**2019/11/20  
108學年度第一學期 高維度資料分析 期中考**

系級: 姓名:

# 貼上執行程式碼及執行結果及圖形   
(程式碼字形為Courier New，10點字，單行距。)

(貼圖時，請注意圖形之大小，應適中; 圖內之文字數字應可識別。)

# 1 使用之套件: “mice” & “VIM”

#(a)

> mamm <- read.table("108-1-HDDA-MidtermExam/mammographic\_masses.data", sep = ",", na.strings = "?")

> names(mamm) <- c("BI-RADS", "Age", "Shape", "Margin", "Density", "Severity")

summary(mamm)

BI-RADS Age Shape Margin

Min. : 0.000 Min. :18.00 Min. :1.000 Min. :1.000

1st Qu.: 4.000 1st Qu.:45.00 1st Qu.:2.000 1st Qu.:1.000

Median : 4.000 Median :57.00 Median :3.000 Median :3.000

Mean : 4.348 Mean :55.49 Mean :2.722 Mean :2.796

3rd Qu.: 5.000 3rd Qu.:66.00 3rd Qu.:4.000 3rd Qu.:4.000

Max. :55.000 Max. :96.00 Max. :4.000 Max. :5.000

NA's :2 NA's :5 NA's :31 NA's :48

Density Severity

Min. :1.000 Min. :0.0000

1st Qu.:3.000 1st Qu.:0.0000

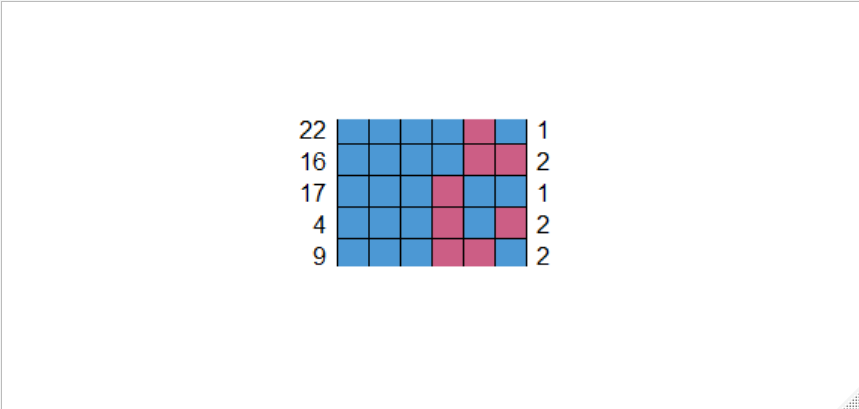
Median :3.000 Median :0.0000

Mean :2.911 Mean :0.4631

3rd Qu.:3.000 3rd Qu.:1.0000

Max. :4.000 Max. :1.0000

NA's :76



library(mice)

library(VIM)

md.pattern(mamm)

