

Homework 3

Gianni Spiga

5/1/2022

Problem 1

6.1.)

```
#pg 359
effluent <- read.table("~/GitHub/STA135/Homework/HW3/T6-1.dat")
colnames(effluent) <- c("x1BOD", "x2SS", "x3BOD", "x4SS")

#statistical differences from textbook
meanDiff <- c(-9.36, 13.27)
S <- matrix(c(199.26, 88.38, 88.38, 418.61),
            nrow = 2,
            ncol = 2)
n <- 11

fstat = (((n - 1) * 2) / 9) * qf(0.05, 2, 9, lower.tail = FALSE)

eig <- eigen(S)
eig$values
```

```
## [1] 449.7882 168.0818
```

```
eig$vectors
```

```
##           [,1]      [,2]
## [1,] 0.3326804 -0.9430396
## [2,] 0.9430396  0.3326804
```

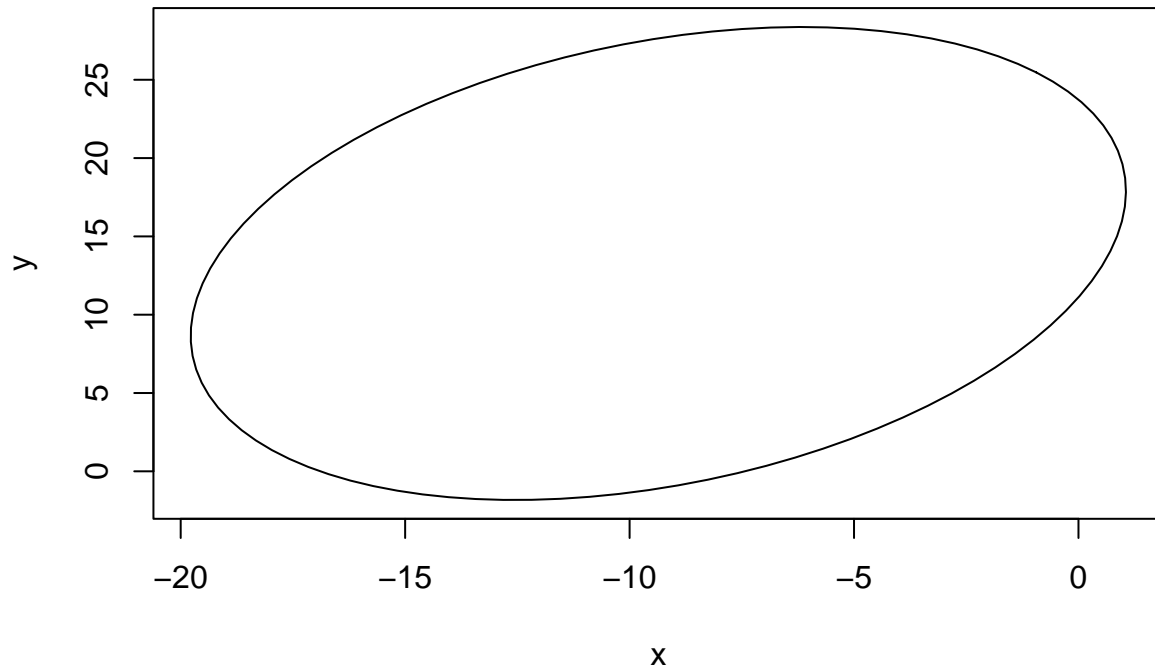
```
library(ellipse)
```

```
## Warning: package 'ellipse' was built under R version 4.1.3
```

```
##
## Attaching package: 'ellipse'
```

```
## The following object is masked from 'package:graphics':
##
##      pairs
```

```
region <- ellipse(S / 11, centre = meanDiff, level = 0.95)
plot(region, type = 'l')
```



Since our value of $\delta = 0$ does not fall into our 95% confidence region, we reject $H_0 = 0$.
6.2.)

```
bonf <- qt(1 - (0.05 / (2 * 2)), n - 1)

BOD_dff_ci <-
  c(meanDiff[1] - bonf * sqrt(S[1, 1] / n), meanDiff[1] + bonf * sqrt(S[1, 1] /
    n))

SS_dff_ci <-
  c(meanDiff[2] - bonf * sqrt(S[2, 2] / n), meanDiff[2] + bonf * sqrt(S[2, 2] /
    n))

cat("The Bonferroni corrected confidence interval for the BOD is ",
    BOD_dff_ci,
    "\n")

## The Bonferroni corrected confidence interval for the BOD is -20.56962 1.849624

cat("The Bonferroni corrected confidence interval for the SS is ",
    SS_dff_ci,
    "\n")
```

```
## The Bonferroni corrected confidence interval for the SS is -2.977472 29.51747
```

Since $\delta = 0$ falls in both of these simultaneous Bonferroni corrected confidence intervals, we fail to reject $H_0 = 0$ in this case. We can see when comparing these intervals to the example that the bonferroni intervals are more narrow.

