R語言學習整理 2017/06/25 - 2017/07/18

[1. 環境設定 1](#_Toc488156312)

[2. 語言基礎 2](#_Toc488156313)

[2.1. 基礎指令 2](#_Toc488156314)

[2.2. 數學運算 3](#_Toc488156315)

[2.3. 設定變數 4](#_Toc488156316)

[2.4. 基礎資料型態 4](#_Toc488156317)

[2.5. 向量(Vector) 4](#_Toc488156318)

[2.6. 陣列(Matrix) 7](#_Toc488156319)

[2.7. 階層(Factor) 10](#_Toc488156320)

[2.8. 資料框(Data Frame) 11](#_Toc488156321)

[2.9. 清單(Lists) 14](#_Toc488156322)

[2.10. 流程控制(Flow Control) 16](#_Toc488156323)

[2.11. 函式(Function) 18](#_Toc488156324)

[2.12. Windows執行R檔案 19](#_Toc488156325)

[3. 資料收集 21](#_Toc488156326)

[4. 資料儲存 21](#_Toc488156327)

[4.1. 連接資料庫 21](#_Toc488156328)

[4.1.1. MySQL 24](#_Toc488156329)

[4.1.2. SQL Server 25](#_Toc488156330)

[5. 資料視覺化 26](#_Toc488156331)

[5.1. Quantmod繪製股票技術分析圖表 26](#_Toc488156332)

[6. 資料探索 28](#_Toc488156333)

[6.1. 分組之描述性統計 28](#_Toc488156334)

[6.2. 線性迴歸分析 29](#_Toc488156335)

[7. 機械學習 31](#_Toc488156336)

[8. 其他 31](#_Toc488156337)

丘祐瑋 數據分析師假日精修班 程式碼

<https://github.com/ywchiu/rtibame>

# 環境設定

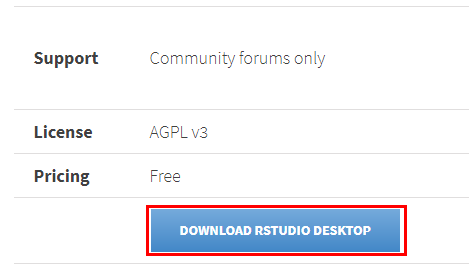
1.下載R

<https://cloud.r-project.org/>



2.下載R Studio

<https://www.rstudio.com/products/rstudio/>



# 語言基礎

## 基礎指令

#以下網站有許多R的整合範例與新資訊，建議FB追蹤

#https://www.r-bloggers.com/

#查詢函式用法

help(seq)

?seq

#檢查資料型態

numer <- 3.14159

class(numer)

#檢查架構

str(df)

#檢查摘要

y <- seq(1,20)

summary(y)

#檢視資料

View(iris)

## 數學運算

#數字相加

a<-2

b<-3

a+b

#數字相減

a<-3

b<-2

a-b

#數字相乘

a<-3

b<-2

a\*b

#數字相除

a<-3

b<-2

a/b

#指數

a<-3

b<-2

a^b

#取餘數

a<-3

b<-2

a%%b

## 設定變數

#可以使用=或<-指定變數

a<-3

b=2

#若是變數數字開頭使用`符號

`2330.TW`

## 基礎資料型態

numer <- 3.14159

char = "hello world"

logic <- TRUE

#使用class檢查資料型態

class(numer)

class(char)

class(logic)

## 向量(Vector)

#數學運算

a <- c(100,100)

b <- c(2,3)

a + 55

a + (66)

a + b

a - b

a \* b

a / b

a ^ b

a %% b

#組合

x <- c(85,73)

y <- c(72,64)

z <- c(59,66)

c(x, y, z)

#循序產生Vector

y <- seq(from=1,to=20)

y

y <- seq(from=1,to=20,by 2)

y

#簡化寫法

y <- seq(1,20)

y

y <- seq(1,20,2)

y

#加總

sum(y)

#平均數

mean(y)

#摘要

summary(y)

#處理遺失值

y<- c(11,22,33,NA)

y

sum(y)

?sum

sum(y , na.rm=TRUE)

#取出指定位置的值

y[1]

y[2]

y[3]

y[4]

#設定名稱

Name\_vec <- c("Name1","Name2","Name3","Name4")

names(y) <- Name\_vec

y

names(y)

#條件過濾

y <- c(11,22,33)

