")  
#install.packages("highcharter")  
#install.packages("plotly")  
#install.packages("bigmemory")  
#install.packages("biganalytics")  
#install.packages("ff")  
#install.packages("biglm")  
library(Quandl)

## Warning: 패키지 'Quandl'는 R 버전 4.3.2에서 작성되었습니다

## 필요한 패키지를 로딩중입니다: xts

## 필요한 패키지를 로딩중입니다: zoo

##   
## 다음의 패키지를 부착합니다: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(tidyverse)

## Warning: 패키지 'tidyverse'는 R 버전 4.3.2에서 작성되었습니다

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.3 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::first() masks xts::first()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ dplyr::last() masks xts::last()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(highcharter)

## Warning: 패키지 'highcharter'는 R 버전 4.3.2에서 작성되었습니다

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo   
## Highcharts (www.highcharts.com) is a Highsoft software product which is  
## not free for commercial and Governmental use

library(quantmod)

## 필요한 패키지를 로딩중입니다: TTR

library(plotly)

## Warning: 패키지 'plotly'는 R 버전 4.3.2에서 작성되었습니다

##   
## 다음의 패키지를 부착합니다: 'plotly'  
##   
## The following object is masked from 'package:ggplot2':  
##   
## last\_plot  
##   
## The following object is masked from 'package:stats':  
##   
## filter  
##   
## The following object is masked from 'package:graphics':  
##   
## layout

library(bigmemory)

## Warning: 패키지 'bigmemory'는 R 버전 4.3.2에서 작성되었습니다

library(biganalytics)

## Warning: 패키지 'biganalytics'는 R 버전 4.3.2에서 작성되었습니다

## 필요한 패키지를 로딩중입니다: foreach

## Warning: 패키지 'foreach'는 R 버전 4.3.2에서 작성되었습니다

##   
## 다음의 패키지를 부착합니다: 'foreach'  
##   
## The following objects are masked from 'package:purrr':  
##   
## accumulate, when  
##   
## 필요한 패키지를 로딩중입니다: biglm

## Warning: 패키지 'biglm'는 R 버전 4.3.2에서 작성되었습니다

## 필요한 패키지를 로딩중입니다: DBI

library(ff)

## Warning: 패키지 'ff'는 R 버전 4.3.2에서 작성되었습니다

## 필요한 패키지를 로딩중입니다: bit  
##   
## 다음의 패키지를 부착합니다: 'bit'  
##   
## The following object is masked from 'package:dplyr':  
##   
## symdiff  
##   
## The following object is masked from 'package:base':  
##   
## xor  
##   
## Attaching package ff  
## - getOption("fftempdir")=="C:/Users/luisf/AppData/Local/Temp/Rtmpu0vhbs/ff"  
##   
## - getOption("ffextension")=="ff"  
##   
## - getOption("ffdrop")==TRUE  
##   
## - getOption("fffinonexit")==TRUE  
##   
## - getOption("ffpagesize")==65536  
##   
## - getOption("ffcaching")=="mmnoflush" -- consider "ffeachflush" if your system stalls on large writes  
##   
## - getOption("ffbatchbytes")==16777216 -- consider a different value for tuning your system  
##   
## - getOption("ffmaxbytes")==536870912 -- consider a different value for tuning your system  
##   
##   
## 다음의 패키지를 부착합니다: 'ff'  
##   
## The following object is masked from 'package:bigmemory':  
##   
## is.readonly  
##   
## The following objects are masked from 'package:utils':  
##   
## write.csv, write.csv2  
##   
## The following objects are masked from 'package:base':  
##   
## is.factor, is.ordered

library(biglm)

Quandl.api\_key("mFJ-d-Lj4n2fvywxg\_DN")

Quandl( code, type = c(“raw”, “ts”, “zoo”, “xts”, “timeSeries”), transform = c(““,”diff”, “rdiff”, “normalize”, “cumul”, “rdiff\_from”), collapse = c(““,”daily”, “weekly”, “monthly”, “quarterly”, “annual”), order = c(“desc”, “asc”), meta = FALSE, force\_irregular = FALSE, … )

해당 함수는 R 프로그래밍 언어에서 사용되는 함수로, 주로 금융 데이터를 가져오는 데 사용됩니다. 아래는 함수의 각 매개변수에 대한 설명입니다:

1. code: 데이터를 가져올 금융 기관 또는 데이터셋의 코드 또는 식별자를 지정하는 매개변수입니다. Quandl에서 특정 시계열 데이터를 고유하게 식별하는 데 사용될 수 있는 코드일 것입니다.
2. type: 데이터의 출력 구조 유형을 지정하는 매개변수입니다. 옵션으로는 다음이 있습니다: “raw”: 원시 데이터. “ts”: 시계열 객체. “zoo”: Zoo 객체. “xts”: xts 객체. “timeSeries”: TimeSeries 객체.
3. transform: 데이터에 적용할 변환을 지정하는 매개변수로, 다음과 같은 옵션이 있습니다: ““: 변환 없음.”diff”: 차분. “rdiff”: 상대 차분. “normalize”: 정규화. “cumul”: 누적. “rdiff\_from”: 지정된 지점에서의 상대 차분.
4. collapse: 데이터를 축소할 빈도를 지정하는 매개변수로, 다음과 같은 옵션이 있습니다: ““: 축소하지 않음.”daily”: 일일 데이터. “weekly”: 주간 데이터. “monthly”: 월간 데이터. “quarterly”: 분기별 데이터. “annual”: 연간 데이터.
5. order: 가져온 데이터의 정렬 순서를 지정하는 매개변수로, 다음과 같은 옵션이 있습니다: “desc”: 내림차순. “asc”: 오름차순.
6. meta: 출력에 메타데이터를 포함할지 여부를 나타내는 논리형 매개변수입니다.
7. force\_irregular: 불규칙한 시계열을 강제로 생성할지 여부를 나타내는 논리형 매개변수입니다.
8. Slice and Dice the Data ex) mydata = Quandl(“FRED/GDP”, start\_date=“2001-12-31”, end\_date=“2005-12-31”)
9. Request specific columns ex) mydata = Quandl(c(“FRED/GDP.1”, “WIKI/AAPL.4”))

Qdata2 <- Quandl("NSE/OIL", collapse = "monthly")  
head(Qdata2)

## Date Open High Low Last Close Total Trade Quantity  
## 1 2019-01-31 172.05 174.95 172.05 174.55 174.55 431122  
## 2 2018-12-31 178.10 179.00 174.35 175.00 174.80 761462  
## 3 2018-11-30 190.00 195.80 185.35 186.75 186.25 6828327  
## 4 2018-10-31 196.15 198.60 194.00 196.95 197.60 426894  
## 5 2018-09-30 216.75 224.95 210.60 221.20 219.55 1139867  
## 6 2018-08-31 212.80 216.75 212.80 213.65 213.60 760196  
## Turnover (Lacs)  
## 1 749.99  
## 2 1343.75  
## 3 12935.68  
## 4 837.69  
## 5 2457.46  
## 6 1632.53

Qdata3 <- Quandl("FRED/GDP", start\_date = as.Date("2000-01-01"), end\_date = as.Date("2020-08-31"), type = "xts")  
head(Qdata3)

## [,1]  
## 2000 Q1 10002.18  
## 2000 Q2 10247.72  
## 2000 Q3 10318.17  
## 2000 Q4 10435.74  
## 2001 Q1 10470.23  
## 2001 Q2 10599.00

Qdata4 = Quandl("FRED/GDP", transform = "rdiff")  
head(Qdata4)

## Date Value  
## 1 2021-10-01 0.03404876  
## 2 2021-07-01 0.02028872  
## 3 2021-04-01 0.03188700  
## 4 2021-01-01 0.02610297  
## 5 2020-10-01 0.01603812  
## 6 2020-07-01 0.08528480

#Crude Oil Prices: West Texas Intermediate  
oil\_daily <- Quandl("FRED/DCOILWTICO", type = "raw", collapse = "daily", start\_date = "2008-01-01", end\_date = "2024-01-01")  
head(oil\_daily)

## Date Value  
## 1 2022-05-23 110.32  
## 2 2022-05-20 112.63  
## 3 2022-05-19 112.21  
## 4 2022-05-18 109.67  
## 5 2022-05-17 112.31  
## 6 2022-05-16 114.07

oil\_weekly <- Quandl("FRED/DCOILWTICO", type = "xts", collapse = "weekly", start\_date = "2008-01-01", end\_date = "2024-01-01")  
head(oil\_weekly)

## [,1]  
## 2008-01-06 97.90  
## 2008-01-13 92.74  
## 2008-01-20 90.55  
## 2008-01-27 90.37  
## 2008-02-03 89.03  
## 2008-02-10 91.77

oil\_monthly <- Quandl("FRED/DCOILWTICO", type = "xts", collapse = "monthly", start\_date = "2008-01-01", end\_date = "2024-01-01")  
head(oil\_monthly)

## [,1]  
## 1 2008 91.67  
## 2 2008 101.78  
## 3 2008 101.54  
## 4 2008 113.70  
## 5 2008 127.35  
## 6 2008 139.96

tail(index(oil\_monthly))

## [1] "12 2021" "1 2022" "2 2022" "3 2022" "4 2022" "5 2022"

mdy(last(index(oil\_monthly)))

## [1] "2022-05-20"

#mdy 함수는 날짜를 월-일-년(월/일/년) 형식으로 파싱하고 Date 객체로 반환하는 함수입니다  
#index(oil\_monthly):시계열 데이터에서 시간 정보를 나타내는 인덱스를 추출합니다. 여기서는 날짜 또는 기간에 해당하는 값들이 추출됩니다.  
#last(...): 추출된 값 중에서 마지막 값을 선택합니다. 따라서 이 부분은 시계열 데이터의 마지막 시간 인덱스를 나타냅니다.  
index(oil\_monthly) = seq(mdy("01/01/2008"), mdy(last(index(oil\_monthly))), by = "months")  
tail(index(oil\_monthly))

## [1] "2021-12-01" "2022-01-01" "2022-02-01" "2022-03-01" "2022-04-01"  
## [6] "2022-05-01"

head(index(oil\_monthly))

## [1] "2008-01-01" "2008-02-01" "2008-03-01" "2008-04-01" "2008-05-01"  
## [6] "2008-06-01"

hchart(oil\_monthly)

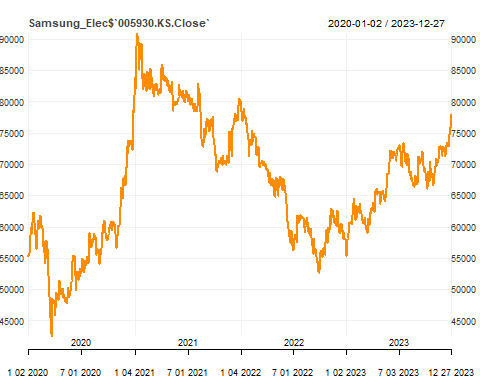


highchart(type = "stock") %>% hc\_add\_series(oil\_monthly, color = "cornflowerblue") %>%  
 hc\_yAxis(  
 title = list(text = "monthly price"),  
 labels = list(format = "${value}"), #y 축값에 $를 붙이라는 의미  
 opposite = FALSE #y title과 label이 왼쪽에 위치한다. //TRUE = 오른쪽에 위치  
 ) %>% hc\_add\_theme(hc\_theme\_flat())



