Failed.

Correction: Please redo Question 1 compute the simultaneous confidence interval in Question 3.

Lab1

Naveen (navga709), Sridhar(sriad858), Juan(juado206), Samia(sambu064)

2019-12-02

Contents

Question 1. Test of outliers

Suggestion: redo it without using any packages. Once you have got your mahalanobis distances, a simple call to qchisq() should suffice to get the p-values. Make sure you completely understand Result 4.7 in the text book before coding.

a

```
dataset <- read.table("T1-9.dat")

mean <- colMeans(dataset[, 2:8])
Sx <- cov(dataset[, 2:8])
D2M <- mahalanobis(dataset[, 2:8], mean, Sx)
```

To answer this question properly we need to calculate again, as we did on the previous lab assignment, the Mahalanobis vector, which can be easily done.

```
       ARG        AUS        AUT        BEL        BER        BRA
 6.6500027  3.9356705  4.3680848  2.4623238  7.6305669  2.1885513
       CAN        CHI        CHN        COL        COK        CRC
 3.0916940  4.1542602  3.0167240  5.3994790 19.8340006  4.4647286
       CZE        DEN        DOM        FIN        FRA        GER
10.9014563  1.3174764  4.7284159  2.7556601  5.0862790  3.4467449
       GBR        GRE        GUA        HUN        INA        IND
 3.5572682  9.5403223  3.6846053  1.7086674  4.7128021  4.2129432
       IRL        ISR        ITA        JPN        KEN       KORS
 5.4848378  2.5421901  1.4553927  3.2695339  7.6254662  8.0349184
      KORN        LUX        MAS        MRI        MEX        MYA
26.1671415 11.1088463  8.2371487  6.6649850 14.2309322  2.0429608
       NED        NZL        NOR        PNG        PHI        POL
 5.9522081  1.3505689  6.8880631 30.5072477  9.0658836  1.3311122
       POR        ROM        RUS        SAM        SIN        ESP
 2.3695737  6.3491309  3.7506462 35.0140631  8.0163948  1.4160474
       SWE        SUI        TPE        THA        TUR        USA
 0.7846063  3.2910927 10.1839962  7.8152191  8.0453684  9.1556967
```

Once this have been done, we can proceed to run the chi-square test on this Mahalanobis data, which returns this data:

```
chisq.test(D2M)
```

The degree of freedom of the chi-square distribution should be 7 instead. See Result 4.7 in the text book. So you are using the function wrong.

```

    Chi-squared test for given probabilities

data:  D2M
X-squared = 364.07, df = 53, p-value < 2.2e-16
```

After that, and thanks to the outliers package (https://cran.rproject. org/web/packages/outliers/index.html) the outliers detection become considerably easier, proceeding as follows:

```
setNames(scores(D2M, type="chisq", prob = 0.999), dataset[, 1])
```

You really don't need to use and should not use any complicated function/packages to process the p-values. Again, please read carefully what Result 4.7 says.

```
  ARG   AUS   AUT   BEL   BER   BRA   CAN   CHI   CHN   COL   COK   CRC
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
  CZE   DEN   DOM   FIN   FRA   GER   GBR   GRE   GUA   HUN   INA   IND
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
  IRL   ISR   ITA   JPN   KEN  KORS  KORN   LUX   MAS   MRI   MEX   MYA
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
  NED   NZL   NOR   PNG   PHI   POL   POR   ROM   RUS   SAM   SIN   ESP
FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
```

```
  SWE   SUI   TPE   THA   TUR   USA
FALSE FALSE FALSE FALSE FALSE FALSE
```

This output on the console shows, as a result of the setNames function, the country's acronym over a TRUE/FALSE indicator, which highlights the observations whose Mahalanobis distance lies beyond the indicated percentile.

In this first case, the function works with the given significance of a=0.001. Analyzing the table above, it can be noticed how only two countries are being identified as outliers. Those are: Papua New Guinea (PNG, 30.5072477) and Samoa (SAM, 35.0140631), which are the two most extreme cases amongst all the calculated distances.

