**國企三 陳湘雲 B08704024**

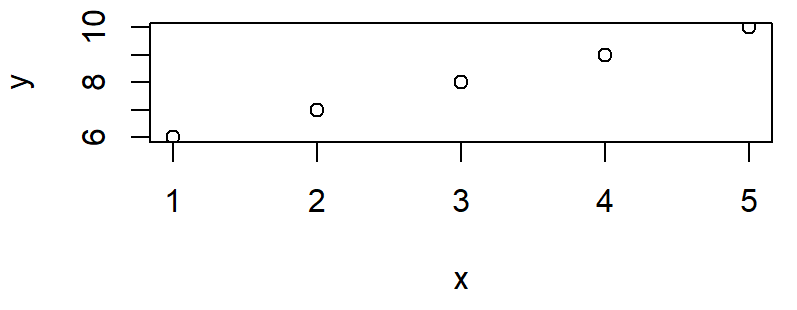
**Week 1**

**Video1.1-1.3**

> x <- 1:5

> y <- 6:10

> plot(x,y)



R Studio 的好處

1. 可顯示結果在同頁面，如上面那個圖
2. 圖可存各種檔案類型並調整點距
3. 能匯入其他已完成檔案
4. 可寫草稿再丟進程式並儲存

> ls()

[1] "x" "y"

> rm(y)-------移除

> ls()-----------看已建立的未知數

[1] "x"

> z=x+3

> z

[1] 4 5 6 7 8

> r=9

> sqrt(r)-------平方根

[1] 3

> exp(r)----------er

[1] 8103.084

> getwd()-------------存檔的位置

[1] "C:/Users/w4566/Desktop/台大/大三/數量方法與分析決策"

save

**week 2**

**Video**

> x1=c(1,3,5,7,9)

> x1

[1] 1 3 5 7 9

> gender=c("male","female")

> gender

[1] "male" "female"

> 2:7

[1] 2 3 4 5 6 7

> seq(from=1,to=7,by=2)

[1] 1 3 5 7

> seq(1,7,2)

[1] 1 3 5 7

**Week3**

**video**

> aa <- data.frame(nickname=c("John","Mary","Leo"), weight=60:62, Height=c(160,170,180))

> mean(aa$weight)-----------$表示你要表裡面的什麼 mean就是平均

[1] 61

attach(aa)--------只是舉例，如果有從外匯進來的表格，要讓R認識裡面的資訊就用attach

detach(aa)----------刪掉匯進來的

> aa

nickname weight Height

1 John 60 160

2 Mary 61 170

3 Leo 62 180

> summary(aa)

nickname weight Height

Length:3 Min. :60.0 Min. :160

Class :character 1st Qu.:60.5 1st Qu.:165

Mode :character Median :61.0 Median :170

Mean :61.0 Mean :170

3rd Qu.:61.5 3rd Qu.:175

Max. :62.0 Max. :180

> x=c(1,1,1,0,0,0,2,1)

> x

[1] 1 1 1 0 0 0 2 1

> class(x)

[1] "numeric"

> x=as.factor(x)

> summary(x)

0 1 2

3 4 1

mean(weight[gender==male])---------- 如果我有輸入性別分類的話

> aa[weight>61]