Apple <- getSymbols("AAPL", from = "2020-01-01", to = "2023-12-31", auto.assign = F)  
SP500 <- getSymbols("^GSPC", from = "2020-01-01", to = "2023-12-31", auto.assign = F) #SP500 index  
Hanwha\_Life <- getSymbols("088350.KS", from = "2020-01-01", to = "2023-12-31", auto.assign = F)   
Samsung\_Life <- getSymbols("032830.KS", from = "2020-01-01", to = "2023-12-31", auto.assign = F)   
Samsung\_Elec <- getSymbols("005930.KS", from = "2020-01-01", to = "2023-12-31", auto.assign = F)   
#auto.assign = FALSE로 설정되어 있다면, 데이터를 받아온 후에는 현재 R 작업 환경에 자동으로 할당되지 않는다.  
#getSymbols("005930.KS", from = "2020-01-01", to = "2023-12-31", auto.assign = T) 로 할 경우, 데이터를 받아와서  
#005930.KS에 저장하게 된다.   
#getSymbols("005930.KS", from = "2020-01-01", to = "2023-12-31", auto.assign = T)

chart\_Series(Samsung\_Elec$`005930.KS.Close`) #tick mark: 숫자 1 옆. 특수문자로 시작할 경우 tick mark 붙여야 함

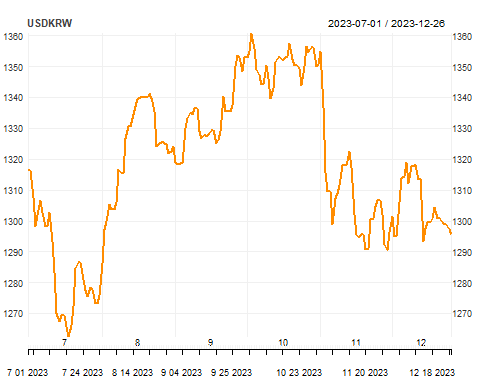


# knit시 오류로 주석처리함  
# chart\_Series(`005930.KS`$`005930.KS.Close`)

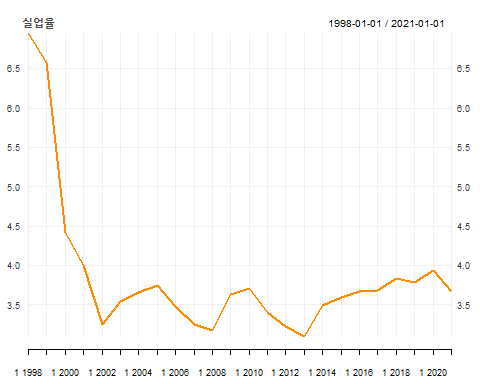
getSymbols("USD/KRW", src ="oanda", from = Sys.Date() - 180, to = Sys.Date())

## [1] "USD/KRW"

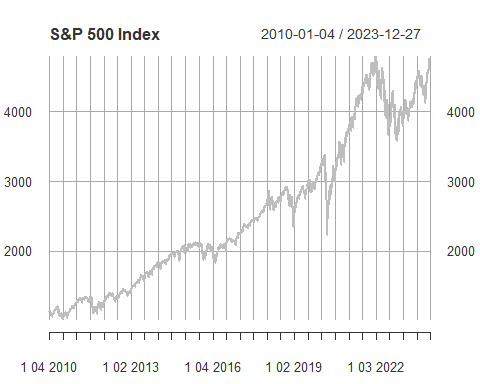
chart\_Series(USDKRW) #getSymbols의 auto.assign 기본값은 TRUE이므로, 위 명령을 실행하면 USDKRW 변수에 데이터를 저장한다.



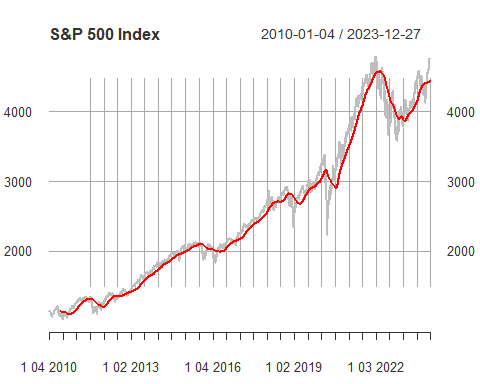
#우리나라 실업율  
실업율 <- getSymbols("LRHUTTTTKRA156N", src = "FRED", from = "1998-01-01", to = "2021-12-31", auto.assign = F)  
chart\_Series(실업율)



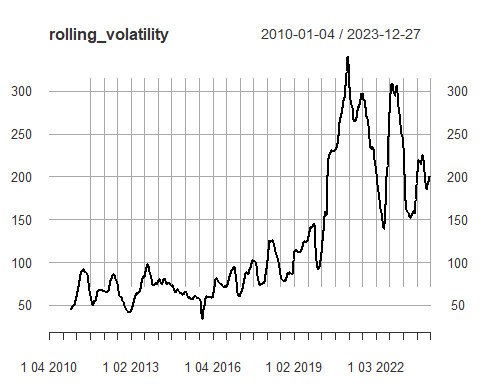
SP500 <- getSymbols("^GSPC", from = "2010-01-01", to = "2023-12-31", auto.assign = F)  
plot(SP500[,4], col = "gray", type = "l", main = "S&P 500 Index")



rolling\_average = rollapply(SP500[,4], 100, mean) # 최근 100개의 평균을 구한다.  
lines(rolling\_average, col = "red", lwd = 2)



rolling\_volatility = rollapply(SP500[,4], 200, sd)  
plot(rolling\_volatility)



head(SP500)

## GSPC.Open GSPC.High GSPC.Low GSPC.Close GSPC.Volume GSPC.Adjusted  
## 2010-01-04 1116.56 1133.87 1116.56 1132.99 3991400000 1132.99  
## 2010-01-05 1132.66 1136.63 1129.66 1136.52 2491020000 1136.52  
## 2010-01-06 1135.71 1139.19 1133.95 1137.14 4972660000 1137.14  
## 2010-01-07 1136.27 1142.46 1131.32 1141.69 5270680000 1141.69  
## 2010-01-08 1140.52 1145.39 1136.22 1144.98 4389590000 1144.98  
## 2010-01-11 1145.96 1149.74 1142.02 1146.98 4255780000 1146.98

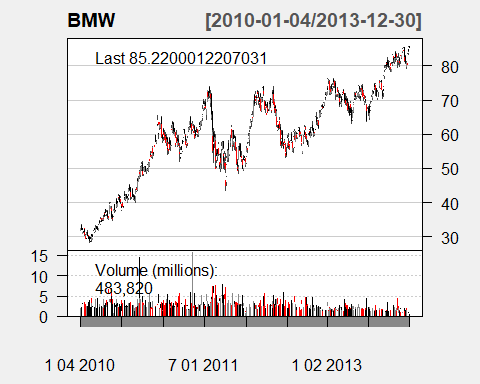
B <- new.env() # 새로운 환경이 생성되며, 이 환경은 변수, 함수, 그리고 다른 환경을 담을 수 있는 공간이 됩니다.  
getSymbols("BMW.DE", env = B, src = "yahoo", from = "2010-01-01", to = "2013-12-31")

## [1] "BMW.DE"

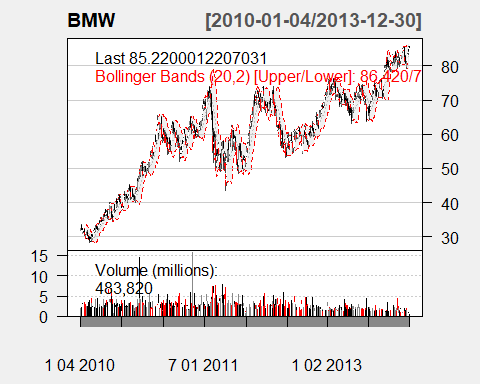
#2010년 1월 1일부터 2013년 12월 31일까지의 BMW 주식 데이터를 Yahoo Finance에서 가져와서 환경 B에 저장하는 작업을 수행합니다. 가져온 데이터는 B라는 환경 내에 "BMW.DE"라는 이름으로 변수로 저장됨  
BMW <- B$BMW.DE  
head(BMW)

## BMW.DE.Open BMW.DE.High BMW.DE.Low BMW.DE.Close BMW.DE.Volume  
## 2010-01-04 31.820 32.455 31.820 32.050 1808170  
## 2010-01-05 31.960 32.410 31.785 32.310 1564182  
## 2010-01-06 32.450 33.040 32.360 32.810 2218604  
## 2010-01-07 32.650 33.200 32.380 33.100 2026145  
## 2010-01-08 33.335 33.430 32.515 32.655 1925894  
## 2010-01-11 32.995 33.050 32.110 32.170 2157825  
## BMW.DE.Adjusted  
## 2010-01-04 18.19094  
## 2010-01-05 18.33851  
## 2010-01-06 18.62230  
## 2010-01-07 18.78690  
## 2010-01-08 18.53433  
## 2010-01-11 18.25905

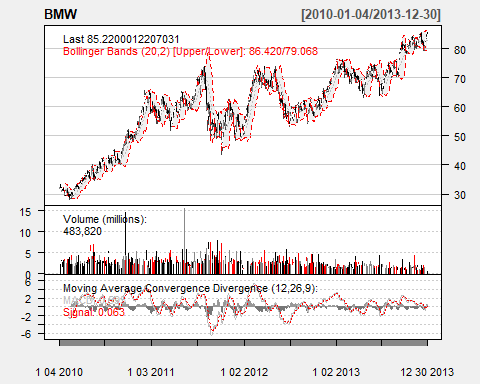
chartSeries(BMW, multi.col = TRUE, theme = "white")



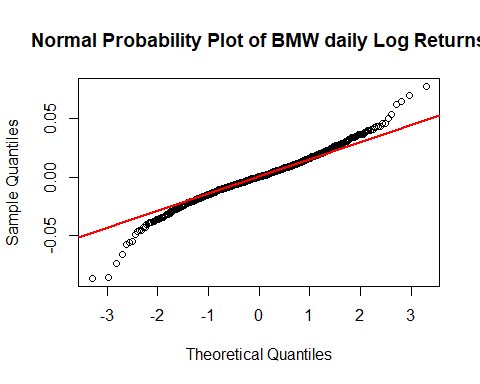
addBBands() #add the Bollinger Bands



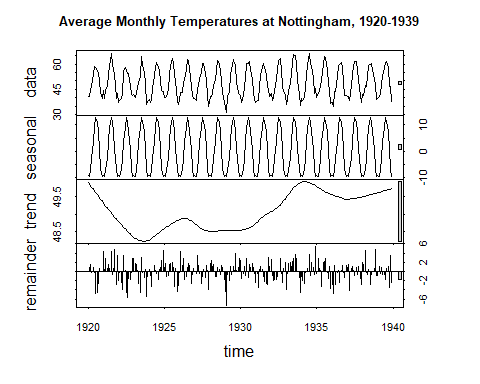
addMACD() #the MACD trend-follwing momentum indicator