Severity BI-RADS Age Shape Margin Density

830 1 1 1 1 1 1 0

56 1 1 1 1 1 0 1

22 1 1 1 1 0 1 1

16 1 1 1 1 0 0 2

17 1 1 1 0 1 1 1

4 1 1 1 0 1 0 2

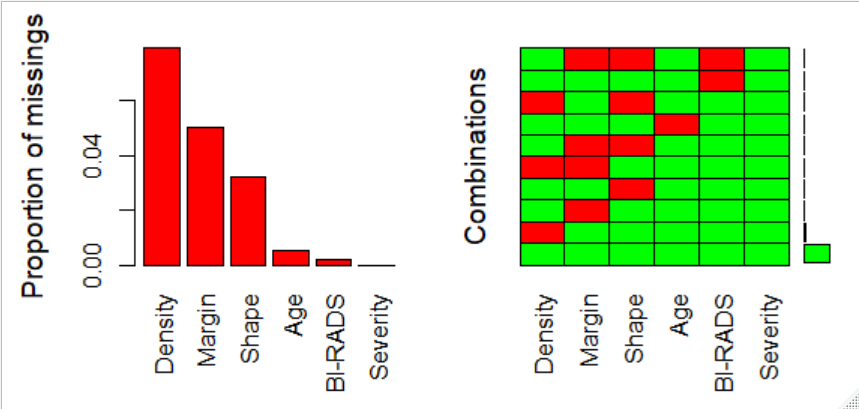
9 1 1 1 0 0 1 2

5 1 1 0 1 1 1 1

1 1 0 1 1 1 1 1

1 1 0 1 0 0 1 3

0 2 5 31 48 76 162



> matrixplot(mamm)

mamm.aggrplot <- aggr(mamm, col = c("green", "red"), nubers = T, prop = T, sortVars = T, labels = names(mamm))

Variables sorted by number of missings:

Variable Count

Density 0.079084287

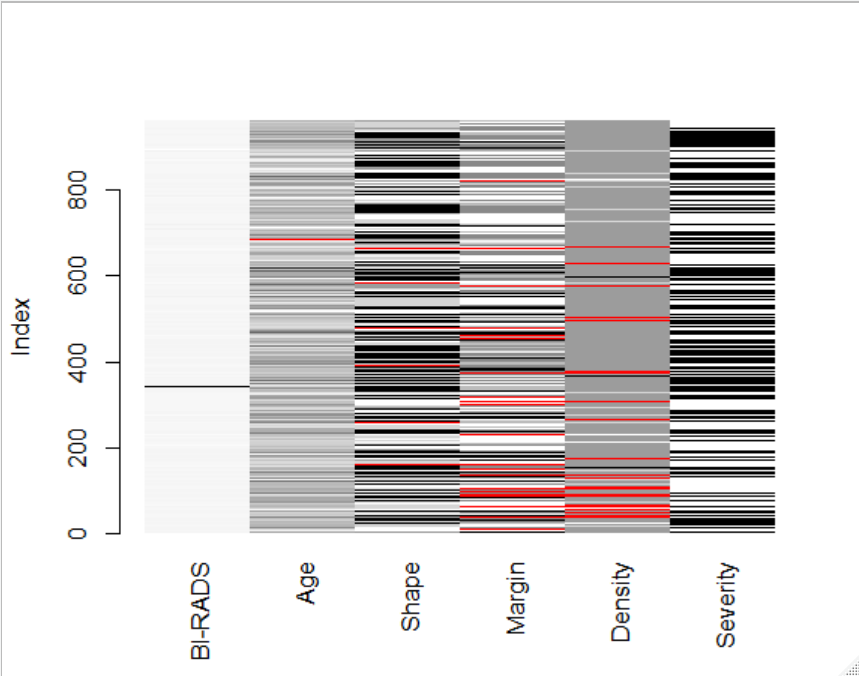
Margin 0.049947971

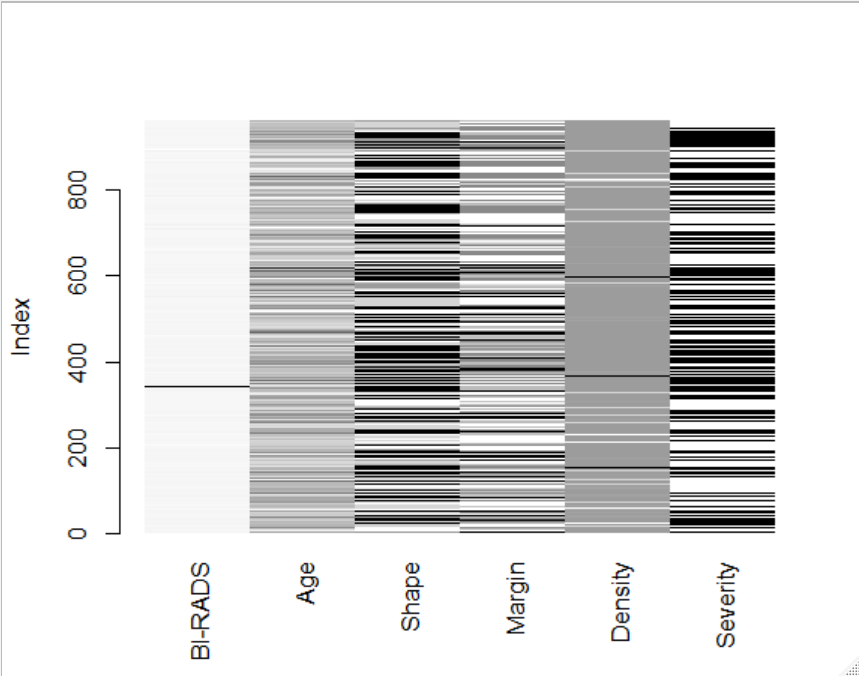
Shape 0.032258065

Age 0.005202914

BI-RADS 0.002081165

Severity 0.000000000





#(b)

> #Mean Substitution

> mean.subst <- function(x) {

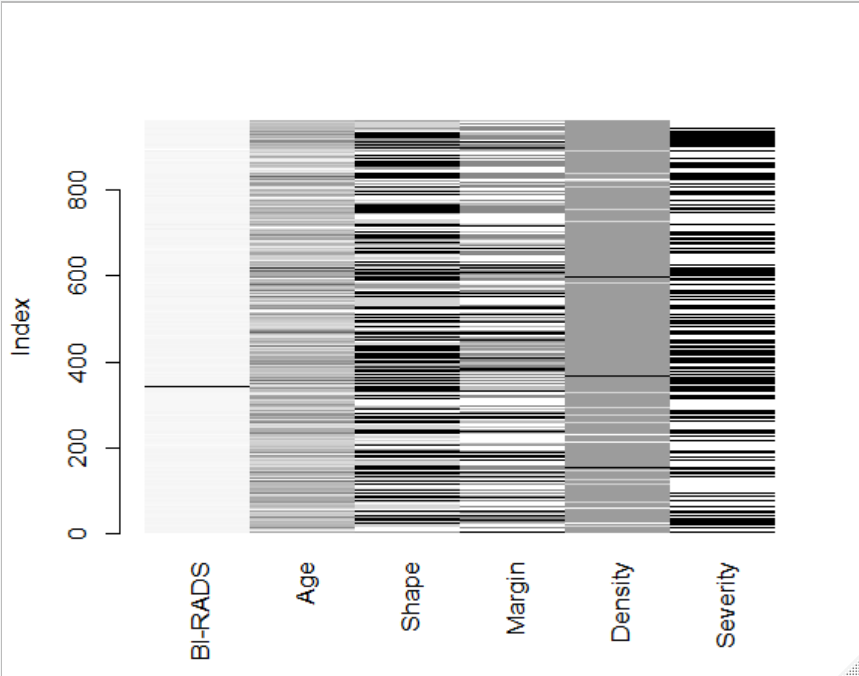
+ x[is.na(x)] <- mean(x, na.rm = TRUE)

+ x

+ }

> mamm.m <- apply(mamm, 2, mean.subst)

> matrixplot(mamm.m)



#kNN with 自訂平均函數

> t\_mean <- function(x){

+ mean(x, trim = 0.1)