6.3.)

```
#Reduce the dataframe by row 8
effluent_red <- effluent[-8, ]

#Now find a new matrix with the differences in columns
effluent_redDiff <-
  as.data.frame(cbind(effluent_red[, 1] - effluent_red[, 3], effluent_red[, 2] -
    effluent_red[, 4]))

meanDiff_red = colMeans(effluent_redDiff)
S_red <- cov(effluent_redDiff)
n_red <- nrow(effluent_red)

#Critical Region
T2 = nrow(effluent_red) * t(meanDiff_red - c(0, 0)) %*% solve(S_red) %*% (meanDiff_red - c(0, 0))

#Reduced Bonferroni
bonf_red <- qt(1 - (0.05 / (2 * 2)), nrow(effluent_red) - 1)

BOD_dff_ci_red <-
  c(
    meanDiff_red[1] - bonf_red * sqrt(S_red[1, 1] / n_red),
    meanDiff_red[1] + bonf_red * sqrt(S_red[1, 1] /
      n_red)
  )

SS_dff_ci_red <-
  c(
    meanDiff_red[2] - bonf_red * sqrt(S_red[2, 2] / n_red),
    meanDiff_red[2] + bonf_red * sqrt(S_red[2, 2] / n_red)
  )

cat("The Bonferroni corrected confidence interval for the BOD is ",
  BOD_dff_ci_red,
  "\n")
```

```
## The Bonferroni corrected confidence interval for the BOD is -21.918 -2.082003
```

```
cat("The Bonferroni corrected confidence interval for the SS is ",
  SS_dff_ci_red,
  "\n")
```

```
## The Bonferroni corrected confidence interval for the SS is -3.355587 20.55559
```

With the removal of the outlier, our results now lead us back to rejecting H_0 since our interval for bio-chemical oxygen demand does not contain 0 anymore.

6.4) a.)

```
effluent_ln <- log(effluent)

effluent_lnDiff <-
  as.data.frame(cbind(effluent_ln[, 1] - effluent_ln[, 3], effluent_ln[, 2] -
                      effluent_ln[, 4]))

meanDiff_ln = colMeans(effluent_lnDiff)
S_ln <- cov(effluent_lnDiff)
n_ln <- nrow(effluent_ln)

#Example 6.1 Analysis
T2_ln <-
  nrow(effluent_ln) * t(meanDiff_ln - c(0, 0)) %*% solve(S_ln) %*% (meanDiff_ln - c(0, 0))
#From textbook pg 277
crit_val <- 9.47
(T2_ln > crit_val)

##      [,1]
## [1,] TRUE
```

We reject H_0 since T^2 of the transformed data is greater than our critical value.

b.)

```
BOD_dff_ci <-
  c(meanDiff_ln[1] - bonf * sqrt(S_ln[1, 1] / n),
    meanDiff_ln[1] + bonf * sqrt(S_ln[1, 1] /
                                n))

SS_dff_ci <-
  c(meanDiff_ln[2] - bonf * sqrt(S_ln[2, 2] / n),
    meanDiff_ln[2] + bonf * sqrt(S_ln[2, 2] / n))

cat("The Bonferroni corrected confidence interval for the BOD is ",
    BOD_dff_ci,
    "\n")

## The Bonferroni corrected confidence interval for the BOD is  -1.094488 -0.02190232

cat("The Bonferroni corrected confidence interval for the SS is ",
    SS_dff_ci,
    "\n")

## The Bonferroni corrected confidence interval for the SS is  -0.04498455 0.6360411
```

c.) We can see that from above that we get confidence intervals that conflict with our original conclusions. We have possibly violated the assumptions of the bivariate normal by creating a pairwise difference of variables that may not be normal after transformation, and yet treated as if they were normal.