Height

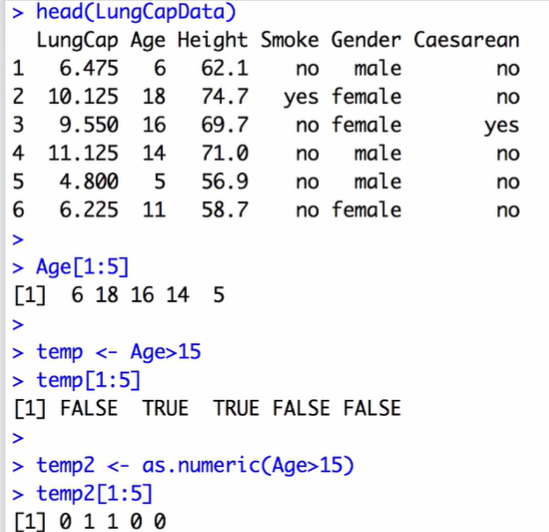
1 160

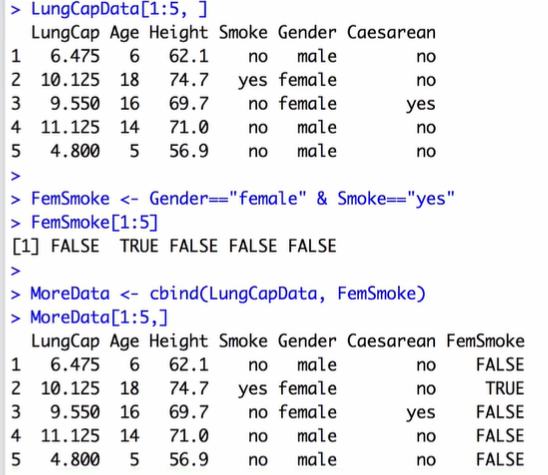
2 170

3 180

**Week 4**

**Video** 1.10-1.12

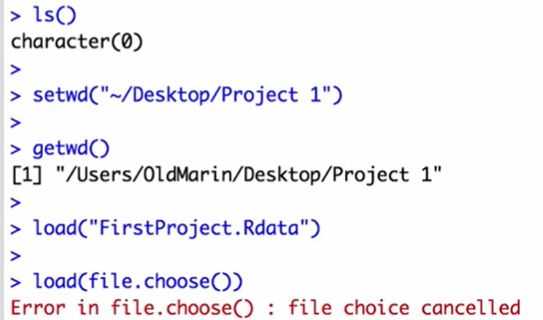




> rm(list=ls())-------------environment被清空

(開了一個空白R studio然後下載剛剛的environment)



**Class**

Youtube影片要找五個指令 要露臉

Base:print (base套件中的print指令)

> aaa <- c(1,8,10)

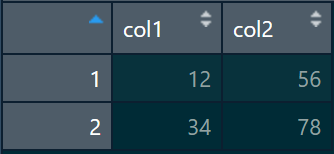
> bbb <- "try"

> ccc <- data.frame(col1=c(12,34),col2=c(56,78))

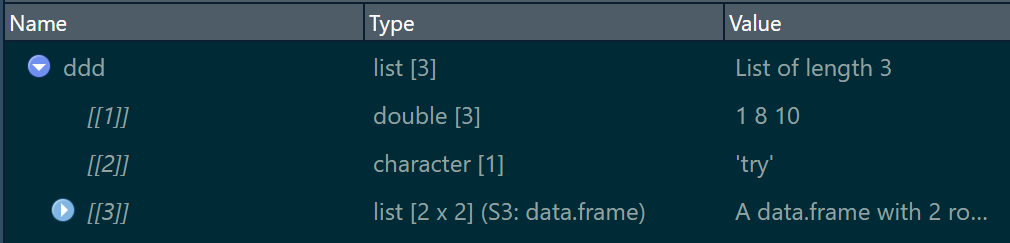
> ddd <- list(aaa,bbb,ccc)

> eee <- list(ddd,aaa)

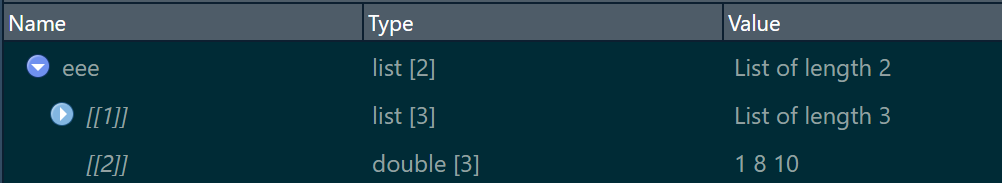
> View(ccc)



> View(ddd)



> View(eee)



> install.packages('stringr')………..下載套件

> library(stringr)

> kk <- "abcdefgheijk"

> pp <- str\_split(kk, "e")

> pp

[[1]]

[1] "abcd" "fgh" "ijk"