#產生真假表

y > 11

#以真假表進行過濾

z <- y[y > 11]

z

#複合條件過濾

z <- y[y > 15 | y < 30]

z

z <- y[y > 15 & y < 30]

z

#應用範例-計算BMI值

height\_vec <- c(180, 169, 173)

weight\_vec <- c(73, 87, 43)

bmi\_vec <- weight\_vec / (height\_vec / 100)^2

bmi\_vec

name\_vec <- c('Brian', 'Toby', 'Sherry')

names(bmi\_vec) <- name\_vec

bmi\_vec

abnormal <- bmi\_vec[bmi\_vec < 18.5 | bmi\_vec > 25]

abnormal

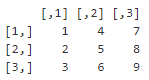
normal <- bmi\_vec[bmi\_vec >= 18.5 & bmi\_vec < 25]

normal

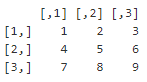
## 陣列(Matrix)

#補充，Matrix與Data Frame差別在於，Matrix內容僅有一種資料型態，Data Frame則可包含全部資料型態

matrix(1:9, nrow = 3)



matrix(1:9, nrow = 3, byrow = TRUE)



#將向量轉為陣列，並設定名稱

kevin <- c(85,73)

marry <- c(72,64)

jerry <- c(59,66)

#方式1

c(kevin, marry, jerry)

mat <- matrix(c(kevin, marry, jerry), nrow=3, byrow=TRUE)

colnames(mat) <- c('first', 'second')

rownames(mat) <- c('kevin', 'marry', 'jerry')

mat

#方式2

mat2 <- matrix(

c(kevin, marry, jerry),

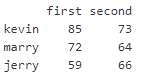
nrow=3,

byrow=TRUE,

dimnames = list(c('kevin', 'marry', 'jerry'), c('first', 'second'))

)

mat2



#取得陣列Row與Column數量

dim(mat2)



#取得陣列Row數量

nrow(mat2)



#取得陣列Column數量

ncol(mat2)



#取得第1列

mat2[ 1 , ]



#取得第1欄

mat2[ , 1]



#取得第2列，第1欄

mat2[ 2 , 1]



#取得第2~3列

mat2[ 2:3 , ]

mat2[c(2,3), ]

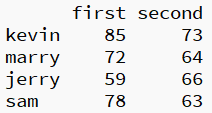


#增加1列，並給予新的名稱

mat3 <- rbind(mat2, c(78 , 63) )

rownames(mat3)[4] <- 'sam'

mat3

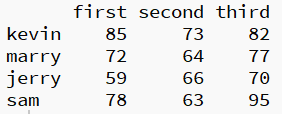


#增加1欄，並給予新的名稱

mat4 <- cbind(mat3, c(82, 77, 70 ,95))

colnames(mat4)[3] <- 'third'

mat4

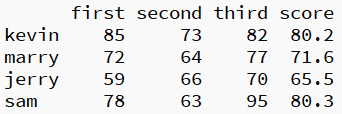


#計算最後成績 (方式1)

score <- mat4[ , 1 ] \* 0.3 + mat4[ , 2] \* 0.3 + mat4[ , 3] \* 0.4

mat5 <- cbind(mat4, score)

mat5



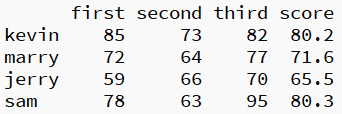
#使用矩陣乘積計算最後成績 (方式2)

socre\_rate <- c(0.3,0.3,0.4)

mat6 <- cbind(mat4, mat4 %\*% socre\_rate)

colnames(mat6)[4] <- 'score'

mat6

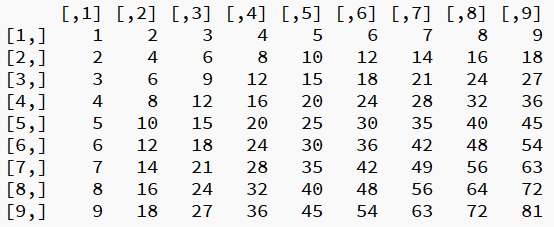


#以矩陣乘積製作99乘法表

m1 <- matrix(1:9)

m2 <- matrix(1:9, nrow=1)

m1 %\*% m2



#以矩陣乘積計算99乘法總和

m1 <- matrix(1:9, nrow=1)

m2 <- matrix(1:81, nrow=9)

#這樣寫也可以 m2 <- matrix(1:81)

m3 <- m1 %\*% m2

sum(m3)



## 階層(Factor)

#使用Factor產生類別資料

#建立向量Vector

Weather<- c('sunny', 'rainny', 'cloudy', 'rainny', 'cloudy')

class(Weather)



#將向量透過Factor轉為類別資料

weather\_category <- factor(Weather)

weather\_category

class(weather\_category)



#檢視Factor內容

levels(weather\_category)



#階層比較，以下會發生錯誤，因為尚未設定Levels順序

weather\_category[1] > weather\_category[2]



#兼立一個新的Factor，並設定Levels順序

temperature <- c("Low", "High", "High", "Medium", "Low", "Medium")

temperature\_category <- factor(temperature, order = TRUE, levels = c("Low", "Medium", "High"))

temperature\_category



#檢視Factor內容

levels(temperature\_category)



