Homework4

Yi Chen

2/16/2020

Homework 4

Problem 1

```
x1 <- c(0.48,40.53,2.19,0.55,0.74,0.66,0.93,0.37,0.22)
x2 <- c(12.57,73.68,11.13,20.03,20.29,0.78,4.64,0.43,1.08)
x <- cbind(x1,x2)
x_bar <- colMeans(x)
mu0 <- c(0.5,20)
n <- nrow(x)
s <- cov(x)
T_square <- n * t(x_bar - mu0) %*% solve(s) %*% (x_bar - mu0)
T_square
```

```
##          [,1]
## [1,] 21.54358
```

The Hoterlling's T^2 statistics is 5799.275.

```
p <- ncol(x)
alpha <- 0.08
scaled_F <- (((n-1)*p)/(n-p)) * qf(1-alpha, p, n-p)
scaled_F
```

```
## [1] 8.462389
```

Scaled F value is 8.46.

Conclusion: Since Hoterlling's T^2 statistics is bigger than scale F cirtical value, we reject H0. The mean vector is significantly differ from the mu vector.

```
p.value <- pf(((n-p)/((n-1)*p))*T_square, p, n-p,lower.tail=FALSE)
p.value
```

```
##            [,1]
## [1,] 0.01033223
```

The p-value is 9.703122e-11, which is much smaller than alpha.

Problem 2

```
x <- read.table("T6-1.DAT")
x <- x[-c(8),]
x
```

```
##    V1 V2 V3 V4
## 1   6 27 25 15
## 2   6 23 28 13
## 3  18 64 36 22
## 4   8 44 35 29
## 5  11 30 15 31
## 6  34 75 44 64
## 7  28 26 42 30
## 9  43 54 34 56
## 10 33 30 29 20
## 11 20 14 39 21
```

```
d = cbind(x$V1 - x$V3, x$V2 - x$V4)
dbar = colMeans(d)

T.sq = nrow(d) * t(dbar) %*% solve(cov(d)) %*% dbar
T.sq
```

```
##          [,1]
## [1,] 11.44652
```

```
n <- 10
p <- 2
alpha = 0.05
scaled_F <- (((n-1)*p)/(n-p)) * qf(1-alpha, p, n-p)
scaled_F
```

```
## [1] 10.03268
```

The Hoterlling's T^2 is still bigger than the scaled F critical value, which mean we need to reject the H0.

```
p.value <- pf(((n-p)/((n-1)*p))*T.sq, p, n-p,lower.tail=FALSE)
p.value
```

```
##             [,1]
## [1,] 0.03753983
```

Anagin, the p-value is 0.04 which is still samller than the alpha 0.05.

Problem 3

question a

In this proble, we need to compare the equality of vector means from t wo multivariate populations (the population is indexed by variable 8). The hyphothesis in this question is:

$$\overrightarrow{\mu_1} - \overrightarrow{\mu_2} = \overrightarrow{0}$$

```
x <- read.table("T6-15.DAT")
sample_1 <- x[which(x$V8==0),1:7]
sample_2 <- x[which(x$V8==1),1:7]
n1 <- nrow(sample_1)
n2 <- nrow(sample_2)
s1 <- cov(sample_1)
s2 <- cov(sample_2)
s_pooled <- ((n1 - 1)/(n1 + n2 - 2))*s1 + ((n2 - 1)/(n1 + n2 - 2))*s2

x1.bar <- colMeans(sample_1)
x2.bar <- colMeans(sample_2)
T.square <- t(x1.bar-x2.bar) %*% solve((1/n1+1/n2)*s_pooled) %*% (x1.bar-x2.ba
r)
T.square
```

```
##          [,1]
## [1,] 106.1348
```

```
p <- nrow(sample_1)
alpha <- 0.05
scaled_F <- (((n1 + n2 - 2) * p)/(n1 + n2 - p -1))*qf(1-alpha, p, n1+n2-p-1)
scaled_F
```

```
## [1] 123.6899
```

The Hoterlling's T^2 is still bigger than the scaled F critical value, which mean we need to reject the H0. The mean of two species are not equal for all variables at the same time.

question b

We can test each variable one by one with the hyphothees that the means between two population are equal.

```
p <- 1
for (i in 1:7){
  print(paste("p-value for the variable",as.character(i),sep = " "))
  Sp <- ((n1 - 1)/(n1 + n2 - 2))*s1[i,i] + ((n2 - 1)/(n1 + n2 - 2))*s2[i,i]
  T.sq.test <- (n1*n2/(n1+n2))*t(x1.bar[i]-x2.bar[i]) %*% solve(Sp) %*% (x1.bar
[i]-x2.bar[i])
  print(as.numeric(sqrt(T.sq.test)))
  print(as.numeric(pf(((n1+n2-p-1)/((n1+n2-2)*p))*T.sq.test, p ,n1+n2-p-1,lowe
r.tail=FALSE)))
}
```

```
## [1] "p-value for the variable 1"
## [1] 2.011733
## [1] 0.04821276
## [1] "p-value for the variable 2"
## [1] 0.8497474
## [1] 0.3984474
## [1] "p-value for the variable 3"
## [1] 6.501135
## [1] 1.109926e-08
## [1] "p-value for the variable 4"
## [1] 0.3428311
## [1] 0.732783
## [1] "p-value for the variable 5"
## [1] 4.92855
## [1] 5.59703e-06
## [1] "p-value for the variable 6"
## [1] 0.3256211
## [1] 0.7457109
## [1] "p-value for the variable 7"
## [1] 1.441561
## [1] 0.1540158
```

According to these p value, we can see the, variable 3 contribute most to rejection. The standardized distance between the means for varible 3 is also the biggest: 6.50.

question c

The fist variable's confidence interval can be calculated in the following way. We print out the lower bound of the confidence interval, then the upper bound.

```
c.sq = (n1 + n2 - 2)*p/(n1+ n2 -p - 1)*qf(1-alpha,p, n1 + n2 - p -1)
## lower bound
(x1.bar[1] - x2.bar[1]) - sqrt(c.sq) * sqrt((1/n1 + 1/n2)*s_pooled[1,1])
```

```
##        V1
## -5.748098
```

```
## up bound
(x1.bar[1] - x2.bar[1]) + sqrt(c.sq) * sqrt((1/n1 + 1/n2)*s_pooled[1,1])
```

```
##         V1
## -0.0233304
```

The rest 6 variable can use the same method.

```
for (i in 2:7){
  print(paste("for the variable",as.character(i),sep = " "))
  ## lower bound
  print(c((x1.bar[i] - x2.bar[i]) - sqrt(c.sq) * sqrt((1/n1 + 1/n2)*s_pooled[i,
i]),
          (x1.bar[i] - x2.bar[i]) + sqrt(c.sq) * sqrt((1/n1 + 1/n2)*s_pooled[i,
i])))
}
```

```
## [1] "for the variable 2"
##        V2        V2
## -2.774313  1.117170
## [1] "for the variable 3"
##        V3        V3
## -5.153084 -2.732630
## [1] "for the variable 4"
##         V4         V4
## -0.9743656  0.6886513
## [1] "for the variable 5"
##        V5        V5
## -6.141330 -2.601527
## [1] "for the variable 6"
##         V6         V6
## -0.6109880  0.4395594
## [1] "for the variable 7"
##         V7         V7
## -0.1317400  0.8174543
```