+ }

>

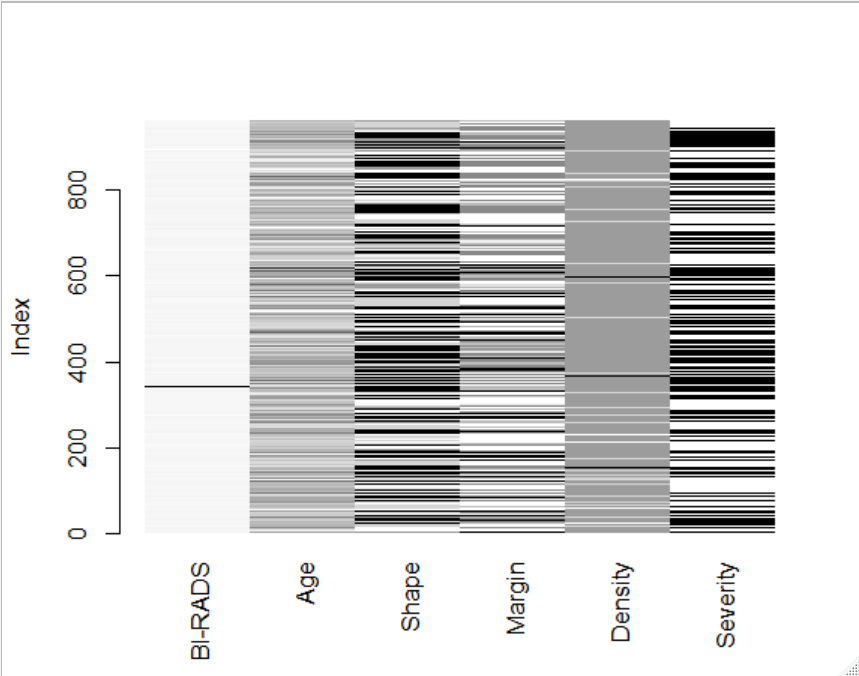
> mamm.kNNt <- kNN(mamm, k = 5, numFun = t\_mean)

> matrixplot(mamm.kNNt[1 : 6])

#kNN

> mamm.kNN <- kNN(mamm, k = 5)

> matrixplot(mamm.kNN[1 : 6])



由三種方法看不出有何差異，

3者補值後的矩陣圖大致相同。但是基本上KNN的方法會是比較好的一種。

#2

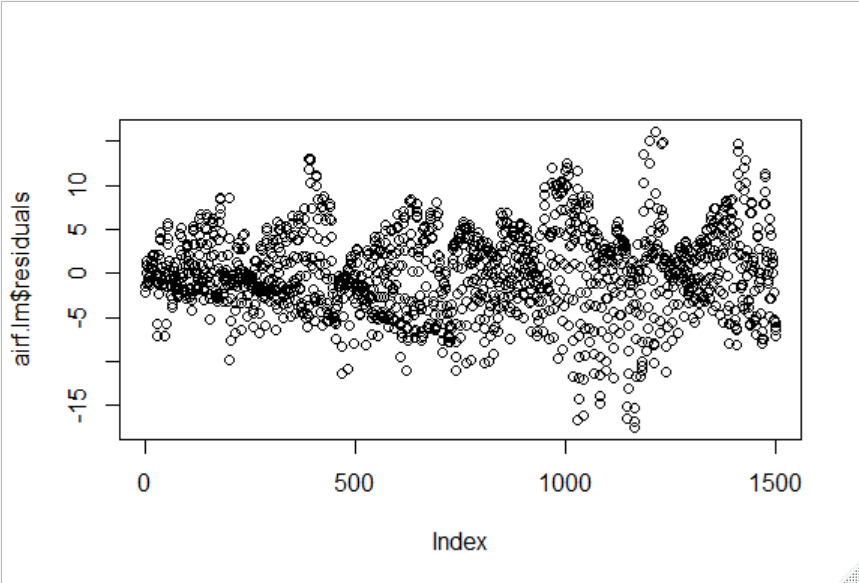
#2

> airf <- read.table("108-1-HDDA-MidtermExam/airfoil\_self\_noise.dat")

> names(airf) <- c("Frequency", "Angle\_of\_Attack", "Chord\_length", "Free\_stream\_velocity", "Ssdt", "Sspl")