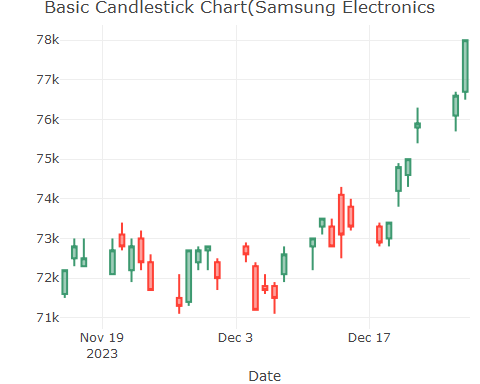
B.Return <- log(BMW$BMW.DE.Close / BMW$BMW.DE.Open)  
qqnorm(B.Return,  
 main = "Normal Probability Plot of BMW daily Log Returns",  
 xlab = "Theoretical Quantiles",  
 ylab = "Sample Quantiles")  
qqline(B.Return, col = "red", lwd = 2.5)



#Seasonal Decomposition of Times Series by Loess  
# stl 함수는 R의 시계열 데이터에 대해 STL (Seasonal-Trend decomposition using LOESS) 분해를 수행하는 함수입니다. STL은 시계열 데이터를 계절성, 추세, 그리고 나머지(residuals)로 분해하여 각 구성 요소를 개별적으로 추정합니다.  
#   
# 여기서 stl 함수의 몇 가지 주요 매개변수에 대한 설명  
#   
# - nottem: STL 분해를 수행할 시계열 데이터입니다. 여기서는 nottem이라는 변수가 해당 데이터를 나타내는 것으로 가정합니다.  
#   
# - s.window: 계절성 구성 요소를 추정할 때 사용되는 LOESS(smoothed time-series) 창의 크기입니다. "periodic"으로 설정하면 자동으로 계절성을 결정  
#   
# - t.window: 추세(trend) 구성 요소를 추정할 때 사용되는 LOESS 창의 크기  
#   
# - t.jump: 추세(trend)의 점프(jump) 또는 간격을 결정합니다. 이 매개변수는 추세를 추정할 때 사용되는 창을 얼마나 건너뛸지를 결정  
#   
# 이러한 설정을 사용하여 stl(nottem, s.window = "periodic", t.window = 70, t.jump = 1)는 nottem 시계열 데이터에 대한 STL 분해를 수행합니다. 계절성은 자동으로 결정되고, 추세는 LOESS 창 크기가 70이고 간격이 1인 설정으로 추정한다  
  
#t.jump 값을 크게 하면 trend 곡선이 변한다.  
plot(stl(nottem, s.window = "periodic", t.window = 70, t.jump = 1), main = "Average Monthly Temperatures at Nottingham, 1920-1939")



#Candl Stick Chart 그려보기  
df <- data.frame(Date = index(Samsung\_Elec), coredata(Samsung\_Elec))  
df <- tail(df, 30)  
  
fig <- df %>%  
 plot\_ly(  
 x = ~Date,  
 type = "candlestick",  
 open = ~X005930.KS.Open,  
 close = ~X005930.KS.Close,  
 high = ~X005930.KS.High,  
 low = ~X005930.KS.Low  
 )  
  
# 하단 range slider on/off설정 F:off, T:on  
fig <-  
 fig %>%  
 layout(title = "Basic Candlestick Chart(Samsung Electronics", xaxis = list(rangeslider = list(visible = F)))  
  
fig



# read.big.matrix는 R 프로그래밍 언어의 bigmemory 패키지에 속한 함수 중 하나입니다. bigmemory 패키지는 대용량 메모리 매핑을 통해 대규모 데이터를 효율적으로 다루는 데 사용되는 도구를 제공합니다.  
# read.big.matrix 함수는 대용량의 행렬 데이터를 효율적으로 메모리에 읽어오는 데 사용됩니다. 특히, 메모리에 맞지 않는 큰 데이터셋을 작은 조각으로 나누어 읽어들일 때 유용합니다.  
x <- read.big.matrix("bmw.csv", header = TRUE, type = "double")  
head(x)

## Index BMW.DE.Open BMW.DE.High BMW.DE.Low BMW.DE.Close BMW.DE.Volume  
## [1,] 1 31.820 32.455 31.820 32.050 1808170  
## [2,] 2 31.960 32.410 31.785 32.310 1564182  
## [3,] 3 32.450 33.040 32.360 32.810 2218604  
## [4,] 4 32.650 33.200 32.380 33.100 2026145  
## [5,] 5 33.335 33.430 32.515 32.655 1925894  
## [6,] 6 32.995 33.050 32.110 32.170 2157825  
## BMW.DE.Adjusted  
## [1,] 21.72463  
## [2,] 21.90087  
## [3,] 22.23979  
## [4,] 22.43635  
## [5,] 22.13472  
## [6,] 21.80597

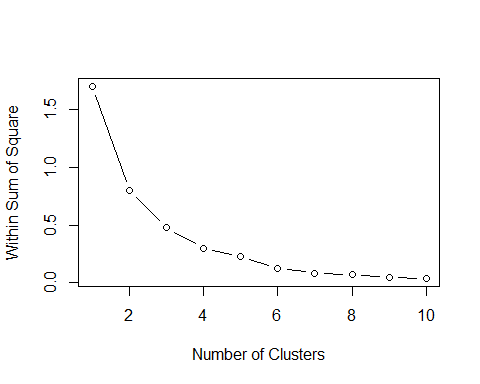
x <- x[,-1]  
head(x)

## BMW.DE.Open BMW.DE.High BMW.DE.Low BMW.DE.Close BMW.DE.Volume  
## [1,] 31.820 32.455 31.820 32.050 1808170  
## [2,] 31.960 32.410 31.785 32.310 1564182  
## [3,] 32.450 33.040 32.360 32.810 2218604  
## [4,] 32.650 33.200 32.380 33.100 2026145  
## [5,] 33.335 33.430 32.515 32.655 1925894  
## [6,] 32.995 33.050 32.110 32.170 2157825  
## BMW.DE.Adjusted  
## [1,] 21.72463  
## [2,] 21.90087  
## [3,] 22.23979  
## [4,] 22.43635  
## [5,] 22.13472  
## [6,] 21.80597

class(x)

## [1] "matrix" "array"

# lapply 함수는 R에서 리스트에 적용되는 함수로, 주어진 리스트의 각 요소에 함수를 적용하여 리스트를 반환합니다. 여기서 1:10은 1부터 10까지의 정수로 이루어진 리스트를 나타냅니다. 따라서 lapply(1:10, function(i){bigkmeans(x, center = i, iter.max = 100, nstart = 10)})은 1부터 10까지의 정수를 가지고 있는 리스트에 대해 bigkmeans 함수를 각각 적용하는 것을 의미합니다.  
  
# 이 코드에서 사용된 함수와 매개변수들에 대한 설명은 다음과 같습니다:  
# 1:10: 1부터 10까지의 정수로 이루어진 리스트를 생성합니다.  
# function(i){...}: 리스트의 각 요소에 대해 실행될 함수를 정의합니다. 함수의 인자 i는 리스트의 각 요소를 나타냅니다.  
# bigkmeans(x, center = i, iter.max = 100, nstart = 10): bigkmeans는 대용량 데이터셋에 대해 k-평균 군집화를 수행하는 함수입니다.  
# x: 군집화를 수행할 데이터셋.  
# center = i: 클러스터의 개수를 i로 설정합니다. i는 1부터 10까지의 정수로 변화하면서 군집화가 수행됩니다.  
# iter.max = 100: 반복 횟수의 상한값을 설정합니다.  
# nstart = 10: 무작위 초기화를 10번 시도하며 가장 좋은 결과를 선택합니다.  
# 따라서 lapply(1:10, function(i){bigkmeans(x, center = i, iter.max = 100, nstart = 10)})은 군집화를 클러스터 수가 1부터 10까지 변화하면서 각각 수행하고, 그 결과를 리스트로 반환합니다. 반환된 리스트의 각 요소는 각 클러스터 수에 대한 군집화 결과입니다.  
Results <- lapply(1:10, function(i){bigkmeans(x, center = i, iter.max = 100, nstart = 10)})  
Within <- lapply(Results, function(x) sum(x$withinss)/(10^15))   
#10^15 로 나눈 이유는 Within cluster sum of squares by cluster 값이 {값}\*e^15로 표기 되기 때문에 e^15를 없애기 위함이다.   
plot(1:10, Within, type = "b", xlab = "Number of Clusters", ylab = "Within Sum of Square")