> class(pp[[1]])

[1] "character"

> pp <- str\_split(kk,c("d","h"))

> pp

[[1]]

[1] "abc" "efgheijk"

[[2]]

[1] "abcdefg" "eijk"

> pp <- str\_split(kk, "de")

> pp

[[1]]

[1] "abc" "fgheijk"

> jj <- str\_split(kk,"")

> jj

[[1]]

[1] "a" "b" "c" "d" "e" "f" "g" "h" "e" "i" "j" "k"

> kk <- "John Wang"

> yy <- str\_split(kk, " ")

> yy

[[1]]

[1] "John" "Wang"

> zz <- paste0("Mary ", yy[[1]][2])

> zz

[1] "Mary Wang"

> kk <- c("John Wang", "Tom Cruise", "Kelly Chen")

> yy <- str\_split(kk, " ")

> yy

[[1]]

[1] "John" "Wang"

[[2]]

[1] "Tom" "Cruise"

[[3]]

[1] "Kelly" "Chen"

> library(pdftools)

> pdfText1 = pdf\_text(paste0(getwd(),"/05160218039.004.pdf"))

> Text1=pdfText1[1]

> pos1=str\_locate(Text1,"由投標人")

> pos1

start end

[1,] NA NA

> pos3=str\_locate(text1,"主旨")

> wantedString=str\_sub(text1,(pos1[1,2]+2),(pos3[1,1]-3))

> wantedString

[1] NA

> allFiles <- list.files(getwd(), full.names=F)

> allFiles

character(0)

> file.rename("05160218039.004.pdf","上課用檔案.pdf")

[1] TRUE

**Week 5**

**Video** 1.13-1.15

install.packages("epiR")

library(epiR)

help(packages=epiR)

rm.packages(epiR)

**Week 6**

**Video** 1.16-2.1

> getwd()

[1] "C:/Users/w4566/Desktop/台大/大三/數量方法與分析決策"

> StockData <- read.table(file="C:/Users/w4566/Desktop/台大/大三/數量方法與分析決策/StockExample.csv",sep=",", header=T, row.names=1)

> View(StockData)

> View(StockData)

> StockData

Stock1 Stock2 Stock3 Stock4

Day1 185.74 1.47 1605 95.05

Day2 184.26 1.56 1580 97.49

Day3 162.21 1.39 1490 88.57

Day4 159.04 1.43 1520 85.55

Day5 164.87 1.42 1550 92.04

Day6 162.72 1.36 1525 91.70

Day7 157.89 NA 1495 89.88

Day8 159.49 1.43 1485 93.17

Day9 150.22 1.57 1470 90.12

Day10 151.02 1.54 1510 92.14

> apply(X=StockData, MARGIN=2, FUN=mean)

Stock1 Stock2 Stock3 Stock4

163.746 NA 1523.000 91.571

> apply(X=StockData, MARGIN=2, FUN=mean, na.rm=TRUE)

Stock1 Stock2 Stock3 Stock4

163.746000 1.463333 1523.000000 91.571000

> AVG <- apply(X=StockData, MARGIN=2, FUN=mean, na.rm=TRUE)

> AVG

Stock1 Stock2 Stock3 Stock4

163.746000 1.463333 1523.000000 91.571000

> apply(StockData, 2, mean, na.rm=TRUE)

Stock1 Stock2 Stock3 Stock4

163.746000 1.463333 1523.000000 91.571000

> colMeans(StockData, na.rm=TRUE)

Stock1 Stock2 Stock3 Stock4

163.746000 1.463333 1523.000000 91.571000

> # notice that we don't need to include "MARGIN", etc, as long

> # as we enter info in the specified order

> apply(StockData, 2, mean, na.rm=TRUE)

Stock1 Stock2 Stock3 Stock4

163.746000 1.463333 1523.000000 91.571000

> colMeans(StockData, na.rm=TRUE)

Stock1 Stock2 Stock3 Stock4

163.746000 1.463333 1523.000000 91.571000

> apply(X=StockData, MARGIN=2, FUN=max, na.rm=TRUE)