> airf.lm <- lm(formula = Sspl ~ Frequency + Angle\_of\_Attack + Chord\_length + Free\_stream\_velocity + Ssdt, data = airf)

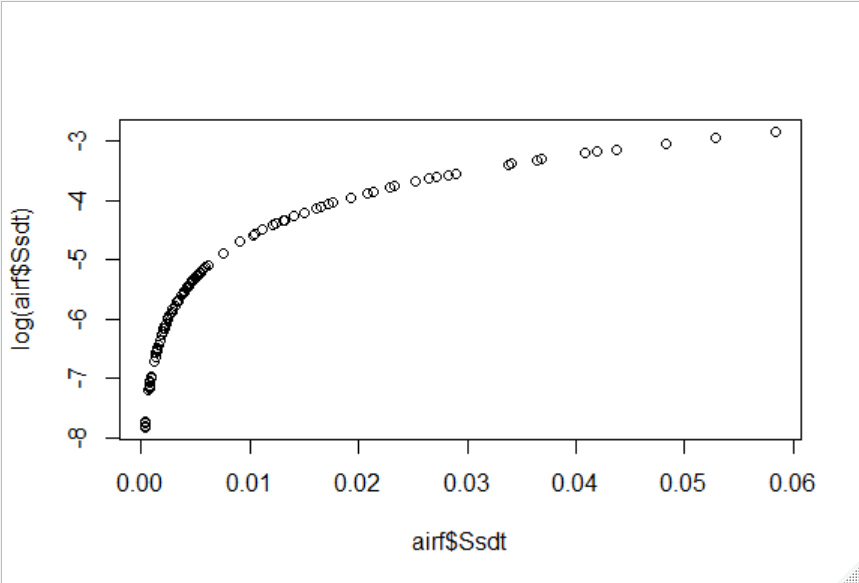
> plot(airf.lm$residuals)