Results

## [[1]]  
## K-means clustering with 1 clusters of sizes 1017  
##   
## Cluster means:  
## [1] 5.958961e+01 6.033826e+01 5.878460e+01 5.960954e+01 2.561593e+06  
## [6] 4.232860e+01  
##   
## Clustering vector:  
## [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [223] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [260] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [297] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [334] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [371] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [408] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [445] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [482] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [519] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [556] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [593] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [630] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [667] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [704] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [741] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [778] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [815] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [852] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [889] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [926] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [963] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [1000] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
##   
## Within cluster sum of squares by cluster:  
## [1] 1.702861e+15  
##   
## Available components:  
##   
## [1] "cluster" "centers" "withinss" "size"   
##   
## [[2]]  
## K-means clustering with 2 clusters of sizes 180, 837  
##   
## Cluster means:  
## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 54.47614 55.59542 53.30133 54.52558 4595234 38.04962  
## [2,] 60.68928 61.35823 59.96380 60.70286 2124251 43.24881  
##   
## Clustering vector:  
## [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 1 1 2 2 2 1 2 2 2 1 2 2 2  
## [38] 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2  
## [75] 1 2 2 2 2 2 1 2 2 2 1 1 1 1 1 1 1 2 1 1 2 1 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2  
## [112] 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 1 1 1 2 2 2 2 2 2 2 2 2 2  
## [149] 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 1 1 2  
## [186] 2 2 1 2 2 1 2 1 1 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 1 2 2 1 1 2 2 2 2 2  
## [223] 1 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1  
## [260] 2 2 2 2 2 2 1 2 2 1 1 1 2 2 2 2 1 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 1 2  
## [297] 2 1 1 2 2 2 2 2 1 2 2 1 1 1 1 2 2 2 1 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2  
## [334] 2 2 2 2 2 2 2 2 1 1 1 2 2 1 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2  
## [371] 2 2 2 1 2 2 1 2 1 2 2 1 2 2 2 2 2 2 2 2 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 1 1  
## [408] 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 2 2 2 1 1 2 1 1 1 2 1 1 1 1 1 1 2 2 2 1 1  
## [445] 1 1 2 2 1 1 1 1 1 1 2 2 2 2 1 1 2 2 1 2 2 2 1 2 2 1 2 1 2 2 2 1 2 2 2 2 1  
## [482] 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [519] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [556] 2 2 2 1 2 1 2 2 1 1 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2  
## [593] 2 1 2 2 2 1 2 2 2 2 1 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 1  
## [630] 2 2 2 2 2 2 1 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 1 1 1 2 2  
## [667] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 1 1 2 2 2 2 2 1 2 2 2 1 1 2 2 2 2  
## [704] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 1 1 2 2 2 2 2 2 2 2 2  
## [741] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [778] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [815] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [852] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 2 2  
## [889] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [926] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2  
## [963] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [1000] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
##   
## Within cluster sum of squares by cluster:  
## [1] 4.300350e+14 3.683095e+14  
##   
## Available components:  
##   
## [1] "cluster" "centers" "withinss" "size"   
##   
## [[3]]  
## K-means clustering with 3 clusters of sizes 542, 409, 66  
##   
## Cluster means:  
## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 63.19836 63.77132 62.53810 63.20363 1736850 45.47610  
## [2,] 55.72375 56.63454 54.83730 55.80077 3119202 38.97277  
## [3,] 53.91068 55.09750 52.42174 53.69727 5879001 37.27694  
##   
## Clustering vector:  
## [1] 1 1 1 1 1 1 2 1 2 2 1 2 2 2 2 1 2 2 3 3 1 1 1 2 2 2 1 1 2 2 1 2 1 2 2 1 1  
## [38] 1 1 1 1 2 1 1 2 1 1 1 2 1 1 1 2 2 2 2 2 1 2 2 1 1 2 2 2 1 1 1 1 1 2 1 2 1  
## [75] 2 2 2 2 1 2 2 2 1 1 2 2 3 3 3 2 2 2 2 2 2 3 3 3 1 2 2 2 1 1 2 1 2 2 1 2 1  
## [112] 2 2 2 1 2 1 3 2 1 1 2 2 2 1 2 2 2 1 1 2 2 1 1 3 3 3 2 1 2 1 1 1 1 2 1 1 1  
## [149] 1 3 2 2 2 1 1 1 2 1 1 2 1 2 1 1 1 2 1 1 1 2 2 1 2 1 1 2 2 2 1 2 2 2 3 2 2  
## [186] 2 2 2 1 2 2 2 3 2 2 2 2 2 1 2 2 1 2 1 1 1 1 1 1 1 2 2 2 2 2 3 2 2 1 1 1 1  
## [223] 2 1 1 1 1 2 2 2 2 1 1 1 2 2 2 2 2 2 2 3 2 1 1 1 1 2 1 1 1 1 2 2 1 1 1 1 2  
## [260] 1 2 1 2 1 2 2 1 2 2 3 3 2 2 2 2 2 2 1 2 2 1 2 3 2 2 2 1 1 2 2 1 1 1 2 2 2  
## [297] 1 2 2 1 2 1 1 2 3 2 2 3 2 2 3 2 2 2 2 1 1 1 1 1 2 1 1 2 2 2 2 2 2 1 2 1 2  
## [334] 2 1 1 2 2 1 1 2 3 2 2 2 2 3 3 2 1 1 1 1 3 2 1 1 2 1 1 1 1 1 1 1 1 2 2 2 1  
## [371] 2 2 2 3 1 2 2 2 3 1 2 2 2 2 1 1 1 2 1 2 3 2 2 2 1 1 1 2 1 2 1 2 2 1 2 3 2  
## [408] 3 3 3 3 3 3 2 2 2 2 3 3 3 3 2 3 2 1 2 2 2 2 2 2 2 2 2 2 2 3 3 3 2 2 1 3 3  
## [445] 3 2 2 2 3 3 3 3 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 1 1 1 2  
## [482] 2 2 2 2 2 2 2 2 2 3 2 2 2 1 1 2 2 1 2 3 2 3 2 1 1 1 1 1 1 1 1 1 2 1 2 1 2  
## [519] 2 1 1 2 1 2 2 2 2 1 1 1 2 2 1 2 2 2 2 1 2 2 2 2 1 1 1 2 2 1 1 2 2 1 2 1 1  
## [556] 2 1 1 2 2 3 2 2 2 3 1 3 2 3 2 2 1 1 1 1 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 1  
## [593] 1 2 2 1 2 3 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 1 1 1 2 2 2 2 1 2 2 1 2 1 2 2 3  
## [630] 2 1 1 1 1 1 2 1 1 2 1 2 1 2 2 1 1 1 2 2 1 1 1 2 2 1 2 2 2 2 1 2 2 3 2 2 2  
## [667] 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 2 1 2 2 2 2 1 1 2 1 2 1 1 1 3 3 1 2 2 2  
## [704] 1 1 1 2 2 1 1 1 2 1 1 1 2 1 3 1 1 2 1 1 1 1 1 2 2 1 2 2 2 2 1 1 1 1 2 2 1  
## [741] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [778] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1  
## [815] 1 1 2 2 1 2 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 2 1 1 1 1 1 2 1 1  
## [852] 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 1  
## [889] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1  
## [926] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1  
## [963] 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [1000] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1  
##   
## Within cluster sum of squares by cluster:  
## [1] 1.182420e+14 1.164207e+14 2.460167e+14  
##   
## Available components:  
##   
## [1] "cluster" "centers" "withinss" "size"   
##   
## [[4]]  
## K-means clustering with 4 clusters of sizes 14, 464, 409, 130  
##   
## Cluster means:  
## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 52.75714 53.91214 50.72786 52.24821 8536187 36.12650  