#進行Factor的Levels順序比較

temperature\_category[3] > temperature\_category[1]



temperature\_category[4] > temperature\_category[3]



## 資料框(Data Frame)

#補充，Matrix與Data Frame差別在於，Matrix內容僅有一種資料型態，Data Frame則可包含全部資料型態

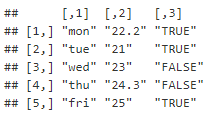
days <- c('mon','tue','wed','thu','fri')

temp <- c(22.2,21,23,24.3,25)

rain <- c(TRUE, TRUE, FALSE, FALSE, TRUE)

# Matrix內容僅有一種資料型態

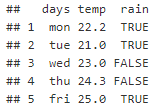
matrix(c(days, temp, rain), nrow = 5)



# Data Frame則可包含全部資料型態

df <- data.frame(days,temp,rain)

df

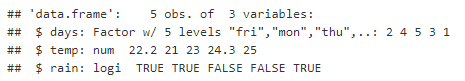


class(df)



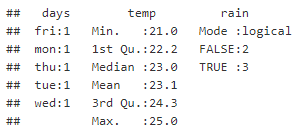
#檢查架構

str(df)



#檢視資料摘要

summary(df)

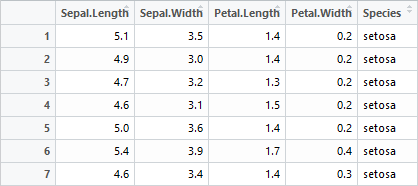


#data調用內建資料集

#將iris資料集載入Global Environment成為一個DataFrame

data(iris)

View(iris)

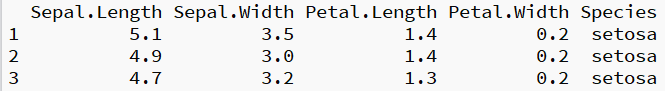


#前幾筆資料

head(iris)

head(iris,3)

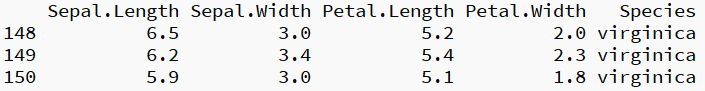
head(n = 3, x = iris)



#後幾筆資料

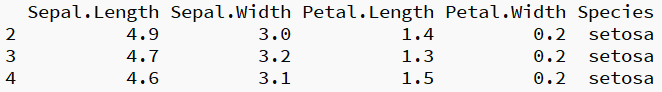
tail(iris)

tail(iris,3)



#取指定範圍列向量資料

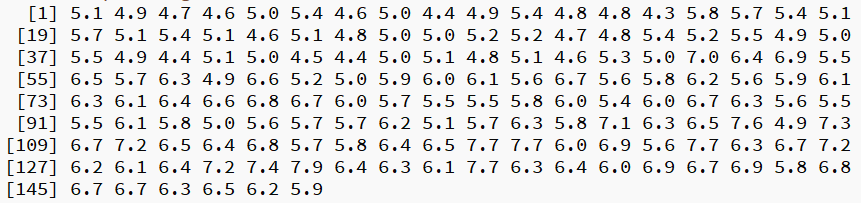
iris[2:4 , ]



#取指定範圍欄向量資料

iris[ ,1 ]

iris$Sepal.Length



#取指定範圍列與欄向量資料

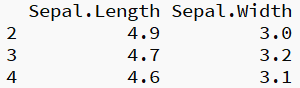
iris[2:4, 1]

iris[2:4, 'Sepal.Length']



iris[2:4, 1:2]

iris[2:4, c('Sepal.Length', 'Sepal.Width') ]



#用Sort做資料排序

a <- c(2,5,1,3,6,4)

sort(a)



head(sort(iris$Sepal.Length))



#用Order 取得排序後的索引

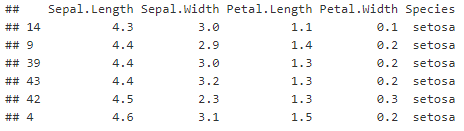
a <- c(2,5,1,3,6,4)

order(a)



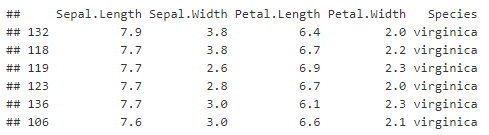
#Order順排序

head(iris[order(iris$Sepal.Length), ])



#Order逆排序

head(iris[order(iris$Sepal.Length, decreasing = TRUE), ])



## 清單(Lists)

#清單的好處，在資料處理時，我們可以將不同資料結構的資料放在一起

item <- list(thing='hat', size=8.25)

item

class(item)



class(item$thing)

