|  |  |  |
| --- | --- | --- |
| Assignment 1 | August 12  15338673 | |
| Paul-Willem Janse van Rensburg | | Multivariate Statistical Analysis |

1. Multivariate paired comparison hypothesis test: Write an R program that uses a multivariate dataset and level of significance as arguments to do a hypothesis test as described on p.275. The output of the program must contain the following: (i)- test statistic, (ii) critical value, (iii) p-value and (iv) a conclusion. Use the program and the Effluent data (Table 6.1) to test the hypothesis  at . Do you obtain the same answers as Example 6.1?

library(data.table)

library(here)

library(MASS)

convert\_to\_paired\_diff <- function(df){

## Convert a dataframe of treatments to a paired difference dataframe

## param df: dataframe to be converted

## return: list with paired difference dataframe, d, number of variable, p, and number of

## observations, n

# Are there equal number of vars in both treatments

if(ncol(df)%%2 != 0){

stop('Not equal number of variables for treatment 1 and 2')

}

#Number of vars

p = ncol(df)/2

#Number of obs

n <- nrow(df)

#Calculate difference between variables for treatments

d <- matrix(ncol=p,nrow=n)

for(i in 1:p){

col\_name <- paste0('d',i)

d[,i] <- df[,i]-df[,i+p]

}

#Return list with paired difference dataframe, number of vars and number of obs

return(list('d'=as.data.frame(d), 'p' = p, 'n' = n))

}

hypothesis\_test <- function(df, significance){

## Function to do a paired-difference hypothesis test, outputting the calculated t-squared

## along with the critical value and also the result of the hypothesis test and it's p-value

## param df: Dataframe containing treatment 1 and treatment 2 data

## param significance: significance level of test

#Retrieve paired differences

paired\_diff <- convert\_to\_paired\_diff(df)

d <- paired\_diff$d

p <- paired\_diff$p

n <- paired\_diff$n

#Calc stats

dbar <- colMeans(d)

sd <- cov(d)

t\_square <- n\*dbar%\*%solve(sd)%\*%dbar

crit <- ((n-1)\*p/(n-p))\*qf(1-significance,p,n-p)

p\_val <- 1-pf(t\_square, p, n-p)

#Logic to reject or accept hypothesis at specific level

if(t\_square > crit){

print(paste0('T^2 = ',round(t\_square, 4),' > ', round(crit, 4),

' We thus reject the null hypothesis and conclude that there is a nonzero mean difference between the measurements with significance level alpha = '

, significance))

}else{

print(paste0('T^2 = ',round(t\_square, 4),' < ', round(crit, 4),

' We thus do not reject the null hypothesis and conclude that there is not a significant difference between the measurements at significance level alpha = '

, significance))

}

#Also return p-value

print(paste0('p-value = ', p\_val))

}

table\_6\_1 <- as.data.frame(fread(file = here('assignment\_1/T6-1.dat')))

hypothesis\_test(table\_6\_1, 0.05)

[1] "T^2 = 13.6393 > 9.4589 We thus reject the null hypothesis and conclude that there is a nonzero mean difference between the measurements with significance level alpha = 0.05"

[1] "p-value = 0.00188652536222111"

1. Confidence intevals for  (p. 276): Consider the following 2 intervals

-confidence interval

.

Bonferroni confidence interval

.

Write an R program that gives these intervals as output. The program must receive themultivariate data set, vector **a** and the level of significance as arguments. Use the program and calculate 95% confidence intervals for the Effluent data (Table 6.1). Do you obtain the same answers as Example 6.1?

confidence\_interval <- function(df, a, significance){

## Function to construct T^2 and Bonferroni confidence interval

## param df: dataframe for which to construct confidence interval

## param a: vector in form to select variable for which to construct CI

## param significance: significance level for which to construct CI

## return: list with T^2 CI, t\_conf\_int, and Bonferroni CI, bonf\_conf\_int

#Retrieve paired differences

paired\_diff <- convert\_to\_paired\_diff(df)

d <- paired\_diff$d

p <- paired\_diff$p

n <- paired\_diff$n

#Calc stats

dbar <- colMeans(d)

sd <- cov(d)