## [2,] 55.84135 56.63908 55.02256 55.85927 2727072 39.07965  
## [3,] 65.74045 66.29278 65.10391 65.76056 1558529 47.60363  
## [4,] 54.35231 55.49973 53.19831 54.43581 4483335 37.99674  
##   
## Clustering vector:  
## [1] 3 3 2 3 3 2 2 2 2 2 3 2 2 2 2 3 2 2 4 4 2 2 2 2 2 2 2 2 2 4 3 2 2 4 2 2 2  
## [38] 2 2 3 3 2 3 3 2 2 2 2 4 3 3 3 2 2 4 2 4 2 2 2 3 3 2 4 2 2 2 3 3 2 2 2 4 3  
## [75] 4 2 2 2 3 2 4 2 2 3 2 2 4 1 4 4 4 2 4 2 2 1 4 4 2 4 2 2 3 3 2 2 2 2 2 2 3  
## [112] 4 2 2 3 2 2 4 2 3 2 2 2 2 2 2 4 2 3 3 2 2 2 3 1 4 4 4 2 2 3 2 3 3 2 2 3 2  
## [149] 3 4 2 2 2 3 3 3 2 2 3 2 3 2 2 3 3 2 3 3 3 2 2 3 2 3 3 2 2 4 2 2 2 2 1 4 2  
## [186] 2 2 4 2 2 2 2 4 2 2 2 2 2 3 2 4 2 4 3 2 3 2 3 3 2 2 2 2 2 2 4 2 2 3 2 2 3  
## [223] 2 3 3 3 2 2 2 2 2 3 2 2 2 2 4 2 2 2 2 4 2 3 3 3 3 2 3 3 3 3 2 2 3 3 3 2 4  
## [260] 3 2 3 2 2 2 4 3 2 4 4 4 2 2 2 2 4 2 2 2 2 3 2 4 2 2 2 2 2 2 2 2 3 2 2 2 2  
## [297] 3 2 2 2 2 2 3 2 4 2 2 4 4 4 4 2 2 2 4 3 3 2 2 2 2 3 2 2 4 2 2 2 2 3 2 2 2  
## [334] 2 3 3 2 2 3 2 2 4 2 4 2 2 1 1 2 3 2 3 2 4 2 3 3 2 3 3 3 3 3 3 3 3 2 4 2 3  
## [371] 2 2 2 4 2 2 2 2 4 3 2 2 2 2 3 3 2 2 2 2 4 2 4 4 2 3 3 2 3 2 2 2 2 2 2 4 4  
## [408] 4 4 1 1 1 4 4 2 2 2 4 1 4 4 4 4 2 2 2 4 4 2 4 4 4 2 4 4 4 4 4 1 2 2 2 4 4  
## [445] 4 4 2 2 1 1 4 4 4 2 2 2 2 2 4 4 2 2 4 2 2 2 4 2 2 4 2 4 2 2 2 4 2 2 3 2 2  
## [482] 2 2 2 2 2 2 2 2 2 4 2 2 2 2 3 2 2 2 2 4 2 4 2 2 3 3 3 3 3 3 3 3 2 3 2 3 2  
## [519] 2 3 3 2 3 2 2 2 2 2 2 3 2 2 3 2 4 2 2 3 2 2 4 2 3 3 2 2 2 3 3 2 2 3 2 2 3  
## [556] 2 3 2 4 2 1 2 2 4 4 2 4 2 4 2 2 2 2 3 2 2 2 2 3 2 2 2 2 4 2 2 2 2 2 2 2 3  
## [593] 2 4 2 3 2 4 2 2 2 2 4 2 2 2 4 2 4 2 2 2 2 3 3 2 2 2 4 2 2 2 2 3 2 2 2 4 4  
## [630] 2 2 3 2 3 2 4 2 2 2 2 2 3 2 2 3 3 3 2 2 3 3 3 2 4 2 2 2 4 2 2 2 4 4 2 2 2  
## [667] 3 3 3 3 3 3 3 2 3 3 3 2 2 3 3 3 4 2 3 2 4 4 2 3 3 2 2 2 3 2 3 4 4 3 2 2 2  
## [704] 3 2 3 4 2 3 3 3 2 3 3 3 2 2 4 3 2 2 3 3 3 3 2 2 2 2 2 4 2 2 3 3 3 3 2 2 2  
## [741] 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 2 3 3 3 2 2 3 3 3 3 3 3  
## [778] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 3 3 2 3 3 3 3 3 3  
## [815] 3 3 2 4 3 2 2 2 3 3 3 3 2 2 3 3 3 3 3 2 3 3 3 3 2 2 2 3 2 2 3 3 3 3 2 3 3  
## [852] 3 3 3 2 3 2 2 3 2 3 3 3 2 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 2 3 3 3 4 4 2 2 2  
## [889] 3 2 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3  
## [926] 3 3 3 3 3 3 3 3 3 3 3 3 3 4 2 3 3 3 3 3 3 3 3 2 2 3 3 3 3 3 3 3 3 3 3 3 3  
## [963] 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 4 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [1000] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3  
##   
## Within cluster sum of squares by cluster:  
## [1] 1.089163e+14 6.949191e+13 6.433384e+13 5.606045e+13  
##   
## Available components:  
##   
## [1] "cluster" "centers" "withinss" "size"   
##   
## [[5]]  
## K-means clustering with 5 clusters of sizes 13, 112, 351, 298, 243  
##   
## Cluster means:  
## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 51.96692 53.14769 49.82923 51.40808 8688140 35.58659  
## [2,] 53.54848 54.68049 52.35973 53.59790 4631749 37.35742  
## [3,] 57.75177 58.41536 57.06222 57.80675 2156030 40.89390  
## [4,] 56.18369 57.06745 55.29757 56.19586 3025998 39.23764  
## [5,] 69.61325 70.11930 68.98912 69.60944 1295986 50.84343  
##   
## Clustering vector:  
## [1] 3 5 3 3 3 3 4 3 4 4 5 4 3 4 4 3 4 4 2 2 3 3 3 3 4 4 3 3 4 2 3 3 3 2 4 3 3  
## [38] 3 3 3 3 4 3 3 3 3 3 3 2 3 5 5 4 4 2 4 2 3 4 4 3 3 3 2 4 3 3 5 3 3 4 3 2 3  
## [75] 2 4 4 3 3 4 2 4 3 5 4 4 2 1 2 2 2 4 2 4 4 1 2 2 3 2 4 4 3 5 3 3 3 4 3 4 3  
## [112] 2 4 4 3 4 3 2 4 3 3 4 4 3 3 3 2 4 5 3 4 4 3 5 1 2 2 2 3 3 3 3 3 3 4 3 3 3  
## [149] 3 2 4 4 4 5 3 5 4 3 5 4 3 4 3 5 5 4 3 3 5 4 4 3 4 5 3 4 4 2 3 4 4 4 1 2 4  
## [186] 4 3 2 3 4 4 4 2 4 4 4 4 4 5 4 4 3 4 3 3 3 3 3 3 3 4 4 4 4 3 2 4 4 3 3 3 5  
## [223] 4 3 3 5 3 4 4 4 4 3 3 3 4 4 2 4 3 4 4 2 4 3 3 5 5 4 5 3 5 5 4 4 5 5 3 3 2  
## [260] 3 4 3 3 3 4 2 5 4 2 2 2 4 3 4 3 4 4 3 4 3 3 3 2 4 3 4 3 3 4 4 3 3 3 4 4 4  
## [297] 3 4 4 3 4 3 3 4 2 4 4 2 4 4 2 4 3 4 4 3 3 3 3 3 4 3 3 4 2 4 4 4 3 5 4 3 3  
## [334] 4 5 5 4 4 3 3 4 2 4 2 4 4 2 1 3 3 3 3 3 2 4 3 3 4 3 5 3 3 3 3 5 5 4 2 4 5  
## [371] 4 3 4 2 3 4 4 4 2 3 4 4 4 3 5 3 3 4 3 4 2 4 4 4 3 5 3 4 5 3 3 4 4 3 4 2 2  
## [408] 2 2 1 1 1 2 2 4 4 3 2 1 2 2 2 2 4 3 4 2 2 4 2 2 4 4 2 2 2 2 2 1 4 4 3 2 2  
## [445] 2 2 4 4 1 1 2 2 2 4 4 4 4 4 2 2 3 4 2 4 4 4 2 4 4 2 4 2 4 4 4 4 4 3 3 3 4  
## [482] 4 4 4 4 4 4 4 4 4 2 4 4 3 3 3 4 4 3 3 2 4 2 4 3 3 5 5 5 5 5 5 5 4 3 4 3 4  
## [519] 4 3 3 4 3 4 4 4 4 3 3 3 4 3 3 3 4 3 4 3 4 4 2 4 5 3 3 4 4 5 5 4 4 3 4 3 3  
## [556] 4 3 3 2 3 1 3 3 2 2 3 2 3 2 4 4 3 3 3 3 4 4 4 3 4 3 4 4 4 4 4 3 3 4 4 4 3  
## [593] 3 4 4 3 4 2 4 3 4 4 2 4 3 4 2 4 2 3 4 4 3 5 5 3 4 4 2 4 3 4 4 3 3 3 4 4 2  
## [630] 4 3 3 3 3 3 2 3 3 4 3 3 3 4 4 3 3 3 4 3 5 3 3 4 4 3 4 4 4 4 3 3 2 2 4 4 4  
## [667] 3 3 5 5 5 5 3 4 5 3 3 3 3 5 5 5 2 4 5 4 4 2 4 3 3 4 3 4 5 3 3 2 2 5 3 4 3  
## [704] 3 3 3 4 4 5 3 3 4 3 5 3 4 3 2 3 3 3 3 3 5 5 3 3 4 3 4 2 4 4 5 3 5 3 3 4 3  
## [741] 3 3 3 5 5 3 3 5 5 5 3 3 3 5 3 3 3 5 5 5 5 5 4 5 5 3 5 5 5 3 3 3 5 3 5 5 5  
## [778] 5 5 5 5 5 5 5 5 5 3 3 3 3 5 5 5 5 5 5 5 5 5 3 5 3 4 4 3 3 5 3 5 3 5 5 5 5  
## [815] 5 5 4 2 5 4 3 4 5 5 5 3 4 3 5 3 3 5 5 3 3 3 5 5 4 3 4 3 3 3 5 5 5 5 4 3 5  
## [852] 5 5 5 3 3 4 3 3 4 3 5 5 3 5 5 5 3 5 3 5 5 5 3 3 5 5 3 3 3 5 3 5 4 2 4 4 3  
## [889] 3 3 5 3 3 5 3 5 3 5 5 5 5 5 5 5 3 5 5 3 5 5 5 5 5 4 5 5 5 5 5 5 5 5 5 5 3  
## [926] 5 5 5 5 5 5 5 5 5 5 5 5 5 2 3 5 3 5 5 5 5 5 5 3 4 5 5 5 5 5 5 5 5 5 5 5 5  
## [963] 5 5 5 3 3 5 4 5 3 5 5 5 5 5 5 3 5 5 2 5 3 3 5 5 5 5 3 5 5 5 5 5 5 5 5 5 5  
## [1000] 5 3 3 5 5 5 5 5 3 5 5 5 3 3 4 5 5 5  
##   
## Within cluster sum of squares by cluster:  
## [1] 1.047139e+14 4.671611e+13 1.967260e+13 3.204509e+13 2.055148e+13  
##   
## Available components:  
##   
## [1] "cluster" "centers" "withinss" "size"   
##   
## [[6]]  
## K-means clustering with 6 clusters of sizes 38, 3, 344, 106, 237, 289  
##   
## Cluster means:  
## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 55.38684 56.67974 53.63250 55.07579 6033135 38.18529  
## [2,] 55.26667 56.66833 53.65833 56.41667 13309756 38.81306  
## [3,] 57.92683 58.58808 57.23981 57.97406 2132815 41.04293  
## [4,] 53.88736 55.00085 52.86566 54.04217 4187962 37.76887  
## [5,] 69.69363 70.19983 69.07260 69.69264 1285343 50.92648  
## [6,] 55.97180 56.81118 55.08813 55.95872 2954025 39.06181  
##   
## Clustering vector:  
## [1] 3 5 3 3 3 3 6 3 6 6 5 6 6 6 6 3 6 6 4 4 3 3 3 3 6 6 3 3 6 4 3 6 3 4 6 3 3  
## [38] 3 3 3 3 6 3 3 3 3 3 3 4 3 5 5 6 6 4 6 4 3 6 6 3 3 3 4 6 3 3 5 3 3 6 3 4 3  
## [75] 4 6 6 3 3 6 4 6 3 5 6 6 4 1 1 4 4 6 4 6 6 1 4 4 3 4 6 6 3 5 3 3 3 6 3 6 3  
## [112] 4 6 6 3 6 3 4 6 3 3 6 6 3 3 3 4 6 5 3 6 6 3 5 1 1 4 4 3 3 3 3 3 3 6 3 3 3  
## [149] 3 4 6 6 6 5 3 5 6 3 5 6 3 6 3 5 5 6 3 3 5 6 6 3 6 5 3 6 6 4 3 6 6 6 2 4 6  
## [186] 6 3 4 3 6 6 6 4 6 6 6 6 6 5 6 4 3 4 3 3 3 3 3 3 3 6 6 6 6 3 1 6 6 3 3 3 3  
## [223] 4 3 3 5 3 6 6 6 6 3 3 3 6 6 4 6 3 6 6 1 6 3 3 5 5 6 5 3 5 5 6 6 5 5 3 3 4  
## [260] 3 6 3 3 3 6 4 5 6 4 4 1 6 6 6 3 4 6 3 6 3 3 6 1 6 3 6 3 3 6 6 3 3 3 6 4 6  
## [297] 3 6 6 3 6 3 3 6 4 6 6 1 4 4 1 6 6 6 4 3 3 3 3 3 6 3 3 6 4 6 6 6 3 5 6 3 3  
## [334] 6 5 5 6 6 3 3 6 1 6 4 6 6 1 2 3 3 3 3 3 1 6 3 3 6 3 5 3 3 3 3 5 5 6 4 6 5  
## [371] 6 3 6 1 3 6 6 6 4 3 6 6 6 6 5 3 3 6 3 6 1 4 4 4 3 3 3 6 5 3 3 6 6 3 6 4 4  
## [408] 1 1 1 2 1 1 4 6 6 3 1 1 1 4 4 1 6 3 6 4 4 6 4 4 4 6 4 4 4 1 4 1 6 6 3 4 1  
## [445] 1 4 6 6 1 1 4 4 4 6 6 6 6 6 4 4 3 6 4 6 6 6 4 6 6 4 6 1 6 6 6 4 6 3 3 3 6  
## [482] 6 6 6 6 6 6 6 6 6 1 