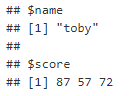


class(item$size)



test <- list(name='toby', score= c(87, 57, 72))

test



test$score

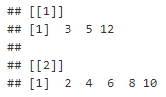


test$score[2]



li <- list(c(3,5,12), c(2,4,6,8,10))

li



li[[1]]



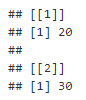
#lappy可以將個別元素拿出來之後，套用函數去做處理

#針對各資料需要做一致性操作時，適合用lappy函數

#套用Sum函數

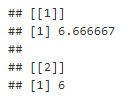
lapply(li, sum)

lapply(li, function(e) sum(e) )



#套用mean函數

lapply(li, function(e) mean(e) )



#將List中 ，將各元素的第1個資料取出

lapply(li, function(e) e[1] )



## 流程控制(Flow Control)

#IF ELSE 控制

x <- 0

if (x < 0) {

print("Negative number")

} else if (x > 0) {

print("Positive number")

} else

print("Zero")



#WHILE 迴圈 (須注意不滿足條件，否則會形成無窮迴圈)

s <- 0

cnt <- 0

while(cnt <= 100){

s <- s + cnt

cnt <- cnt + 1

}

s



#FOR 迴圈

s <- 0

for(i in 1:100){

s <- s + i

}

S



#若有函數可以使用，盡量不使用迴圈，教能較好

sum(1:100)



#依序取出Vector元素 (方式1)

x <- c("sunny","rainy", "cloudy", "rainy", "cloudy")

length(x)

for(i in 1:length(x)) {

print(x[i])

}

#依序取出Vector元素 (方式2)

for(i in seq\_along(x)){

print(x[i])

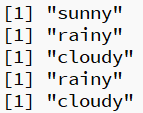
}

#依序取出Vector元素 (方式3)

for(i in x){

print(i)

}



#迴圈實例應用：產生多筆網頁超連結

#Paste字串合併

?paste

paste('hello', 'world')



paste('hello', 'world', sep ='')

paste0('hello', 'world')

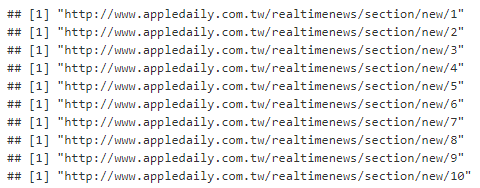


url <- 'http://www.appledaily.com.tw/realtimenews/section/new/'

for(i in 1:10){

print(paste0(url, i))

}



## 函式(Function)

f <- function(a){

a + 3

}

f(5)



#預設值作法

f2 <- function(a, b = 2){

a + b

}

f2(3,5)

f2(3)



#應用範例：文章詞頻計算

#建立詞頻統計函式後，儲存在D:/DataScience/R/Code/Function/wordcount.R

wordcount <- function(article){

#strsplit可根據某個符號，進行切割，結果為Lists型態

words <- strsplit(article , ' ')

# table (x)， x 為 factor、list、data frame，彙整分類組合發生次數

tb <- table (words)

#將前10名資料列出

head(tb[order(tb, decreasing = TRUE)], 10)

}

#引用文章詞頻計算函式

source("D:/DataScience/R/Code/Function/wordcount.R")

#設定文章內容

str\_content <- 'Xi Jinping to visit Hong Kong for 20th anniversary of handover By James Griffiths, CNN Updated 0739 GMT (1539 HKT) June 25, 2017 Chinese President Xi Jinping will visit Hong Kong for the first time as leader to mark the 20th anniversary of the city 39;s handover to China on July 1, 2017.

Chinese President Xi Jinping will visit Hong Kong for the first time as leader to mark the 20th anniversary of the city s handover to China on July 1, 2017.

Story highlights

Chinese President Xi Jinping to visit Hong Kong for the first time as Chinese leader

His visit is expected to be met with massive protests

Hong Kong (CNN)Chinese President Xi Jinping will visit Hong Kong for the 20th anniversary of the city s handover to Chinese sovereignty, according to Chinese state news agency Xinhua.'

#呼叫函式統計結果

wordcount(str\_content)

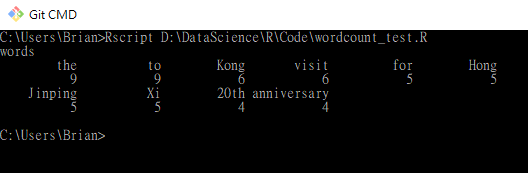
## Windows執行R檔案

#參考資料：把R變成執行檔

#https://www.youtube.com/watch?v=A1IIcZW5UrI

#在系統環境變數，在path加入C:\Program Files\R\R-3.4.1\bin，可用以下指令

Rscript D:\DataScience\R\Code\wordcount\_test.R



#wordcount\_test.R

#引用文章詞頻計算函式

source("D:/DataScience/R/Code/Function/wordcount.R")

#設定文章內容

str\_content <- 'Xi Jinping to visit Hong Kong for 20th anniversary of handover By James Griffiths, CNN Updated 0739 GMT (1539 HKT) June 25, 2017 Chinese President Xi Jinping will visit Hong Kong for the first time as leader to mark the 20th anniversary of the city 39;s handover to China on July 1, 2017.