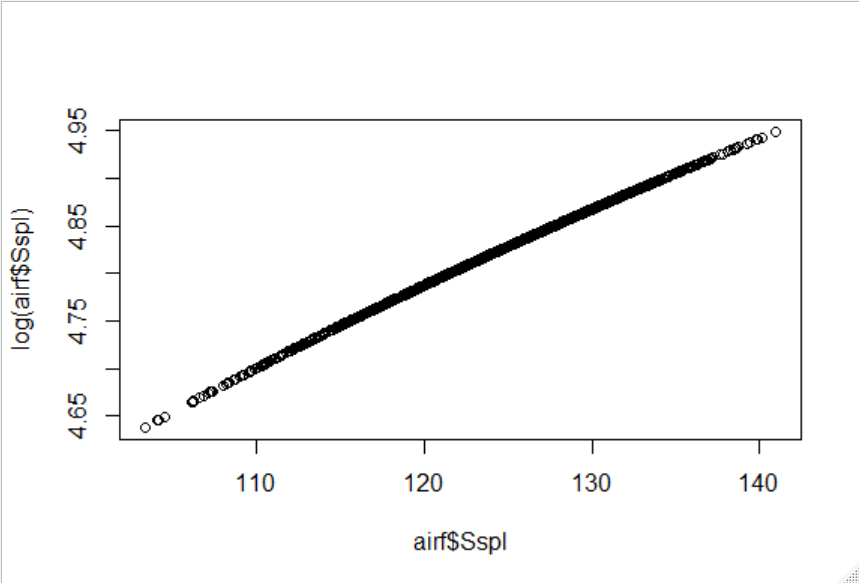


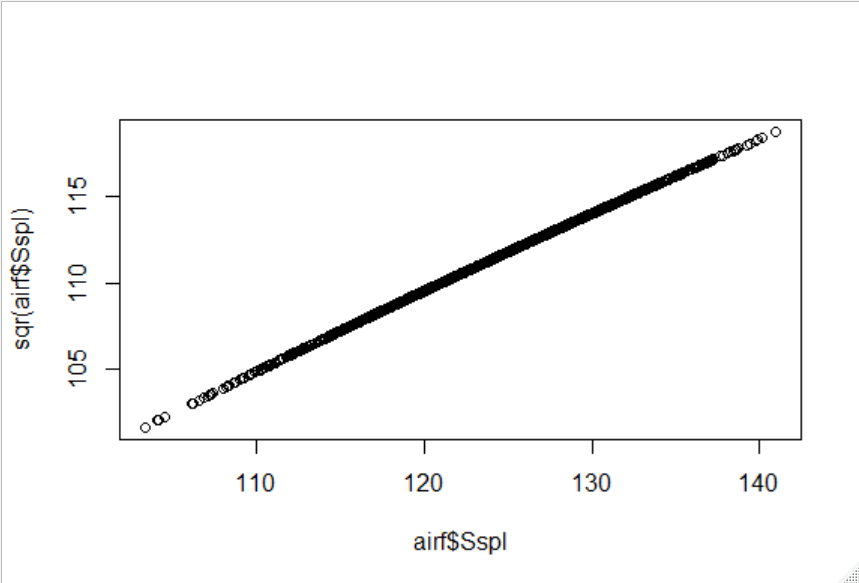
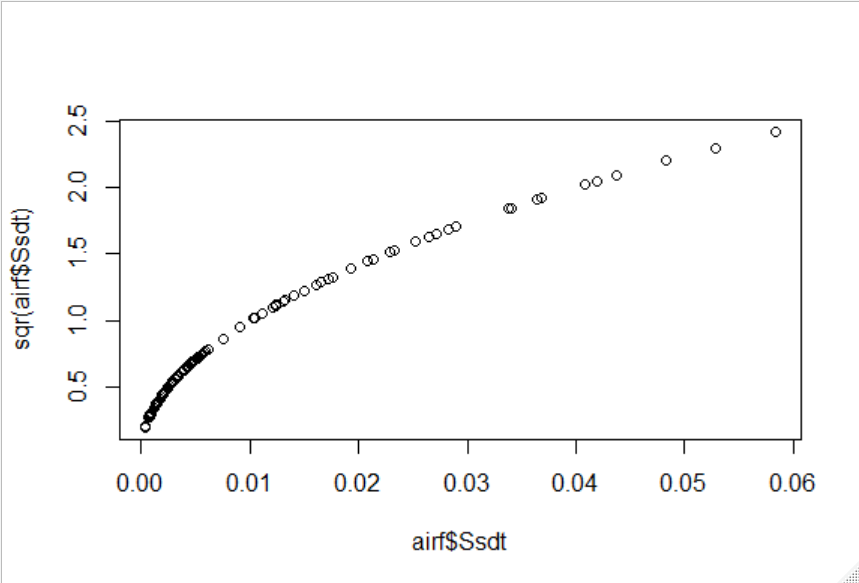
> #log

> plot(airf$Ssdt, log(airf$Ssdt))

> plot(airf$Sspl, log(airf$Sspl))