6 6 3 3 3 6 6 3 3 4 6 1 6 3 3 5 5 5 5 5 5 5 6 3 6 3 6  
## [519] 6 3 3 6 3 6 6 6 6 3 3 3 6 3 3 6 4 6 6 3 6 6 4 6 5 3 3 6 6 5 5 6 6 3 6 3 3  
## [556] 6 3 3 4 6 1 3 3 4 4 3 4 3 1 6 6 3 3 3 3 6 6 6 3 6 6 6 6 4 6 6 3 3 6 6 6 3  
## [593] 3 4 6 3 6 1 6 3 6 6 4 6 3 6 4 6 4 3 6 6 3 5 5 3 6 6 4 6 3 6 6 3 3 3 6 4 1  
## [630] 6 3 3 3 3 3 4 3 3 6 3 6 3 6 6 3 3 3 6 3 5 3 3 6 4 3 6 6 4 6 3 3 4 4 6 6 6  
## [667] 3 3 5 5 5 5 3 6 5 3 3 3 3 5 5 5 4 6 3 6 4 4 6 3 3 6 3 6 5 3 3 4 4 5 6 6 3  
## [704] 3 3 3 4 6 5 3 3 6 3 5 3 6 3 4 3 3 6 3 3 5 5 3 3 6 3 6 4 6 6 5 3 5 3 3 6 3  
## [741] 3 3 3 5 5 3 3 5 5 5 3 3 3 5 3 3 3 5 5 5 5 5 6 5 5 3 5 5 5 3 3 3 5 3 5 5 5  
## [778] 5 5 5 5 5 5 5 5 5 3 3 3 3 5 5 5 5 5 5 5 5 5 3 5 3 6 6 3 3 5 3 5 3 5 5 5 5  
## [815] 5 5 6 4 5 6 3 6 5 3 5 3 6 3 5 3 3 5 5 3 3 3 5 5 6 3 6 3 3 3 5 5 5 5 6 3 5  
## [852] 5 3 5 3 3 6 3 3 6 3 5 5 3 5 5 5 3 5 3 5 5 5 3 3 5 5 3 3 3 5 3 5 4 4 6 6 3  
## [889] 3 3 5 3 3 5 3 5 3 5 5 5 5 5 5 5 3 5 5 3 5 5 5 5 3 6 5 5 5 5 5 5 5 5 5 5 3  
## [926] 5 5 5 5 5 5 5 5 5 5 5 5 5 4 3 5 3 5 5 5 5 5 5 3 6 5 5 5 5 5 5 5 5 5 5 5 5  
## [963] 5 5 5 3 3 5 6 5 3 5 5 5 5 5 5 3 5 5 4 5 3 3 5 5 5 5 3 5 5 5 5 5 5 5 5 5 5  
## [1000] 5 3 3 5 5 5 5 5 3 5 5 5 3 3 6 5 5 5  
##   
## Within cluster sum of squares by cluster:  
## [1] 2.719466e+13 1.979307e+13 1.844339e+13 1.610130e+13 1.946418e+13  
## [6] 2.317594e+13  
##   
## Available components:  
##   
## [1] "cluster" "centers" "withinss" "size"   
##   
## [[7]]  
## K-means clustering with 7 clusters of sizes 334, 278, 42, 2, 111, 13, 237  
##   
## Cluster means:  
## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 57.94093 58.59433 57.25120 57.97907 2120924 41.07010  
## [2,] 56.06563 56.89518 55.18685 56.05083 2898306 39.11850  
## [3,] 54.37250 55.57905 52.96226 54.16476 5265605 37.58825  
## [4,] 55.10500 55.61750 54.63250 55.31000 15098987 37.77691  
## [5,] 54.68360 55.80216 53.70689 54.92712 3957920 38.41447  
## [6,] 52.53769 53.82269 50.36500 52.01192 7362604 35.99902  
## [7,] 69.69363 70.19983 69.07260 69.69264 1285343 50.92648  
##   
## Clustering vector:  
## [1] 1 7 1 1 1 1 2 1 2 2 7 2 2 2 2 1 2 2 3 3 1 1 1 1 5 2 1 1 2 5 1 2 1 5 2 1 1  
## [38] 1 1 1 1 2 1 1 1 1 1 1 5 1 7 7 2 2 5 2 5 1 2 2 1 1 1 5 2 1 1 7 1 1 2 1 5 1  
## [75] 5 2 2 1 1 2 5 2 1 7 5 2 5 6 3 5 5 2 5 5 2 6 3 3 1 5 2 2 1 7 1 1 1 2 1 2 1  
## [112] 5 2 2 1 2 1 5 2 1 1 2 2 2 1 1 5 2 7 1 2 2 1 7 6 3 3 5 1 1 1 1 1 1 2 1 1 1  
## [149] 1 3 2 2 2 7 1 7 2 1 7 2 1 2 1 7 7 2 1 1 7 2 2 1 2 7 1 2 5 5 1 2 2 2 4 5 2  
## [186] 2 2 5 1 2 5 2 3 2 2 2 2 2 7 2 5 1 5 1 1 1 1 1 1 1 2 2 5 2 1 3 5 2 1 1 1 1  
## [223] 5 1 1 7 1 2 2 2 2 1 1 1 2 5 5 2 2 2 2 3 2 1 1 7 7 2 7 1 7 7 5 2 7 7 1 1 5  
## [260] 1 2 1 2 1 2 5 7 2 5 3 3 2 2 2 2 5 2 1 2 1 1 2 3 5 2 2 1 1 2 2 1 1 1 2 5 2  
## [297] 1 5 5 1 2 1 1 2 3 2 2 3 5 5 6 2 2 2 5 1 1 1 1 1 2 1 1 2 5 2 2 2 1 7 2 1 1  
## [334] 2 7 7 2 2 1 1 2 3 5 5 2 2 6 4 1 1 1 1 1 3 2 1 1 2 1 7 1 1 1 1 7 7 2 5 2 7  
## [371] 2 1 2 3 1 2 2 2 3 1 2 5 2 2 7 1 1 2 1 2 3 5 5 5 1 1 1 2 7 1 1 2 2 1 2 5 5  
## [408] 3 3 6 6 6 3 5 2 2 1 3 6 3 3 5 3 2 1 2 5 5 2 5 5 5 2 5 5 5 3 3 6 2 2 1 3 3  
## [445] 3 5 2 2 6 6 5 5 5 2 2 2 2 2 5 5 1 2 5 2 2 2 5 2 2 5 2 3 2 2 2 5 2 1 1 1 2  
## [482] 2 2 2 2 2 2 2 2 2 3 2 2 2 1 1 2 2 1 1 5 2 3 2 1 1 7 7 7 7 7 7 7 2 1 2 1 2  
## [519] 2 1 1 2 1 5 2 2 2 1 1 1 2 1 1 2 5 2 2 1 2 2 5 2 7 1 1 2 2 7 7 2 2 1 2 1 1  
## [556] 2 1 1 5 2 6 1 2 5 5 1 3 1 3 2 2 1 1 1 1 2 2 2 1 2 2 2 2 5 2 2 1 1 2 2 2 1  
## [593] 1 5 2 1 2 3 2 1 2 2 5 2 1 2 5 2 5 1 2 2 1 7 7 1 2 2 5 2 1 2 2 1 1 1 2 5 3  
## [630] 2 1 1 1 1 1 5 1 1 5 1 2 1 2 2 1 1 1 2 2 7 1 1 2 5 1 2 2 5 2 1 1 5 3 5 2 2  
## [667] 1 1 7 7 7 7 1 2 7 1 1 1 1 7 7 7 5 2 1 2 5 5 2 1 1 2 1 5 7 1 1 5 5 7 2 2 1  
## [704] 1 1 1 5 2 7 1 1 2 1 7 1 2 1 3 1 1 2 1 1 7 7 1 1 5 1 5 5 2 2 7 1 7 1 2 2 1  
## [741] 1 1 1 7 7 1 1 7 7 7 1 1 1 7 1 1 1 7 7 7 7 7 2 7 7 1 7 7 7 1 1 1 7 1 7 7 7  
## [778] 7 7 7 7 7 7 7 7 7 1 1 1 1 7 7 7 7 7 7 7 7 7 1 7 1 2 2 1 1 7 1 7 1 7 7 7 7  
## [815] 7 7 2 5 7 2 1 2 7 1 7 1 2 1 7 1 1 7 7 1 1 1 7 7 2 1 2 1 1 1 7 7 7 7 2 1 7  
## [852] 7 1 7 1 1 2 1 1 2 1 7 7 1 7 7 7 1 7 1 7 7 7 1 1 7 7 1 1 1 7 1 7 5 5 5 2 1  
## [889] 1 1 7 1 1 7 1 7 1 7 7 7 7 7 7 7 1 7 7 1 7 7 7 7 1 2 7 7 7 7 7 7 7 7 7 7 1  
## [926] 7 7 7 7 7 7 7 7 7 7 7 7 7 5 1 7 1 7 7 7 7 7 7 1 2 7 7 7 7 7 7 7 7 7 7 7 7  
## [963] 7 7 7 1 1 7 2 7 1 7 7 7 7 7 7 1 7 7 5 7 1 1 7 7 7 7 1 7 7 7 7 7 7 7 7 7 7  
## [1000] 7 1 1 7 7 7 7 7 1 7 7 7 1 1 2 7 7 7  
##   
## Within cluster sum of squares by cluster:  
## [1] 1.681839e+13 1.806076e+13 6.259478e+12 5.849888e+11 1.291091e+13  
## [6] 8.826774e+12 1.946418e+13  
##   
## Available components:  
##   
## [1] "cluster" "centers" "withinss" "size"   
##   
## [[8]]  
## K-means clustering with 8 clusters of sizes 227, 207, 2, 194, 25, 86, 139, 137  
##   
## Cluster means:  
## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 56.80874 57.48608 56.10346 56.84925 2223026 40.12321  
## [2,] 56.91720 57.70828 56.04386 56.92164 2743745 39.77632  
## [3,] 55.10500 55.61750 54.63250 55.31000 15098986 37.77691  
## [4,] 63.45492 64.02840 62.79314 63.45758 1692785 45.67114  
## [5,] 55.92440 57.22660 53.83840 55.35220 6602297 38.41219  
## [6,] 52.90512 54.06703 51.84738 53.08413 4533176 37.04324  
## [7,] 72.43500 72.89522 71.85572 72.43518 1097110 53.25828  
## [8,] 54.65916 55.64562 53.74788 54.71839 3405482 38.11562  
##   
## Clustering vector:  
## [1] 4 4 1 1 4 1 8 1 2 2 4 2 2 8 8 4 2 2 6 6 1 1 1 1 8 8 1 1 8 8 1 2 1 6 8 1 1  
## [38] 1 1 1 4 2 1 4 1 1 1 1 6 4 4 4 2 2 6 2 6 1 2 2 1 4 2 6 2 1 1 4 4 1 2 1 6 4  
## [75] 6 2 2 1 4 8 8 2 1 7 8 8 6 5 6 6 6 8 6 8 8 5 6 6 1 6 2 2 1 7 1 1 1 8 1 2 1  
## [112] 6 2 2 1 8 1 6 8 1 1 2 2 2 1 2 8 2 7 4 2 8 1 7 5 6 6 8 1 2 1 1 4 4 2 1 4 1  
## [149] 4 6 2 8 8 4 1 4 8 1 4 8 1 2 1 4 4 2 4 1 7 2 8 1 2 7 1 2 8 6 1 2 8 2 3 6 8  
## [186] 2 2 6 1 2 8 8 6 8 8 8 2 2 4 2 8 1 8 1 1 4 1 4 1 1 2 2 8 8 2 6 8 8 4 1 1 4  
## [223] 8 4 4 4 1 2 8 2 8 1 1 1 8 8 6 2 2 2 8 6 8 4 4 4 4 2 4 4 7 7 8 8 4 7 1 1 8  
## [260] 1 2 4 2 1 8 6 4 8 6 6 6 2 2 2 2 8 2 1 8 2 4 2 5 8 2 2 1 1 8 2 1 1 1 2 8 2  
## [297] 1 8 8 1 8 1 1 2 6 2 2 5 8 8 5 8 2 2 8 4 1 1 1 1 8 1 1 2 6 2 8 2 2 4 2 1 1  
## [334] 2 4 7 2 2 4 1 2 5 8 8 8 2 5 3 1 1 1 4 1 5 2 1 4 2 4 7 1 1 4 4 4 7 2 6 2 7  
## [371] 2 1 2 6 1 2 8 2 6 1 2 8 2 2 7 4 1 2 1 2 6 8 8 8 1 4 4 2 4 1 1 8 8 1 2 6 6  
## [408] 5 5 5 5 5 5 6 2 8 1 5 5 5 6 6 6 8 1 2 6 6 8 6 6 8 2 8 6 8 6 6 5 8 2 1 6 6  
## [445] 6 6 2 8 5 5 6 6 6 8 2 8 8 8 6 6 1 2 8 2 8 2 6 8 2 6 2 5 2 2 2 8 2 1 4 1 8  
## [482] 8 2 8 2 8 2 2 8 2 6 2 2 2 1 1 2 2 1 2 6 8 6 8 1 4 7 7 7 7 7 7 4 2 1 2 4 8  
## [519] 2 1 1 2 1 8 2 2 8 1 1 4 2 1 1 2 8 2 2 1 8 2 6 2 4 1 1 2 2 4 4 2 2 1 2 1 1  
## [556] 2 4 1 6 2 5 2 2 8 6 1 6 1 5 2 2 1 1 4 1 8 8 8 1 2 2 2 8 8 2 2 2 1 2 2 2 4  
## [593] 1 8 2 4 2 6 2 1 2 2 6 2 1 2 6 2 6 1 2 2 1 4 7 1 2 2 6 2 1 8 8 1 2 1 2 8 5  
## [630] 8 1 4 1 4 1 8 1 1 8 1 2 4 2 8 4 4 4 2 2 4 1 4 2 8 1 2 8 8 2 1 1 6 6 8 2 2  
## [667] 4 1 4 4 4 4 1 2 4 1 4 1 1 7 7 7 8 2 4 8 8 6 8 1 1 8 1 8 4 1 1 6 6 4 2 2 1  
## [704] 1 1 1 8 2 4 4 4 2 4 4 1 2 1 6 1 1 2 1 4 7 7 1 2 8 1 8 6 2 2 4 1 7 4 2 2 1  
## [741] 1 1 4 4 7 1 1 4 4 7 4 1 4 7 4 4 4 4 7 7 4 7 8 7 7 1 4 7 7 1 1 4 4 1 4 7 7  
## [778] 4 7 4 7 7 4 7 7 7 1 4 1 1 4 4 7 7 7 7 7 4 7 4 4 1 2 2 1 4 4 1 4 1 4 7 4 7  
## [815] 7 4 2 6 4 8 1 2 4 4 4 4 2 1 4 4 1 7 4 1 4 1 4 4 2 1 2 4 1 1 4 4 4 4 2 1 7  
## [852] 4 4 7 1 1 8 1 4 2 4 4 4 1 4 7 4 4 4 1 4 7 4 4 1 7 4 4 1 1 4 1 4 8 6 8 2 1  
## [889] 4 1 4 1 1 4 4 4 4 7 4 4 7 7 7 7 4 7 7 4 7 7 7 7 4 2 7 7 4 7 7 7 7 7 7 7 4  
## [926] 4 7 7 7 7 7 4 4 7 4 7 7 7 6 1 4 4 7 7 7 7 7 7 1 8 7 7 7 7 7 7 7 4 7 7 7 7  
## [963] 7 7 7 1 1 4 2 7 1 7 4 4 4 7 7 4 7 7 6 4 1 4 7 7 7 7 1 4 7 7 7 7 7 7 7 7 7  
## [1000] 7 4 4 7 7 7 7 7 4 7 4 7 4 4 2 7 7 7  
##   
## Within cluster sum of squares by cluster:  
## [1] 4.625655e+12 4.777994e+12 5.849888e+11 5.144955e+12 2.474972e+13  
## [6] 1.620837e+13 6.784487e+12 8.131639e+12  
##   
## Available components:  
##   
## [1] "cluster" "centers" "withinss" "size"   
##   
## [[9]]  
## K-means clustering with 9 clusters of sizes 203, 194, 77, 139, 13, 121, 41, 2, 227  
##   
## Cluster means:  
## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 56.94049 57.73983 56.07128 56.94458 2737656 39.80042  