Chinese President Xi Jinping will visit Hong Kong for the first time as leader to mark the 20th anniversary of the city s handover to China on July 1, 2017.

Story highlights

Chinese President Xi Jinping to visit Hong Kong for the first time as Chinese leader

His visit is expected to be met with massive protests

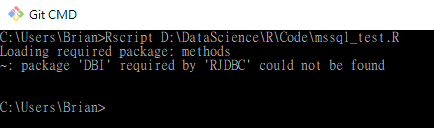
Hong Kong (CNN)Chinese President Xi Jinping will visit Hong Kong for the 20th anniversary of the city s handover to Chinese sovereignty, according to Chinese state news agency Xinhua.'

#呼叫函式統計結果

wordcount(str\_content)

#目前測試資料庫寫入有錯誤，再請教老師

Rscript D:\DataScience\R\Code\mssql\_test.R



#sqltest.R

library(RJDBC , lib.loc = "C:/Users/Brian/Documents/R/win-library/3.4/")

library(DBI , lib.loc = "C:/Users/Brian/Documents/R/win-library/3.4/")

if (interactive()) {

library(RJDBC , lib.loc = "C:/Users/Brian/Documents/R/win-library/3.4/")

library(DBI , lib.loc = "C:/Users/Brian/Documents/R/win-library/3.4/")

}

drv <- JDBC("com.microsoft.sqlserver.jdbc.SQLServerDriver",

"C:\\Program Files\\Microsoft SQL Server\\sqljdbc42.jar",

identifier.quote= "\"")

conn <- dbConnect(drv,

"jdbc:sqlserver://localhost:1433;databaseName=R",

"sa",

"lucky")

dbWriteTable(conn, "iris", iris)

df <- dbGetQuery(conn, "SELECT \* FROM iris WHERE Species = 'setosa' ")

df

dbDisconnect(conn)

# 資料收集

# 資料儲存

## 連接資料庫

使用R連接資料庫的方式

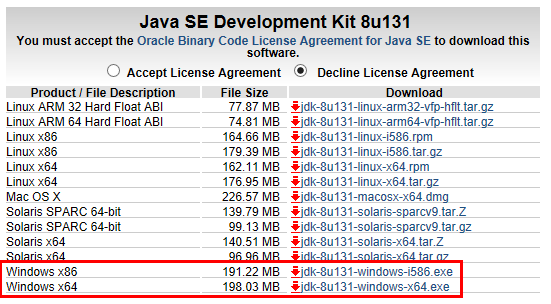
1.RMySQL

2.RODBC

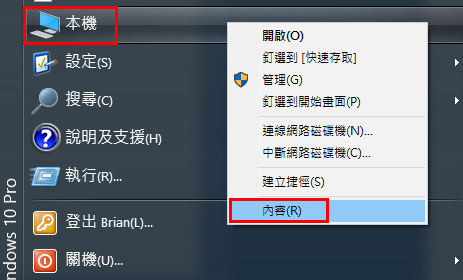
3.RJDBC (老師推薦使用，以下介紹RJDBC使用方式)