#SQRT\*10

> sqr <- function(x){

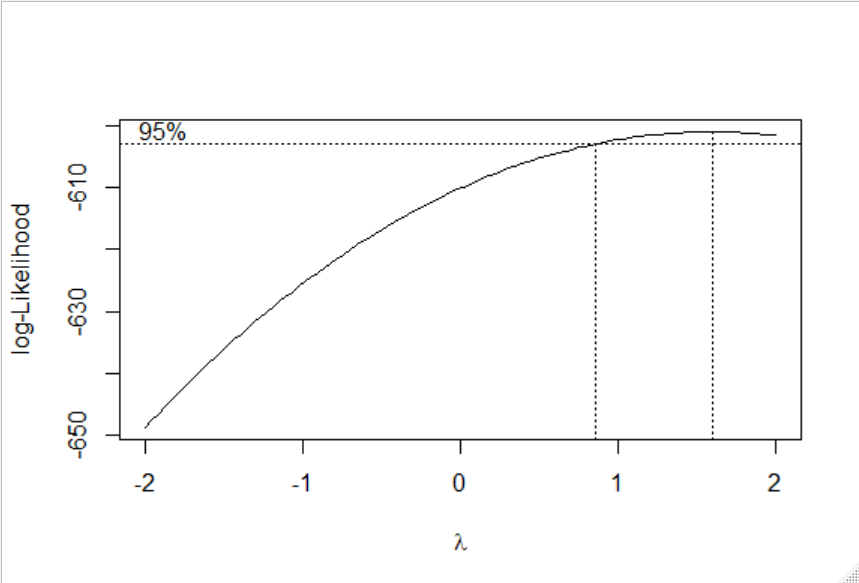
+ sqrt(x) \* 10

+ }

>

> plot(airf$Sspl, sqr(airf$Sspl))

> plot(airf$Ssdt, sqr(airf$Ssdt))



#Box-Cox

> library(MASS)

> airf.lm <- lm(formula = Sspl ~ Frequency + Angle\_of\_Attack + Chord\_length + Free\_stream\_velocity + Ssdt, data = airf)

>

> airfbc <- boxcox(Sspl ~ Frequency + Angle\_of\_Attack + Chord\_length + Free\_stream\_velocity + Ssdt, data = airf, lambda = seq(-2, 2, 1/10))

我認為BoxCox方法最佳。因為它可以藉由嘗試許多不同的Lamda值去配適最好的結果。

#3

#4

#地址轉經緯度

geoPoint <- function(address, key, verbose=FALSE) {

#若未輸入地址, return錯誤

if(verbose) cat(address,"\n")

root = "https://maps.googleapis.com/maps/api/place/findplacefromtext/"

#Google編碼為UTF8, 中文要轉碼後帶入URL才會正確

address = iconv(address,"big5","utf8")

#POI API型態(XML or JSON)

return.call = "json"

sensor = "false"

#產生URL

url\_gen = paste(root, return.call, "?input=", address, "&inputtype=textquery&fields=geometry&key=", key, sep = "")

#擷取網頁原始碼

html\_code = fromJSON(url\_gen)

#若status為OK抓取資料, 若不為OK return status

if(html\_code$status=="OK") {

return(cbind(html\_code$candidates$geometry$location, address))

} else {

return(paste("Status:", html\_code$status, sep = " "))

}

}

> library(jsonlite)

> covst <- read.table("108-1-HDDA-MidtermExam/SanShia7-11.csv", sep = ",", header = T)

> llpos <- matrix(NaN, length(covst$門市地址), 2)

> k <- 1

> for (i in 1 : length(covst$門市地址)) {

+ templl <- geoPoint(covst$門市地址[i], "AIUKxupk") #放入GoogleAPI金鑰

+ if(sum(templl[1] == "Status: ZERO\_RESULTS") == 1){

+ llpos[i, 1] <- NA

+ llpos[i, 2] <- NA

+ cat("遺漏", k, "個", "\n")

+ k <- k + 1

+ next()}

+ else{

+ llpos[i, 1] <- templl[1, 1]

+ llpos[i, 2] <- templl[1, 2]

+ cat("完成,", i, "個", "\n")}

+ }



> covst$lat <- llpos[, 1]

> covst$lng <- llpos[, 2]

> tw.newtaipei.zh <- get\_map(location = c(lon = 121.375, lat = 24.943403), zoom = 13, language = "zh-TW" , maptype = "roadmap")

Source : https://maps.googleapis.com/maps/api/staticmap?center=24.943403,121.375&zoom=13&size=640x640&scale=2&maptype=roadmap&language=zh-TW&key=xxx

> ggmap(tw.newtaipei.zh) + geom\_point(data = covst, aes(x = covst$lng, covst$lat, color = "red"), size = 1)