#Build up factors to add and subtract using supplied functions

t\_conf\_int <- sqrt((((n-1)\*p/(n-p))\*qf(1-significance,p,n-p))\*(a%\*%sd%\*%a/n))

bonf\_conf\_int <- qt(1-significance/(2\*p), n-1)\*sqrt((a%\*%sd%\*%a/n))

#Return list with CI's

return(list('t\_conf\_int' = c(a%\*%dbar - t\_conf\_int, a%\*%dbar + t\_conf\_int),

'bonf\_conf\_int' = c(a%\*%dbar - bonf\_conf\_int, a%\*%dbar + bonf\_conf\_int)))

}

a <- c(1,0)

confidence\_interval(table\_6\_1, a,0.05)

$t\_conf\_int

[1] -22.45327 3.72600

$bonf\_conf\_int

[1] -20.573107 1.845835

a <- c(0,1)

confidence\_interval(table\_6\_1, a,0.05) $t\_conf\_int

[1] -5.700119 32.245574

$bonf\_conf\_int

[1] -2.974903 29.520358

1. Confidence region for the mean,: Consider the following expression for the  confidence region for the mean (p. 276):

.

Use this expression and write an R program to construct a confidence region for any two dimensional dataset. Thus, your R program should use any two dimensional dataset and level of significance as arguments to draw an ellipse. Use this program and draw an ellipse for the Effluent data (Table 6.1).

draw\_ellipse <- function(df, significance){

## Function to draw confidence region

## param df: two dimensional dataframe for which to construct confidence region

## significance: significance level for which to construct confidence region

#Retrieve paired differences

p <- ncol(df)

n <- nrow(df)

# Calc stats

dbar <- colMeans(df)

sd <- cov(df)

crit <- ((n-1)\*p/(n-p))\*qf(1-significance,p,n-p)

#Setup region

x\_min <- min(df[,1])

x\_max <- max(df[,1])

y\_min <- min(df[,2])

y\_max <- max(df[,2])

x <- seq(x\_min,x\_max,length=100)

y <- seq(y\_min,y\_max,length=100)

grid\_data <- as.matrix(expand.grid(x,y))

ng<-nrow(grid\_data)

np<-length(y)

# Obtaining points inside the elliptic region using sign()

sign\_grid<-matrix(rep(0,ng),nrow=ng,ncol=1)

for (i in 1:ng){

sign\_grid[i,]<-sign((t(dbar - grid\_data[i,])%\*%solve(sd)%\*%(dbar-grid\_data[i,]))-crit)

}

# Plotting the boundary of ellipse

plot(df[,1],df[,2],xlim=c(x\_min,x\_max),ylim=c(y\_min,y\_max),xlab="x",ylab="y",

col="red",pch=16,cex=1.25)

par(new=T)

contour(x,y,matrix(sign\_grid,np),add=TRUE,drawlabels=FALSE,levels=0,lty=2,lwd=2,

col="black")

mat<-cbind(sign\_grid,grid\_data)

region<-matrix(mat[mat[,1]==-1],ncol=3)

# Plotting the points inside the ellipse

par(new=T)

plot(region[,2:3],xlim=c(x\_min,x\_max),ylim=c(y\_min,y\_max),col="red",xlab="x",

ylab="y",pch=".",main="Ellipse" )

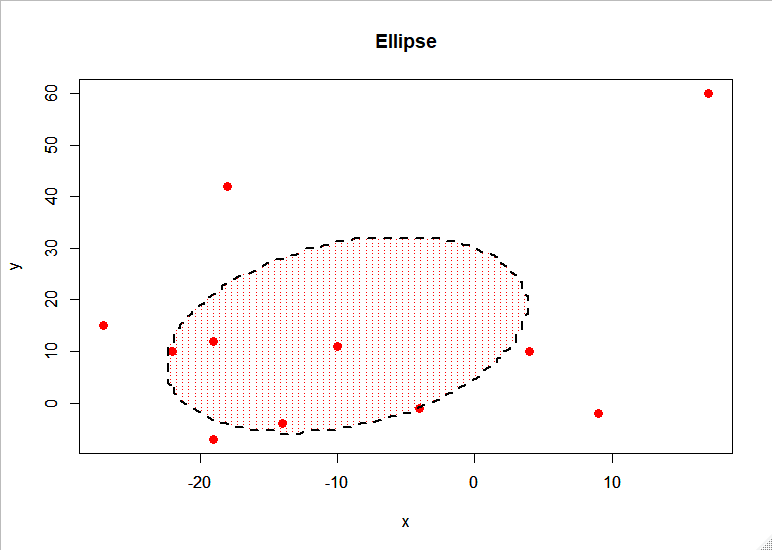
}

paired\_diff <- convert\_to\_paired\_diff(table\_6\_1)

d <- paired\_diff$d

significance <- 0.05

draw\_ellipse(d, significance)