The study could be finished here but seems effective to go a bit deeper and try to modify this quantile through some a modifications in order to find a better differentiation between outliers and average observations. More specifically, the Bonferroni correction enables the calibration of this parameter. Supposing that, for our dataset, each country would be compared once with each other country on the list, the number of total combinations comes to 1431. As a consequence, the chosen correction method results in a new a value of 1 - 0.761 = 0.239. Running again the scores function with this new parameter returns the following:

```
setNames(scores(D2M, type="chisq", prob = 0.761), dataset[, 1])
```

```
  ARG   AUS   AUT   BEL   BER   BRA   CAN   CHI   CHN   COL   COK   CRC
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
  CZE   DEN   DOM   FIN   FRA   GER   GBR   GRE   GUA   HUN   INA   IND
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
  IRL   ISR   ITA   JPN   KEN  KORS  KORN   LUX   MAS   MRI   MEX   MYA
FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
  NED   NZL   NOR   PNG   PHI   POL   POR   ROM   RUS   SAM   SIN   ESP
FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
  SWE   SUI   TPE   THA   TUR   USA
FALSE FALSE FALSE FALSE FALSE FALSE
```

This is a misconception. Make sure you understand Result 4.7 first; once you do, your first intuition should most likely tell you that you want to divide alpha/2 by the number of data points, if you want to do a Bonferroni.

The question asks whether this intuitive urge of doing a Bonferroni is justified. The answer, in fact, is no. If one really do understand what Result 4.7 really means, they can see that the false discovery rate of the outlier test is controlled at about 100*alpha%, and this rate of making error is independent of the number of data point there are.

With this new limit, two new countries are identified as outliers: Cook Islands (COK, 19.8340006) and North Korea (KORN, 26.1671415). Those two values are far from being standard, therefore can be concluded that the correction is accurate. Beyond this conclusion, alpha could be even more reduced, depending on where we want to place the "outlier threshold". As a last example, setting a to 0.35 makes also Mexico (MEX, 14.2309322) an outlier:

```
setNames(scores(D2M, type="chisq", prob = 0.65), dataset[, 1])
```

```
  ARG   AUS   AUT   BEL   BER   BRA   CAN   CHI   CHN   COL   COK   CRC
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
  CZE   DEN   DOM   FIN   FRA   GER   GBR   GRE   GUA   HUN   INA   IND
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
  IRL   ISR   ITA   JPN   KEN  KORS  KORN   LUX   MAS   MRI   MEX   MYA
FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE
  NED   NZL   NOR   PNG   PHI   POL   POR   ROM   RUS   SAM   SIN   ESP
FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
  SWE   SUI   TPE   THA   TUR   USA
FALSE FALSE FALSE FALSE FALSE FALSE
```

b

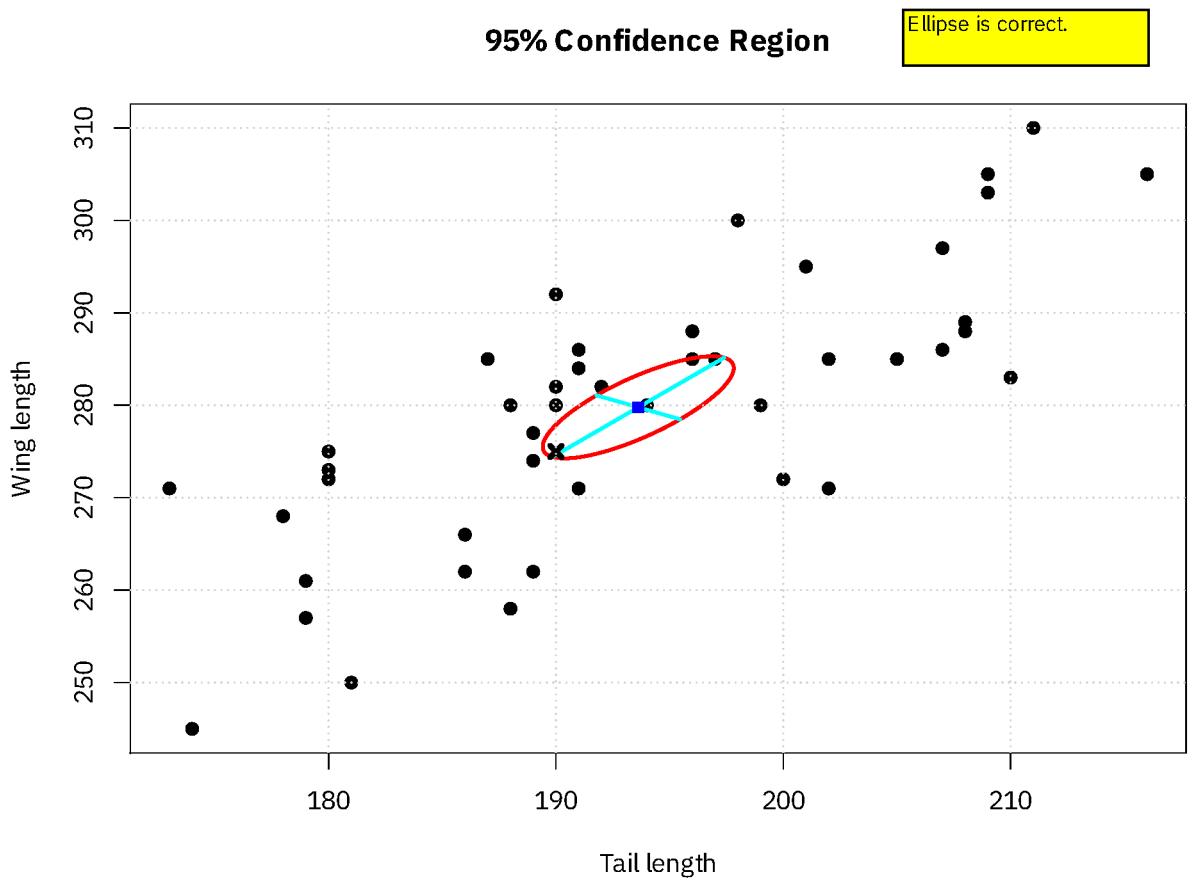
The reason for this is that Euclidean distance simply computes the ordinary straight line between two points and the Mahalanobis distance takes the covariate into account, which leads to elliptic decision boundaries in 2D, therefore narrower than the Eucliden circular, which helps to properly identify the outliers, as done in the previous section