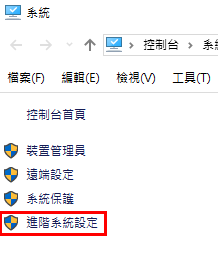
1.下載JDK，Win10安裝WindowsX64版本

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

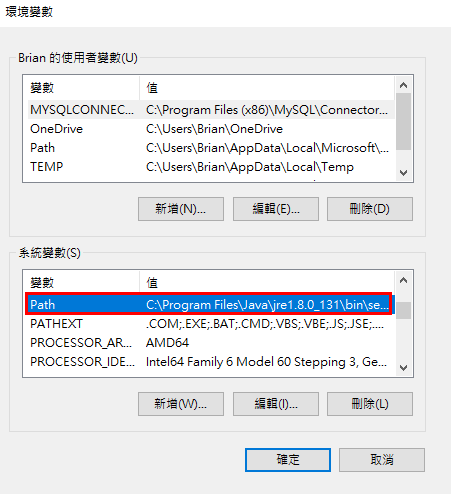


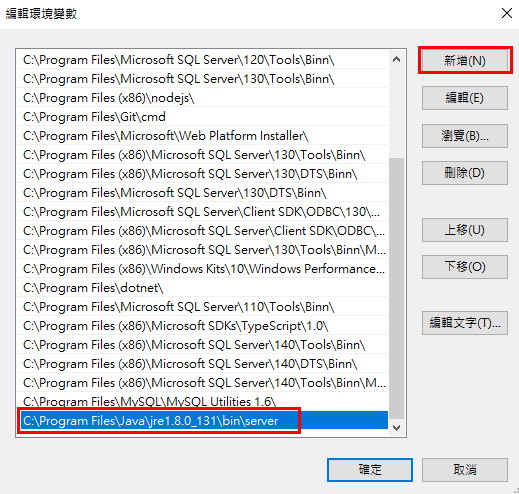
2.將jvm.dll的路徑加到系統變數的Path後，再安裝RJDBC套件







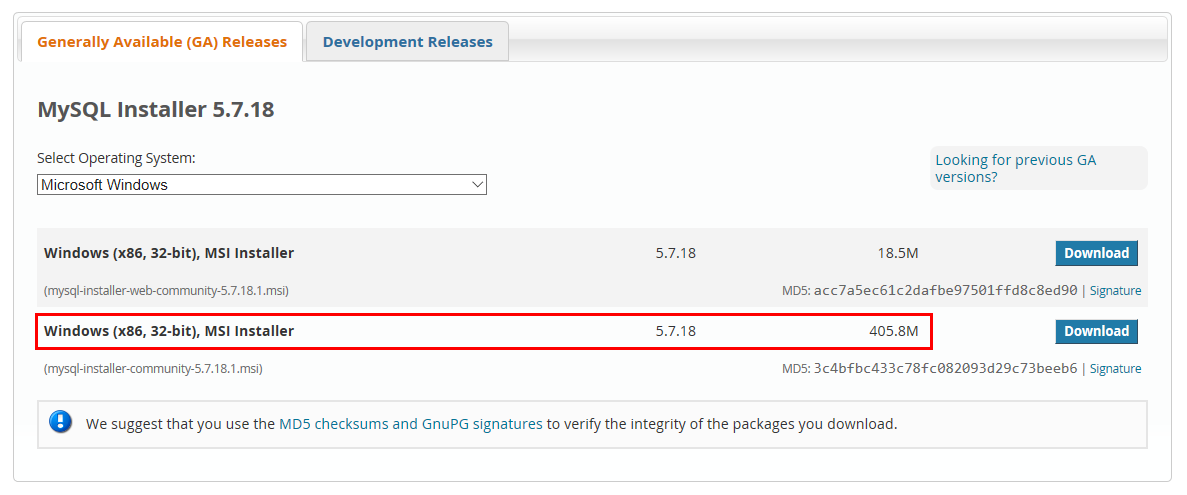




### MySQL

1.下載MySQL，預設值安裝

<https://dev.mysql.com/downloads/windows/installer/5.7.html>



2.新增R資料庫，進行連線測試，將iris寫入資料庫後，再讀取出來，最後關閉資料庫連線

(mysql-connector-java-5.1.41-bin.jar為MySQL的JDBC的JAR檔)

library(RJDBC)

drv <- JDBC("com.mysql.jdbc.Driver",

"C:\\Program Files (x86)\\MySQL\\Connector.J 5.1\\mysql-connector-java-5.1.41-bin.jar",

identifier.quote="`")

conn <- dbConnect(drv,

"jdbc:mysql://localhost/r",

"root",

"lucky")

dbWriteTable(conn, "iris", iris)

df <- dbGetQuery(conn, "SELECT \* FROM iris WHERE Species = 'setosa' ")

df

dbDisconnect(conn)

### SQL Server

1.下載Microsoft JDBC Drivers 4.1 and 4.0 for SQL Server

解壓縮後，將jre8下的sqljdbc42.jar，複製到C:\Program Files\Microsoft SQL Server

<https://www.microsoft.com/zh-tw/download/details.aspx?id=11774>



2.開啟SQL Server Configuration Manager，將TCP/IP的通訊協定設為已啟用

程式位置： C:\Windows\SysWOW64\SQLServerManager13.msc



4. 新增R資料庫，進行連線測試，將iris寫入資料庫後，再讀取出來，最後關閉資料庫連線

library(RJDBC)

drv <- JDBC("com.microsoft.sqlserver.jdbc.SQLServerDriver",

"C:\\Program Files\\Microsoft SQL Server\\sqljdbc42.jar",

identifier.quote= "\"")

conn <- dbConnect(drv,

"jdbc:sqlserver://localhost:1433;databaseName=R",

"sa",

"lucky")

dbWriteTable(conn, "iris", iris)

df <- dbGetQuery(conn, "SELECT \* FROM iris WHERE Species = 'setosa' ")

df

dbDisconnect(conn)

# 資料視覺化

## Quantmod繪製股票技術分析圖表

#install.packages("quantmod")

library(quantmod)