1. Multivariate repeated measures design: Write an R program that uses a multivariate dataset, matrix **C** and level of significance as arguments to do a hypothesis test as described on p.280. The output of the program must contain the following: (i)- test statistic, (ii) critical value, (iii) p-value (iv) a conclusion and (v) the confidence intervals obtained using the following formula on p. 281

.

Use this program and the Sleeping-Dog data (Table 6.2) and repeat Example 6.2 at .

eq\_of\_treatments\_in\_rep\_des <- function(df, C, significance){

## Function to do a hypothesis test for equality of treatments in repeated design, with a confidence interval

## param df: dataframe containing measurements of all treatments applied to each unit (each column as a treatment)

## param C: contrast matrix

## param significance: significance level of hypothesis test

# Calc stats

n <- nrow(df)

q <- ncol(df)

dbar <- colMeans(df)

sd <- cov(df)

t\_squared <- n\*t(C%\*%dbar)%\*%solve(C%\*%sd%\*%t(C))%\*%C%\*%dbar

crit <- (n-1)\*(q-1)/(n-q+1)\*qf(1-significance,q-1,n-q+1)

p\_val <- 1-pf(t\_squared, p, n-p)

#Perform hypothesis test

if(t\_squared > crit){

print(paste0('T^2 = ', t\_squared, ' > ',crit))

print('We therefore reject the null hypothesis H0:C\*mu = 0 and determine that there is possibly treatment effects')

}else{

print(paste0('T^2 = ', t\_squared, ' < ',crit))

print('We therefore do not reject the null hypothesis H0:C\*mu = 0 and determine that there is no treatment effects')

}

print(paste0('p-value = ', p\_val))

#Build up confidence intervals

apply(C,1,function(c){

ci <- sqrt(crit\*(t(c)%\*%sd%\*%c/n))

print(paste0('100(1-',significance,')% Confidence interval: (',c%\*%dbar-ci,'; ',c%\*%dbar+ci,')'))

})

}

C <- matrix(c(-1,-1,1,1,1,-1,1,-1,1,-1,-1,1),nrow=3,byrow = TRUE)

table\_6\_2 <- as.data.frame(fread(file = here('assignment\_1/T6-2.dat')))

eq\_of\_treatments\_in\_rep\_des(table\_6\_2, C, 0.05)

[1] "T^2 = 116.016321200988 > 10.9311913714058"

[1] "We therefore reject the null hypothesis H0:C\*mu = 0 and determine that there is possibly treatment effects"

[1] "p-value = 1.23207666291592e-10"

[1] "100(1-0.05)% Confidence interval: (135.650297268848; 282.98128167852)"

[1] "100(1-0.05)% Confidence interval: (-114.727082717168; -5.37818044072652)"

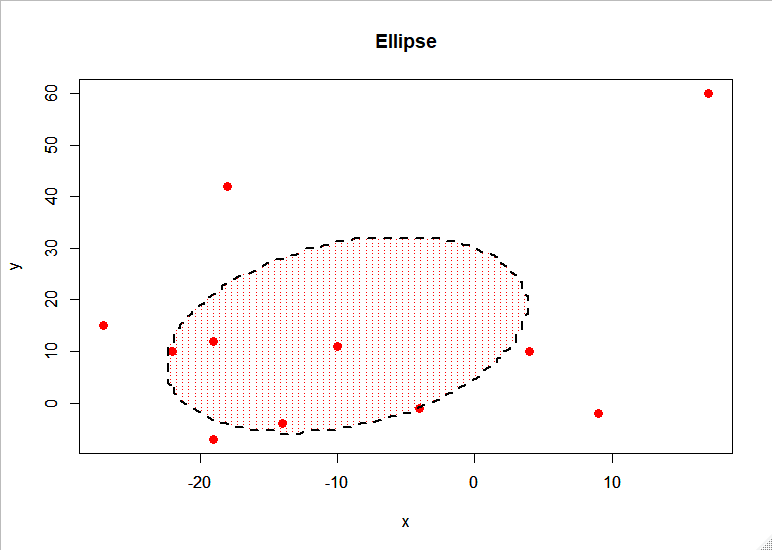
[1] "100(1-0.05)% Confidence interval: (-78.7285843929882; 53.1496370245671)"