## [2,] 63.45492 64.02840 62.79314 63.45758 1692785 45.67114  
## [3,] 53.63162 54.70539 52.57195 53.78487 4149542 37.58929  
## [4,] 72.43500 72.89522 71.85572 72.43518 1097110 53.25828  
## [5,] 52.53769 53.82269 50.36500 52.01192 7362604 35.99902  
## [6,] 54.71483 55.69983 53.84236 54.80570 3321974 38.17290  
## [7,] 54.29451 55.52732 52.90719 54.13195 5280125 37.58032  
## [8,] 55.10500 55.61750 54.63250 55.31000 15098987 37.77691  
## [9,] 56.80874 57.48608 56.10346 56.84925 2223026 40.12321  
##   
## Clustering vector:  
## [1] 2 2 9 9 2 9 6 9 1 1 2 1 1 6 6 2 1 1 7 7 9 9 9 9 6 6 9 9 6 3 9 1 9 3 6 9 9  
## [38] 9 9 9 2 1 9 2 9 9 9 9 3 2 2 2 1 1 3 1 3 9 1 1 9 2 1 3 1 9 9 2 2 9 1 9 3 2  
## [75] 3 1 1 9 2 6 3 1 9 4 6 6 3 5 7 3 3 6 3 6 6 5 7 7 9 3 1 1 9 4 9 9 9 6 9 1 9  
## [112] 3 1 1 9 6 9 3 6 9 9 1 1 1 9 1 3 1 4 2 1 6 9 4 5 7 7 3 9 1 9 9 2 2 1 9 2 9  
## [149] 2 7 6 6 6 2 9 2 6 9 2 6 9 1 9 2 2 1 2 9 4 1 6 9 1 4 9 1 6 3 9 1 6 1 8 3 6  
## [186] 6 1 3 9 1 6 6 7 6 6 6 1 1 2 1 3 9 6 9 9 2 9 2 9 9 1 1 6 6 1 7 6 6 2 9 9 2  
## [223] 6 2 2 2 9 1 6 1 6 9 9 9 6 6 3 1 1 1 6 7 6 2 2 2 2 1 2 2 4 4 6 6 2 4 9 9 3  
## [260] 9 1 2 1 9 6 3 2 6 3 3 7 1 1 1 1 3 1 9 6 1 2 1 7 6 1 1 9 9 6 1 9 9 9 1 6 1  
## [297] 9 6 6 9 6 9 9 1 7 1 1 7 3 3 5 6 1 1 6 2 9 9 9 9 6 9 9 1 3 1 6 1 1 2 1 9 9  
## [334] 1 2 4 1 1 2 9 1 7 6 3 6 1 5 8 9 9 9 2 9 7 1 9 2 1 2 4 9 9 2 2 2 4 1 3 1 4  
## [371] 1 9 1 7 9 1 6 1 7 9 1 6 1 1 4 2 9 1 9 1 7 6 6 3 9 2 2 1 2 9 9 6 6 9 1 3 3  
## [408] 7 7 5 5 5 7 3 1 6 9 7 5 7 7 3 7 6 9 1 3 3 6 3 3 6 1 3 3 3 7 7 5 6 1 9 7 7  
## [445] 7 3 1 6 5 5 3 3 3 6 1 6 6 6 3 3 9 1 3 1 6 1 3 6 1 3 1 7 1 1 1 6 1 9 2 9 6  
## [482] 6 1 6 1 6 1 1 6 1 7 1 1 1 9 9 1 1 9 1 3 6 7 6 9 2 4 4 4 4 4 4 2 1 9 1 2 6  
## [519] 1 9 9 1 9 6 6 1 6 9 9 2 1 9 9 1 6 1 1 9 6 1 3 1 2 9 9 1 1 2 2 1 1 9 6 9 9  
## [556] 1 2 9 3 1 5 1 1 3 3 9 7 9 7 1 1 9 9 2 9 6 6 6 9 1 1 1 6 6 1 1 1 9 1 1 1 2  
## [593] 9 3 1 2 1 7 1 9 1 1 3 1 9 1 3 1 3 9 1 1 9 2 4 9 1 1 3 1 9 6 6 9 1 9 1 6 7  
## [630] 6 9 2 9 2 9 3 9 9 6 9 1 2 1 6 2 2 2 1 1 2 9 2 1 3 9 1 6 6 1 9 9 3 7 6 1 1  
## [667] 2 9 2 2 2 2 9 1 2 9 2 9 9 4 4 4 3 1 2 6 6 3 6 9 9 6 9 6 2 9 9 3 3 2 1 1 9  
## [704] 9 9 9 3 1 2 2 2 1 2 2 9 1 9 7 9 9 1 9 2 4 4 9 1 6 9 6 3 1 1 2 9 4 2 1 1 9  
## [741] 9 9 2 2 4 9 9 2 2 4 2 9 2 4 2 2 2 2 4 4 2 4 6 4 4 9 2 4 4 9 9 2 2 9 2 4 4  
## [778] 2 4 2 4 4 2 4 4 4 9 2 9 9 2 2 4 4 4 4 4 2 4 2 2 9 1 1 9 2 2 9 2 9 2 4 2 4  
## [815] 4 2 1 3 2 6 9 1 2 2 2 2 1 9 2 2 9 4 2 9 2 9 2 2 1 9 1 2 9 9 2 2 2 2 1 9 4  
## [852] 2 2 4 9 9 6 9 2 1 2 2 2 9 2 4 2 2 2 9 2 4 2 2 9 4 2 2 9 9 2 9 2 6 3 6 1 9  
## [889] 2 9 2 9 9 2 2 2 2 4 2 2 4 4 4 4 2 4 4 2 4 4 4 4 2 1 4 4 2 4 4 4 4 4 4 4 2  
## [926] 2 4 4 4 4 4 2 2 4 2 4 4 4 3 9 2 2 4 4 4 4 4 4 9 6 4 4 4 4 4 4 4 2 4 4 4 4  
## [963] 4 4 4 9 9 2 1 4 9 4 2 2 2 4 4 2 4 4 3 2 9 2 4 4 4 4 9 2 4 4 4 4 4 4 4 4 4  
## [1000] 4 2 2 4 4 4 4 4 2 4 2 4 2 2 1 4 4 4  
##   
## Within cluster sum of squares by cluster:  
## [1] 4.387782e+12 5.144955e+12 4.673033e+12 6.784487e+12 8.826774e+12  
## [6] 3.925031e+12 5.896407e+12 5.849888e+11 4.625655e+12  
##   
## Available components:  
##   
## [1] "cluster" "centers" "withinss" "size"   
##   
## [[10]]  
## K-means clustering with 10 clusters of sizes 193, 132, 2, 13, 153, 41, 199, 120, 77, 87  
##   
## Cluster means:  
## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 57.12624 57.92179 56.25153 57.11842 2752772.4 39.93245  
## [2,] 67.46360 68.02902 66.79951 67.49360 1439742.5 49.07563  
## [3,] 55.10500 55.61750 54.63250 55.31000 15098986.5 37.77691  
## [4,] 52.53769 53.82269 50.36500 52.01192 7362604.0 35.99902  
## [5,] 62.14706 62.73003 61.48281 62.16592 1863165.4 44.54885  
## [6,] 54.29451 55.52732 52.90719 54.13195 5280125.4 37.58032  
## [7,] 55.52663 56.22058 54.81447 55.56043 2279055.9 39.04716  
## [8,] 54.65579 55.64483 53.78371 54.74383 3324404.3 38.12937  
## [9,] 53.63162 54.70539 52.57195 53.78487 4149541.9 37.58929  
## [10,] 73.63408 74.05109 73.09908 73.62184 969799.5 54.28332  
##   
## Clustering vector:  
## [1] 5 2 7 5 5 7 8 7 1 1 2 1 1 8 8 5 1 1 6 6 7 7 7 7  
## [25] 8 8 7 7 8 9 7 1 7 9 8 7 7 7 7 7 5 1 5 5 7 7 7 7  
## [49] 9 5 5 2 1 1 9 1 9 7 1 1 7 5 7 9 1 7 7 2 5 7 1 7  
## [73] 9 5 9 1 1 7 5 8 9 1 7 10 8 8 9 4 6 9 9 8 9 8 8 4  
## [97] 6 6 7 9 1 1 5 10 7 7 7 8 7 1 7 9 1 1 7 8 7 9 8 5  
## [121] 7 1 1 1 7 7 9 1 2 5 1 8 7 2 4 6 6 9 7 7 7 7 5 5  
## [145] 1 7 5 7 5 6 8 8 8 2 7 2 8 7 2 8 7 1 7 2 2 1 5 5  
## [169] 10 1 8 7 1 2 7 1 8 9 7 1 8 1 3 9 8 8 1 9 7 1 8 8  
## [193] 6 8 8 8 1 1 2 1 9 7 8 5 7 5 7 5 7 7 1 1 8 8 7 6  
## [217] 8 8 5 7 7 5 8 5 5 5 7 1 8 1 8 5 7 7 8 8 9 1 1 1  
## [241] 8 6 8 5 5 2 2 1 2 5 2 10 8 8 2 10 7 7 9 7 1 5 1 7  
## [265] 8 9 2 8 9 9 6 1 1 1 1 9 1 7 8 7 5 1 6 8 1 1 7 7  
## [289] 8 1 7 7 7 1 8 1 5 8 8 7 8 7 7 1 6 1 1 6 9 9 4 8  
## [313] 1 1 8 5 5 7 7 7 8 7 7 1 9 1 8 1 7 2 1 7 7 1 5 2  
## [337] 1 1 5 7 1 6 8 9 8 1 4 3 7 5 7 5 7 6 1 7 5 1 5 10  
## [361] 7 5 5 5 2 2 1 9 1 10 1 7 1 6 7 1 8 1 6 5 1 8 1 1  
## [385] 2 5 7 1 7 1 6 8 8 9 7 5 5 1 2 7 7 8 8 7 1 9 9 6  
## [409] 6 4 4 4 6 9 1 8 7 6 4 6 6 9 6 8 7 1 9 9 8 9 9 8  
## [433] 1 9 9 9 6 6 4 8 1 7 6 6 6 9 1 8 4 4 9 9 9 8 1 8  
## [457] 8 8 9 9 7 1 9 1 8 1 9 8 1 9 1 6 1 1 1 8 1 7 5 7  
## [481] 8 8 1 8 1 8 1 1 8 1 6 1 1 1 7 7 1 1 7 7 9 8 6 8  
## [505] 7 5 2 10 10 2 10 10 2 1 7 1 5 8 1 5 5 1 5 8 1 1 8 7  
## [529] 7 5 1 7 5 1 8 1 1 5 8 1 9 1 2 7 7 1 1 2 5 1 1 7  
## [553] 8 7 7 1 5 7 9 1 4 7 1 9 9 7 6 7 6 1 1 7 7 5 7 8  
## [577] 8 8 7 1 1 1 8 8 1 1 7 7 1 1 1 5 7 9 1 5 1 6 1 7  
## [601] 1 1 9 1 7 1 9 1 9 7 1 1 7 2 10 7 1 1 9 1 7 8 8 7  
## [625] 7 7 1 8 6 8 7 5 7 5 7 9 7 7 8 7 1 5 1 8 5 5 5 1  
## [649] 1 2 5 5 1 9 7 1 8 8 1 7 7 9 6 8 1 1 5 7 2 2 2 2  
## [673] 7 1 5 5 5 7 7 2 2 2 9 1 5 8 8 9 8 5 7 8 7 8 5 7  
## [697] 5 9 9 2 1 1 7 5 7 5 9 1 2 5 5 1 5 2 7 1 7 6 7 7  
## [721] 1 7 5 10 2 7 7 8 7 8 9 1 1 5 7 2 5 1 1 7 7 7 5 2  
## [745] 10 7 7 5 2 2 5 5 5 2 5 5 5 2 2 2 2 2 8 10 10 7 2 10  
## [769] 2 7 7 5 5 5 2 10 2 2 10 2 10 10 2 10 10 10 5 5 5 5 2 2  
## [793] 2 2 2 10 2 2 10 5 2 7 1 1 7 5 2 7 2 5 2 2 5 10 10 5  
## [817] 1 9 2 8 7 1 2 5 2 5 1 7 2 5 7 2 5 7 5 5 5 2 1 7  
## [841] 1 5 7 7 2 2 2 2 1 7 10 5 5 10 7 5 8 7 5 1 5 2 2 7  
## [865] 2 10 2 5 2 5 5 2 2 5 7 2 2 5 5 7 2 5 2 8 9 8 1 7  
## [889] 5 7 2 5 7 2 5 2 5 10 2 2 10 2 10 10 5 10 10 5 2 2 10 10  
## [913] 5 1 2 10 2 2 2 10 2 10 10 10 5 2 2 10 10 10 10 5 2 10 2 10  
## [937] 10 10 9 7 5 5 2 2 10 10 2 10 7 8 10 2 10 10 10 10 2 2 10 10  
## [961] 10 10 2 10 10 5 5 2 1 10 5 10 5 2 2 2 2 5 10 10 9 2 7 5  
## [985] 10 2 2 2 5 2 10 10 10 10 2 10 10 10 10 10 5 5 2 2 10 2 10 5  
## [1009] 10 2 10 5 5 1 10 10 10  
##   
## Within cluster sum of squares by cluster:  
## [1] 3.802496e+12 1.986660e+12 5.849888e+11 8.826774e+12 2.103887e+12  
## [6] 5.896407e+12 3.105378e+12 3.839290e+12 4.673033e+12 2.885953e+12  
##   
## Available components:  
##   
## [1] "cluster" "centers" "withinss" "size"