#若要將台積電股票數據儲存在變數 tw2330 上，需要加 auto.assign=FALSE，才能成功儲存在變數 tw2330 上

#經測試目前yahoo的台股資料部分有遺失

tw2330 <-getSymbols("2330.TW",from = "2017-01-01",to = "2017-12-31",src = "yahoo",auto.assign=FALSE)

chartSeries(tw2330)

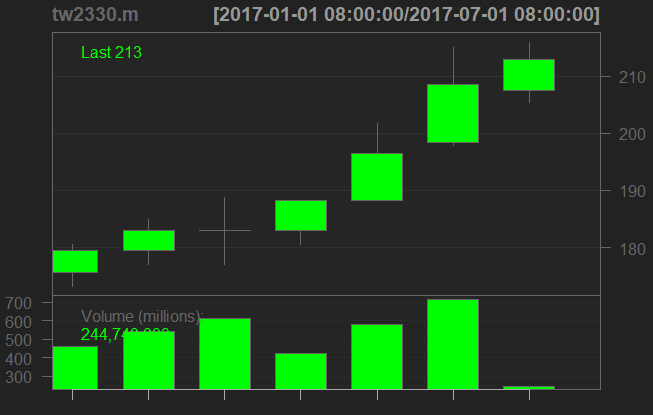


#處理為月週期

tw2330.m<-to.monthly(tw2330)

tail(tw2330.m)

chartSeries(tw2330.m)



#取得最近10年的股價資料，並加上技術指標

tw2330.10y <-getSymbols("2330.TW", from = Sys.Date()-3650, to = Sys.Date(), src = "yahoo" , auto.assign=FALSE)

#subset='2014::2017-12’ 股價走勢圖的時間起始點設定在 2014 年~2017-12

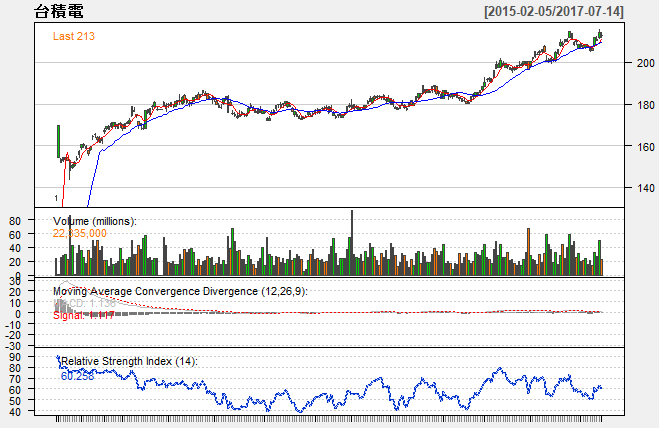
chartSeries(tw2330.10y,subset='2014::2017-12',theme="white",name="台積電")

addSMA(n = 5, col = "red")

addSMA(n = 20, col = "blue")

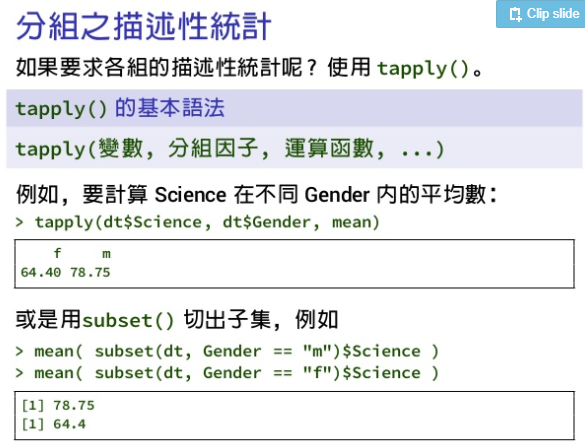
addMACD()

addRSI()



# 資料探索

## 分組之描述性統計



View(iris)

#tapply取得各組平均值

tapply(iris$Sepal.Length, iris$Species, mean)



#tapply取得各組最小值

tapply(iris$Sepal.Length, iris$Species, min)



#tapply取得各組最大值

tapply(iris$Sepal.Length, iris$Species, max)



#subset取得各組平均值

mean( subset(iris , Species== "setosa")$Sepal.Length)



mean ( subset(iris , Species== "versicolor")$Sepal.Length)



mean ( subset(iris , Species== "virginica")$Sepal.Length)



## 線性迴歸分析

#若要看目前可以使用的資料集有那些，可以使用data()這個函式

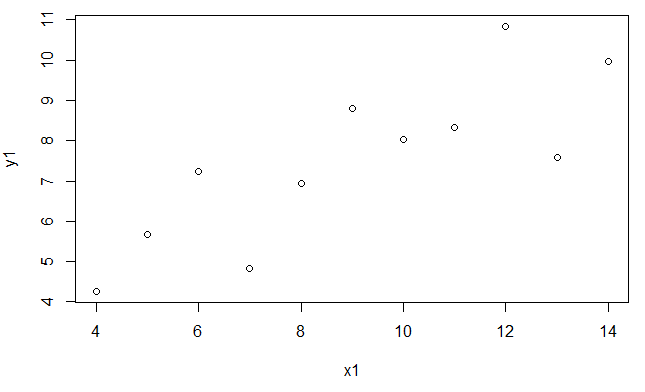
#anscombe內有4組統計數據可用

data("anscombe")