[1] "100(1-0.05)% Confidence interval: (135.650297268848; 282.98128167852)"

[2] "100(1-0.05)% Confidence interval: (-114.727082717168; -5.37818044072652)"

[3] "100(1-0.05)% Confidence interval: (-78.7285843929882; 53.1496370245671)"

1. Use R to do the following exercises:
2. Exercise 6.1



The hypothesis was rejected in Example 6.1, the same as we would have were we to consider just the confidence region. However, the confidence intervals do include the 0 point, both in the simultaneous and Bonferroni cases. The analysis assumes a normal distribution of Dj, complicating the situation further with 1 or 2 outliers. The sample is too small to remove the effects of outliers, and further investigation is needed.

1. Exercise 6.2

As noted in (2) above, the interval appears to be narrower than that of the simultaneous confidence intervals for the individual mean differences.

1. Exercise 6.8

treatment\_1 <- matrix(c(6,5,8,4,7,7,9,6,9,9), nrow=5)

treatment\_2 <- matrix(c(3,1,2,3,6,3), nrow=3)

treatment\_3 <- matrix(c(2,5,3,2,3,1,1,3), nrow=4)

mu1 <- colMeans(treatment\_1)

mu2 <- colMeans(treatment\_2)

mu3 <- colMeans(treatment\_3)

mu <- (mu1+mu2+mu3)/3

est\_treatment\_effect\_1 <- mu1-mu

est\_treatment\_effect\_2 <- mu2-mu

est\_treatment\_effect\_3 <- mu3-mu

residual\_1 <- treatment\_1-mu1

residual\_2 <- treatment\_2-mu2

residual\_3 <- treatment\_3-mu3

> assertthat::are\_equal(treatment\_1, mu+est\_treatment\_effect\_1+residual\_1)

[1] TRUE

> assertthat::are\_equal(treatment\_2, mu+est\_treatment\_effect\_2+residual\_2)

[1] TRUE

> assertthat::are\_equal(treatment\_3, mu+est\_treatment\_effect\_3+residual\_3)

[1] TRUE

B <- nrow(treatment\_1)\*(mu1-mu)%\*%t(mu1-mu) + nrow(treatment\_2)\*(mu2-mu)%\*%t(mu2-mu) + nrow(treatment\_3)\*(mu3-mu)%\*%t(mu3-mu)

W <- (t(treatment\_1)-mu1)%\*%t(t(treatment\_1)-mu1) + (t(treatment\_2)-mu2)%\*%t(t(treatment\_2)-mu2) + (t(treatment\_3)-mu3)%\*%t(t(treatment\_3)-mu3)

total <- (t(treatment\_1)-mu)%\*%t(t(treatment\_1)-mu) + ((t(treatment\_2)-mu)%\*%t(t(treatment\_2)-mu)) + ((t(treatment\_3)-mu)%\*%t(t(treatment\_3)-mu))

> assertthat::are\_equal(B+W, total)

[1] TRUE

|  |  |  |  |
| --- | --- | --- | --- |
| Source of variation | Matrix SS | | Degrees of Freedom |
| Treatment | 37.333 | 49.333 | 2 |
| 49.333 | 85.333 |
| Residual | 18 | -13 | 9 |
| -13 | 18 |
| Total (corrected | 55.333 | 36.333 | 11 |
| 36.333 | 103.3333 |

1. Exercise 6.23 (Write your own R program for Box’s test)
2. Use PROC GLM of SAS to do the following:
3. Repeat Examples 6.9 and 6.13
4. Exercise 6.33 (a), (b) and (c). What are your conclusions?