Question 2. Test, confidence region and confidence intervals for a mean vector

Look at the bird data in file T5-12.dat and solve Exercise 5:20 of Johnson, Wichern. Do not use any extra R package or built{in test but code all required matrix calculations. You MAYNOT use loops!

a

```
bird_data <- read.table("T5-12.dat")
mu <- c(190, 275) #given mu values
x_bar <- colMeans(bird_data)

calcReqVals = function(data){
  n <- nrow(data)
  p <- ncol(data)
  conf <- 0.05
  S <- cov(data)
  eig <- eigen(S)
  x_ <- colMeans(data)

  quantile <- qf(1 - conf, df1=p, df2=n - p)
  scale <- sqrt(eig$values * p * (n - 1) * quantile / (n * (n - p)))
  scaled <- eig$vector %*% diag(scale) # scale eigenvectors to length = square-root
  xMat <- rbind(x_[1] + scaled[1, ], x_[1] - scaled[1, ])
  yMat <- rbind(x_[2] + scaled[2, ], x_[2] - scaled[2, ])

  angles <- seq(0, 2 * pi, length.out=200)
  ellBase <- cbind(scale[1]*cos(angles), scale[2]*sin(angles)) # making a circle base...
  ellax <- eig$vector %*% t(ellBase)

  return(list("ellax"=ellax, "xMat"=xMat, "yMat"=yMat))
}

out = calcReqVals(bird_data)

#Plotting the confidence region
plot(bird_data,pch=19, xlab="Tail length",
     ylab="Wing length", main="95% Confidence Region")
lines((out$ellax + x_bar)[1, ], (out$ellax + x_bar)[2, ], asp=1, type="l", lwd=2, col="red")
matlines(out$xMat, out$yMat, lty=1, lwd=2, col="cyan") #
points(mu[1], mu[2], pch=4, col="black", lwd=3)
grid()
points(x_bar[1],x_bar[2], type="p", col="blue", pch=15)
```



Since the known mean of the male data lies in the confidence interval for the mean of the female data, we are fail to reject the hypothesis that the male and the female have the same mean.

We took most of the above code with understanding from stackoverflow: https://stats.stackexchange.com/questions/9898/how-to-plot-an-ellipse-from-eigenvalues-and-eigenvectors-in-r

We also asked for few help from Maria with respect to code and theory.