6.5.) a.)

```

C <- matrix(c(1,-1, 0, 0, 1, -1),
            nrow = 2,
            ncol = 3,
            byrow = TRUE)
C

##      [,1] [,2] [,3]
## [1,]    1  -1    0
## [2,]    0   1  -1

xbar <- c(46.1, 57.3, 50.4)
S <-
  matrix(c(101.3, 63.0, 71.0 , 63.0, 80.2, 55.6, 71.0, 55.6, 97.4),
        3,
        3,
        byrow = TRUE)
T2_C <-
  n * t(C %*% xbar) %*% (solve(C %*% S %*% t(C))) %*% (C %*% xbar)
crit_val_C <-
  (((40 - 1) * (3 - 1)) / (40 - 3 + 1)) * qf(0.05, 3 - 1, 40 - 3 + 1, lower.tail = FALSE)
T2_C > crit_val_C

##      [,1]
## [1,] TRUE

```

Since our T^2 value is greater than our critical value, we reject H_0 , concluding that there are treatment effects.

b.)

```

c(
  C[1,] %*% xbar - sqrt(crit_val_C) * sqrt(t(C[1,]) %*% S %*% C[1,] / 40) ,
  C[1,] %*% xbar + sqrt(crit_val_C) * sqrt(t(C[1,]) %*% S %*% C[1,] / 40)
)

## [1] -14.239955 -8.160045

```

```

c(
  C[2,] %*% xbar - sqrt(crit_val_C) * sqrt(t(C[2,]) %*% S %*% C[2,] / 40) ,
  C[2,] %*% xbar + sqrt(crit_val_C) * sqrt(t(C[2,]) %*% S %*% C[2,] / 40)
)

## [1]  3.5749 10.2251

```

```

c(
  c(1, 0,-1) %*% xbar - sqrt(crit_val_C) * sqrt(t(c(1, 0,-1)) %*% S %*% c(1, 0,-1) / 40),
  c(1, 0,-1) %*% xbar + sqrt(crit_val_C) * sqrt(t(c(1, 0,-1)) %*% S %*% c(1, 0,-1) / 40)
)

## [1] -7.372644 -1.227356

```

6.6.) a.)

```

treat2_1 <- c(3, 1, 2)
treat2_2 <- c(3, 6, 3)

S_t2 <-
  matrix(c(
    var(treat2_1),
    cov(treat2_1, treat2_2),
    cov(treat2_1, treat2_2),
    var(treat2_2)
  ), nrow = 2, ncol = 2)

#S_t2 <- length(treat2_1) * S_t2

treat3_1 <- c(2, 5, 3, 2)
treat3_2 <- c(3, 1, 1, 3)

S_t3 <-
  matrix(c(
    var(treat3_1),
    cov(treat3_1, treat3_2),
    cov(treat3_1, treat3_2),
    var(treat3_2)
  ), nrow = 2, ncol = 2)

#S_t3 <- length(treat3_1) * S_t3

n1 <- length(treat2_1)
n2 <- length(treat3_1)

Spooled <-
  (((n1 - 1) / (n1 + n2 - 2)) * (S_t2)) + (((n2 - 1) / (n1 + n2 - 2)) * (S_t3))
Spooled

##      [,1] [,2]
## [1,]  1.6 -1.4
## [2,] -1.4  2.0

b.)

meanDiff_pool <-
  c(mean(treat2_1), mean(treat2_2)) - c(mean(treat3_1), mean(treat3_2))

T2_pool <-
  t(meanDiff_pool) %*% solve((1 / 3 + 1 / 4) * Spooled) %*% meanDiff_pool
T2_pool

##      [,1]
## [1,] 3.870968

```

```
crit_val_pool <-
  (((n1 + n2 - 2) * 2) / (n1 + n2 - 2 - 1)) * qf(0.01, 2, n1 + n2 - 2 - 1, lower.tail = FALSE)

T2_pool > crit_val_pool
```

```
##      [,1]
## [1,] FALSE
```

We fail to reject $H_0 : \mu_2 - \mu_3 = 0$ at $\alpha = 0.01$.

c.)

```
#mu21 - mu31
c((mean(treat2_1) - mean(treat3_1)) - sqrt(crit_val_pool) * sqrt((1 / 3 + 1 /
  4) * Spooled[1, 1]),
  (mean(treat2_1) - mean(treat3_1)) + sqrt(crit_val_pool) * sqrt((1 / 3 + 1 /
  4) * Spooled[1, 1])
)
```

```
## [1] -7.480741  5.480741
```

```
#mu22 - mu32
c((mean(treat2_2) - mean(treat3_2)) - sqrt(crit_val_pool) * sqrt((1 / 3 + 1 /
  4) * Spooled[2, 2]),
  (mean(treat2_2) - mean(treat3_2)) + sqrt(crit_val_pool) * sqrt((1 / 3 + 1 /
  4) * Spooled[2, 2])
)
```

```
## [1] -5.245688  9.245688
```