Stock1 Stock2 Stock3 Stock4

185.74 1.57 1605.00 97.49

> apply(X=StockData, MARGIN=2, FUN=quantile, probs=c(0.2, .80),na.rm=TRUE)

Stock1 Stock2 Stock3 Stock4

20% 156.516 1.408 1489 89.618

80% 168.748 1.548 1556 93.546

> apply(X=StockData, MARGIN=2, FUN=plot, type="l")

NULL

> apply(X=StockData, MARGIN=2, FUN=plot, type="l", main="stock", ylab="Price", xlab="Day")

NULL

> apply(X=StockData, MARGIN=1, FUN=sum, na.rm=TRUE)

Day1 Day2 Day3 Day4 Day5 Day6 Day7

1887.26 1863.31 1742.17 1766.02 1808.33 1780.78 1742.77

Day8 Day9 Day10

1739.09 1711.91 1754.70

> rowSums(StockData, na.rm=TRUE)

Day1 Day2 Day3 Day4 Day5 Day6 Day7

1887.26 1863.31 1742.17 1766.02 1808.33 1780.78 1742.77

Day8 Day9 Day10

1739.09 1711.91 1754.70

> plot(apply(X=StockData, MARGIN=1, FUN=sum, na.rm=TRUE), type="l",ylab="Total Market Value", xlab="Day", main="Market Trend")

> X=StockData, MARGIN=1, FUN=sum, na.rm=TRUE),

Error: unexpected ',' in "X=StockData,"

> pch=16, col="blue")

Error: unexpected ',' in " pch=16,"

> points(apply(X=StockData, MARGIN=1, FUN=sum, na.rm=TRUE), pch=16, col="blue")

# Using the 'tAPPLY' function in R

# read in the "LungCapData.csv" data, and attach it

LungCapData <- read.table(file C:/Users/w4566/Desktop/台大/大三/數量方法與分析決策/LungCapData.txt", sep="\t", header=T)

# check the data

summary(LungCapData)

# and attach it

attach(LungCapData)

# calculate the mean Age for Smoker/NonSmoker

tapply(X=Age, INDEX=Smoke, FUN=mean, na.rm=T)

# you don't need to include "X", "INDEX",... as long as you

# ehter them in that order...

# we also don't need to include "na.rm=T" as no missing values

tapply(Age, Smoke, mean)

# we can save the output in a new "object"

m <- tapply(Age, Smoke, mean)

m

# also worth discussing is the use of the "SIMPLIFY" argument

# this is set to TRUE by default...if we set it to "FALSE"...

tapply(Age, Smoke, mean, simplify=FALSE)

# note that we could get the same using [ ],

# although using "tapply" is more efficient

mean(Age[Smoke=="no"])

mean(Age[Smoke=="yes"])

# let's look at applying the "summary" function to groups

tapply(Age, Smoke, summary)

# or, applying the "quantile" function to the groups

tapply(Age, Smoke, quantile, probs=c(0.2, 0.8))

# we can "subset" based on multiple variables/vectors

# calculate the mean Age for Smoker/NonSmoker and male/female

tapply(X=Age, INDEX=list(Smoke, Gender), FUN=mean, na.rm=T)

# a less efficient way to get this done...

mean(Age[Smoke=="no" & Gender=="female"])

mean(Age[Smoke=="no" & Gender=="male"])

mean(Age[Smoke=="yes" & Gender=="female"])

mean(Age[Smoke=="yes" & Gender=="male"])

# a reminder of using 2 grouping variables

tapply(Age, list(Smoke, Gender), mean, na.rm=T)

# an a note that the "by" function is the same as tapply,

# except it presents the results similar to a vector

by(Age, list(Smoke, Gender), mean, na.rm=T)

# and we can subset the elements in the usual way

temp <- by(Age, list(Smoke, Gender), mean, na.rm=T)

temp

temp[4]

# and see the "class" of temp

class(temp)

# we can also convert it to a vector if we prefer

c(temp)

temp2 <- c(temp)

temp2

# and check it's class

class(temp2)