R3 <- bigkmeans(x, center = 3, iter.max = 100, nstart = 10)  
summary(R3)

## Length Class Mode   
## cluster 1017 -none- numeric  
## centers 18 -none- numeric  
## withinss 3 -none- numeric  
## size 3 -none- numeric

R3$cluster

## [1] 2 2 2 2 2 2 3 3 3 3 2 3 3 3 3 2 3 3 1 1 3 3 3 3 3 3 2 2 3 3 2 3 2 1 3 2 3  
## [38] 2 2 2 2 3 2 2 3 2 3 3 1 2 2 2 3 3 1 3 1 3 3 3 2 2 3 3 3 2 3 2 2 3 3 3 1 2  
## [75] 1 3 3 3 2 3 3 3 2 2 3 3 1 1 1 1 1 3 1 3 3 1 1 1 3 1 3 3 2 2 3 2 3 3 3 3 2  
## [112] 1 3 3 2 3 3 1 3 2 2 3 3 3 2 3 3 3 2 2 3 3 3 2 1 1 1 3 2 3 2 3 2 2 3 3 2 3  
## [149] 2 1 3 3 3 2 2 2 3 2 2 3 2 3 3 2 2 3 2 2 2 3 3 2 3 2 2 3 3 1 3 3 3 3 1 1 3  
## [186] 3 3 3 3 3 3 3 1 3 3 3 3 3 2 3 3 3 3 2 3 2 3 2 2 3 3 3 3 3 3 1 3 3 2 3 3 2  
## [223] 3 2 2 2 2 3 3 3 3 2 3 3 3 3 1 3 3 3 3 1 3 2 2 2 2 3 2 2 2 2 3 3 2 2 2 3 3  
## [260] 2 3 2 3 2 3 1 2 3 1 1 1 3 3 3 3 3 3 3 3 3 2 3 1 3 3 3 3 3 3 3 3 2 2 3 3 3  
## [297] 2 3 3 3 3 2 2 3 1 3 3 1 3 3 1 3 3 3 3 2 2 3 3 3 3 2 3 3 1 3 3 3 3 2 3 3 3  
## [334] 3 2 2 3 3 2 3 3 1 3 3 3 3 1 1 3 2 3 2 3 1 3 2 2 3 2 2 2 2 2 2 2 2 3 1 3 2  
## [371] 3 3 3 1 2 3 3 3 1 2 3 3 3 3 2 2 3 3 3 3 1 3 3 3 3 2 2 3 2 3 3 3 3 2 3 1 1  
## [408] 1 1 1 1 1 1 1 3 3 3 1 1 1 1 1 1 3 3 3 1 1 3 1 1 3 3 3 1 3 1 1 1 3 3 3 1 1  
## [445] 1 1 3 3 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 1 3 1 3 3 3 3 3 2 2 2 3  
## [482] 3 3 3 3 3 3 3 3 3 1 3 3 3 2 2 3 3 3 3 1 3 1 3 3 2 2 2 2 2 2 2 2 3 2 3 2 3  
## [519] 3 2 2 3 2 3 3 3 3 3 2 2 3 3 2 3 3 3 3 2 3 3 1 3 2 2 3 3 3 2 2 3 3 2 3 2 2  
## [556] 3 2 3 3 3 1 3 3 3 1 3 1 3 1 3 3 3 2 2 2 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 2  
## [593] 3 3 3 2 3 1 3 3 3 3 1 3 2 3 1 3 1 3 3 3 3 2 2 3 3 3 1 3 3 3 3 2 3 3 3 3 1  
## [630] 3 3 2 3 2 3 3 2 2 3 2 3 2 3 3 2 2 2 3 3 2 2 2 3 3 3 3 3 3 3 3 3 1 1 3 3 3  
## [667] 2 2 2 2 2 2 2 3 2 2 2 3 2 2 2 2 3 3 2 3 3 1 3 2 2 3 3 3 2 3 2 1 1 2 3 3 3  
## [704] 2 3 2 3 3 2 2 2 3 2 2 2 3 3 1 2 3 3 2 2 2 2 3 3 3 3 3 1 3 3 2 2 2 2 3 3 2  
## [741] 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 3 2 2 2 3 3 2 2 2 2 2 2  
## [778] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 2 2 2 2 2 2 2 2 2  
## [815] 2 2 3 3 2 3 2 3 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 3 2 3 2 3 3 2 2 2 2 3 2 2  
## [852] 2 2 2 3 2 3 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 1 3 3 2  
## [889] 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2  
## [926] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 3 3 2 2 2 2 2 2 2 2 2 2 2 2  
## [963] 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 1 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [1000] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2

R3$centers

## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 53.48832 54.67921 52.23121 53.52855 5243880 37.28641  
## [2,] 64.55363 65.11840 63.91027 64.58411 1623721 46.66433  
## [3,] 56.04169 56.87082 55.18087 56.04623 2871428 39.16213

colnames(x)

## [1] "BMW.DE.Open" "BMW.DE.High" "BMW.DE.Low" "BMW.DE.Close"   
## [5] "BMW.DE.Volume" "BMW.DE.Adjusted"

# read.csv.ffdf는 R 언어의 ff 패키지에서 제공되는 함수로, 대용량 CSV 파일을 메모리에 읽어들이기 위한 함수입니다. ff 패키지는 데이터를 파일로부터 읽거나 파일로 쓰는 등의 작업을 수행할 때 메모리를 효율적으로 활용할 수 있도록 도와주는 패키지입니다.  
# 여기서 read.csv.ffdf 함수는 ff 패키지에서 제공되는 대용량 데이터를 처리하기 위한 클래스인 ffdf 클래스의 객체로 데이터를 읽어들입니다.  
download.file("http://www.irs.gov/file\_source/pub/irs-soi/12zpallagi.csv", "soi.csv")  
x <- read.csv.ffdf(file="soi.csv", header = T)  
class(x)

## [1] "ffdf"

BigModel <- biglm(A02300~A00200+AGI\_STUB+NUMDEP+MARS2, data = x)  
summary(BigModel) #변수A02300: 국가에서 지불하는 실업수당당

## Large data regression model: biglm(A02300 ~ A00200 + AGI\_STUB + NUMDEP + MARS2, data = x)  
## Sample size = 166904   
## Coef (95% CI) SE p  
## (Intercept) 131.9412 44.3847 219.4977 43.7782 0.0026  
## A00200 -0.0019 -0.0019 -0.0018 0.0000 0.0000  
## AGI\_STUB -40.1597 -62.6401 -17.6794 11.2402 0.0004  
## NUMDEP 0.9270 0.9235 0.9306 0.0018 0.0000  
## MARS2 -0.1451 -0.1574 -0.1327 0.0062 0.0000

summary(BigModel)$rsq #대용량 데이터에 대한 선형 회귀 모델(biglm 객체)의 결정 계수 (R-squared) 값

## [1] 0.8609021