**HW4**

###### 請將Seizure1的資料整理成Seizure2的排列方式。(本題限用for迴圈回答) ####

setwd("c:/Users/acer/Desktop/R統計HW/")

sei <- read.csv("seizure.csv")

Seizure1 <- read.csv('seizure.csv')

Seizure2 <- read.table("seizure.txt")

tryS2 = matrix(NA,(length(Seizure1$y)/5),6)

for(v in 1:length(Seizure1$y)){

tryS2[v,1] <- Seizure1$trt[((v-1)\*5+1)]

for(i in 2:6){

tryS2[v,i] <- Seizure1$y[(v-1)\*5+(i-1)]

}

}

colnames(tryS2) <- paste("V", 1:6, sep="")

rownames(tryS2) <- seq(1,(length(Seizure1$y)/5),1)

tryS2

###### 請將Seizure2的資料整理成Seizure1的排列方式。(本題限用for迴圈回答) ####

head(Seizure1)

head(Seizure2)

tryS1 <- matrix(NA,dim(Seizure2)[1]\*5,2)

for(v in 1:dim(Seizure2)[1]){

for(n in 2:6){

tryS1[(v-1)\*5+(n-1),1] <- Seizure2[v,n]

tryS1[(v-1)\*5+(n-1),2] <- Seizure2[v,1]

}

}

colnames(tryS1) <- c("y","trt")

rownames(tryS1) <- seq(1,dim(Seizure2)[1]\*5,1)

tryS1

##### Ex S1: 請分別使用for及while迴圈來計算factorial(10)的結果。####

fac <- 10

for(fn in 1:fac){

ifelse(fn == 1, multi <- fn, multi <- multi\*fn)

}

multi

count <- 1

while(count <= fac){

ifelse(count == 1, multiw <- count, multiw <- multiw\*count)

count = count + 1

}

multiw

# Ex S2: 設有 A、 B、 C 三根柱子，A柱子上有20個直徑大小不一的中空圓盤，由大而小疊放在一起，####

# 如下圖(以3個圓盤為例)所示。每次搬一個圓盤，在搬的過程中，直徑大的不可放在直徑小的上面，

# 三根柱子都可放圓盤。試問最少需要搬多少次，方能將這些圓盤從A柱搬到C柱？請以遞迴函式來回答。

hanoi <- function(n, A, B, C) {

if(n == 1){

return(length(c(A, C))/2)

}else {

return(hanoi(n-1, A, C, B) + hanoi(1, A, B, C) + hanoi(n-1, B, A, C))

}

}

hanoi(20, 'A', 'B', 'C')

# for(m in 1:(length(hanoi(num,'A', 'B', 'C'))/2)){

# sprintf("圓盤由 %s 移至 %s", c(hanoi(num,'A', 'B', 'C'))[(2\*(m-1))+1], c(hanoi(num,'A', 'B', 'C'))[2\*m])

# }

# Ex S3: 若學生依座位表(五列十行)過年領的紅包總額如下，請使用apply指令求出每一列、每一行的中位數、最大值、最小值，NA者略過不計。

x <- matrix(c(3600, 5000, 12000, NA, 1000, 2000, 600, 7500, 1800, 9000,

3600, 4500, 10000, 8500, 3000, 10000, 1000, NA, 1200, 10000,

3800, 5500, 9000, 6000, 6600, 3000, 9600, 6500, 8200, 8000,

5000, 6600, 13000, 4500, 5000, NA, 10600, 9500, 7600, 6000,

6600, 8000, 17000, 3000, 7000, 1000, 12600, 8500, 6000, NA),5,10, byrow = TRUE)

arg <- c("median", "max", "min")

mod <- c(1,2)

for(i in mod){

for(argg in arg){

print(apply(x,i, argg, na.rm = T))

}

}

##### Ex S5: (1) 將課程網站上的seizure.csv裡的trt, age, ltime當成解釋變數，y當成反應變數，求線性迴歸係數值，並需與R的內建函數lm結果比對。#####

setwd("c:/Users/acer/Desktop/R統計HW/")

Seizure1 <- read.csv('seizure.csv')

lm.obj <- lm(Seizure1$y ~ Seizure1$trt + Seizure1$age + Seizure1$ltime)

SY <- matrix(Seizure1$y,length(Seizure1$y),1)

SX <- matrix(c(matrix(1,length(Seizure1$y),1), Seizure1$trt, Seizure1$age, Seizure1$ltime), length(Seizure1$y), 4)

b <- t(solve(t(SX)%\*%SX)%\*%t(SX)%\*%SY)

colnames(b) <- c("(Intercept)", "Seizure1$trt", "Seizure1$age", "Seizure1$ltime")

b

##### Ex S5: (2) 承上小題，請自行計算出殘差，並需與lm計算的殘差作比對。

n <- length(Seizure1$y)

SH <- SX%\*%solve(t(SX)%\*%SX)%\*%t(SX)

I <- diag(n)

Se <- (I-SH)%\*%SY

head(Se)

head(lm.obj$residuals)

##### Ex S7: 請用矩陣搭配solve指令回答。####

A <- matrix(c(2,-1,1,4,-1,3,2,-3,2),3,3, byrow = T)

b <- c(3,2,1)

solve(A,b)

##### Ex S8: 請用矩陣搭配solve指令回答。####

A <- matrix(c(2,-1,1,4,-4,3,2,-3,2),3,3, byrow = T)

b <- c(3,2,1)

solve(A,b)

A:此題無解 Error in solve.default(A, b) : Lapack routine dgesv: system is exactly singular: U[3,3] = 0

##### Ex S9: (1) 請使用ceiba課程網站上的seizure.csv裡的y，以R指令來計算

setwd("c:/Users/acer/Desktop/R統計HW/")

sei <- read.csv("seizure.csv")

Seizure1 <- read.csv('seizure.csv')

SY <- matrix(Seizure1$y,length(Seizure1$y),1)

tY <- t(SY)

n <- length(Seizure1$y)

I <- diag(n)

J <- matrix(1,n,n)

A <- (I-(1/n)\*J)

SST <- tY%\*%A%\*%SY

SST

##### Ex S9: (2) (紙筆計算) 與統計學上哪一個常用的量數有關？關係為何？

在這邊的 y'Ay 就是ANOVA中的SST(總平方和) 但拆解來看這邊的A矩陣是由單位矩陣I 減去 (1/n) \* J(1矩陣)而得來

##### Ex S9: (3) 以R指令得到該統計學上量數，並與(1)比對。

aov.Y <- aov(y~ltime, data=Seizure1)

S.aov.Y <- summary(aov.Y)

SST

sum(S.aov.Y[[1]][2])