View(anscombe)

#y1對x1進行繪圖

plot(y1~x1,data=anscombe)



#若希望透過數據模型，分析x1與y1的關係

#運用迴歸分析lm函數，使用最小平方法，找到讓所有點對這條線最適配的模型

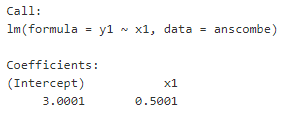
#最小平方法：讓所有點對這條線的距離的Y的平方總和是最小的

#fit是透過lm訓練出來的模型

fit <- lm(y1~x1,data=anscombe)

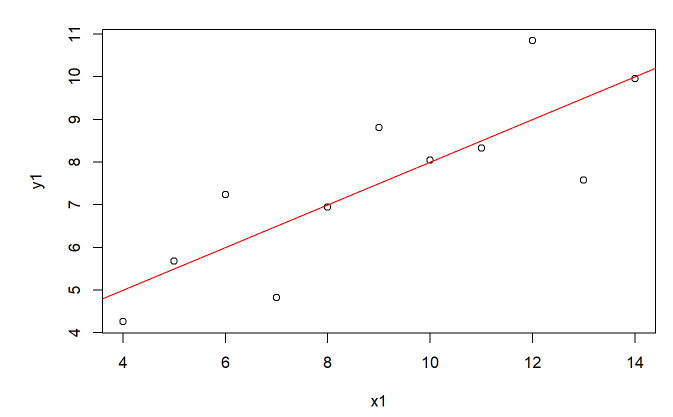
fit

#結果為x1增加1，y1增加0.5001



#將lm模型的結果放到圖上，完成分析

abline(fit,col="red")



#台北市實價登錄分析

library(readr)

lvr <- read\_csv("D:/DataScience/R/Data/lvr\_prices.csv")

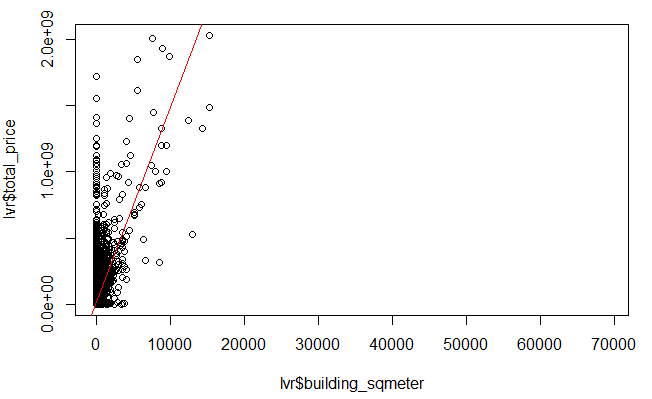
#View(lvr\_prices)

#觀察坪數與價格的關係

plot(lvr$total\_price ~ lvr$building\_sqmeter , data = lvr)

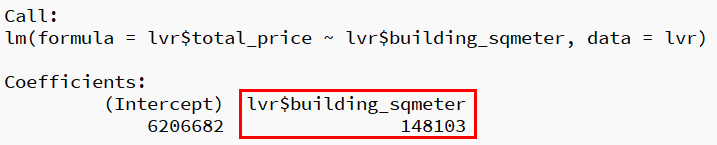
fit <- lm(lvr$total\_price ~ lvr$building\_sqmeter , data = lvr)

abline(fit,col="red")



fit

#每增加1平方公尺，價格增加148103



#坪數換算，每增加1坪，價格增加489596

148103 / 0.3025



#使用log讓資料更容易表現其特性

library(readr)

lvr <- read\_csv("D:/DataScience/R/Data/lvr\_prices.csv")

#取Log觀察坪數與價格的關係

plot(log(lvr$total\_price) ~ log(lvr$building\_sqmeter) , data = lvr)



log補充說明

資料是否要取對數或其他變換，取決於一些因素

例如

1. 在例如迴歸模型中可能需要某種變數變換以方便用線性式表現模型

否則模型可能較複雜或難以解釋

2. 模型之依變數或單一變數之分析， 需要做變數變換以符合或

近似符合某些分析上所要求的條件，例如常態性條件

3. 長尾型偏斜分布的資料，如取對數或平方根轉換，可更好地

表現分布特性或其他目的

4. 因資料之分布特性，在做某些分析時自然導出需要或最好是

做某些變數變換

# 機械學習

# 其他