It is highly appreciated to cite external helps you received. Please keep doing so in the future! :)

b

```
tsq95_intervals <- function(data) {
  conf = 0.05
  n <- nrow(data)
  p <- ncol(data)
  x_bar <- colMeans(data)
  S <- cov(data)
  offset <- sqrt(p * (n - 1) * qf(1 - conf, df1=p, df2=n - p) / (n - p) * diag(S) / n)
  rbind(x_bar - offset, x_bar + offset)
}

bon95_intervals <- function(data) {
  conf = 0.05
  n <- nrow(data)
```

```r
  p <- ncol(data)
  x_bar <- colMeans(data)
  S <- cov(data)
  offset <- sqrt(diag(S) / n) * qt(1 - conf / (2 * p), df=n - 1)
  rbind(x_bar - offset, x_bar + offset)
}

cat("\n95% T-square interval\n")
```

```
95% T-square interval
```

```r
tsq95_intervals(bird_data)
```

```
          V1       V2
[1,] 189.4217 274.2564
[2,] 197.8227 285.2992
```

Correct.

```r
cat("\n95% Bonferroni interval\n")
```

```
95% Bonferroni interval
```

```r
bon95_intervals(bird_data)
```

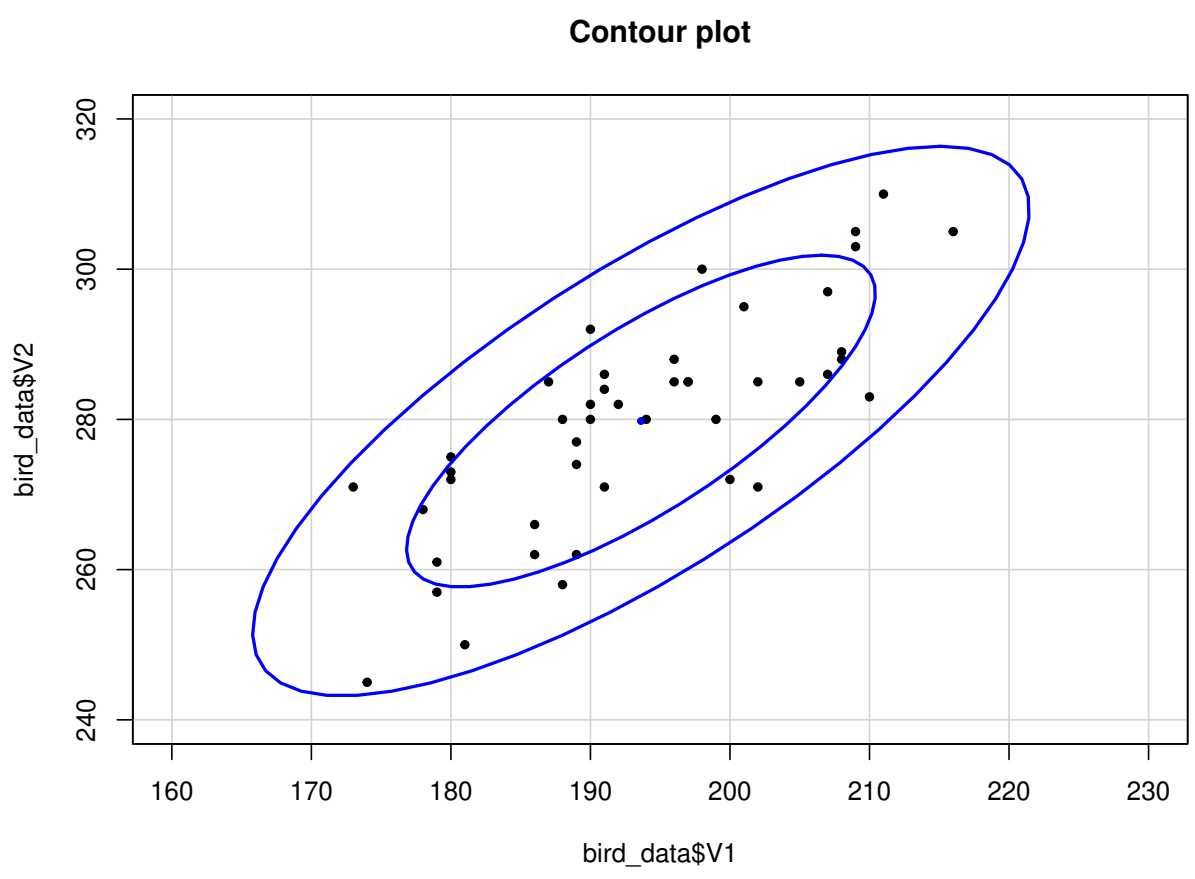
```
          V1       V2
[1,] 189.8216 274.7819
[2,] 197.4229 284.7736
```

Correct.

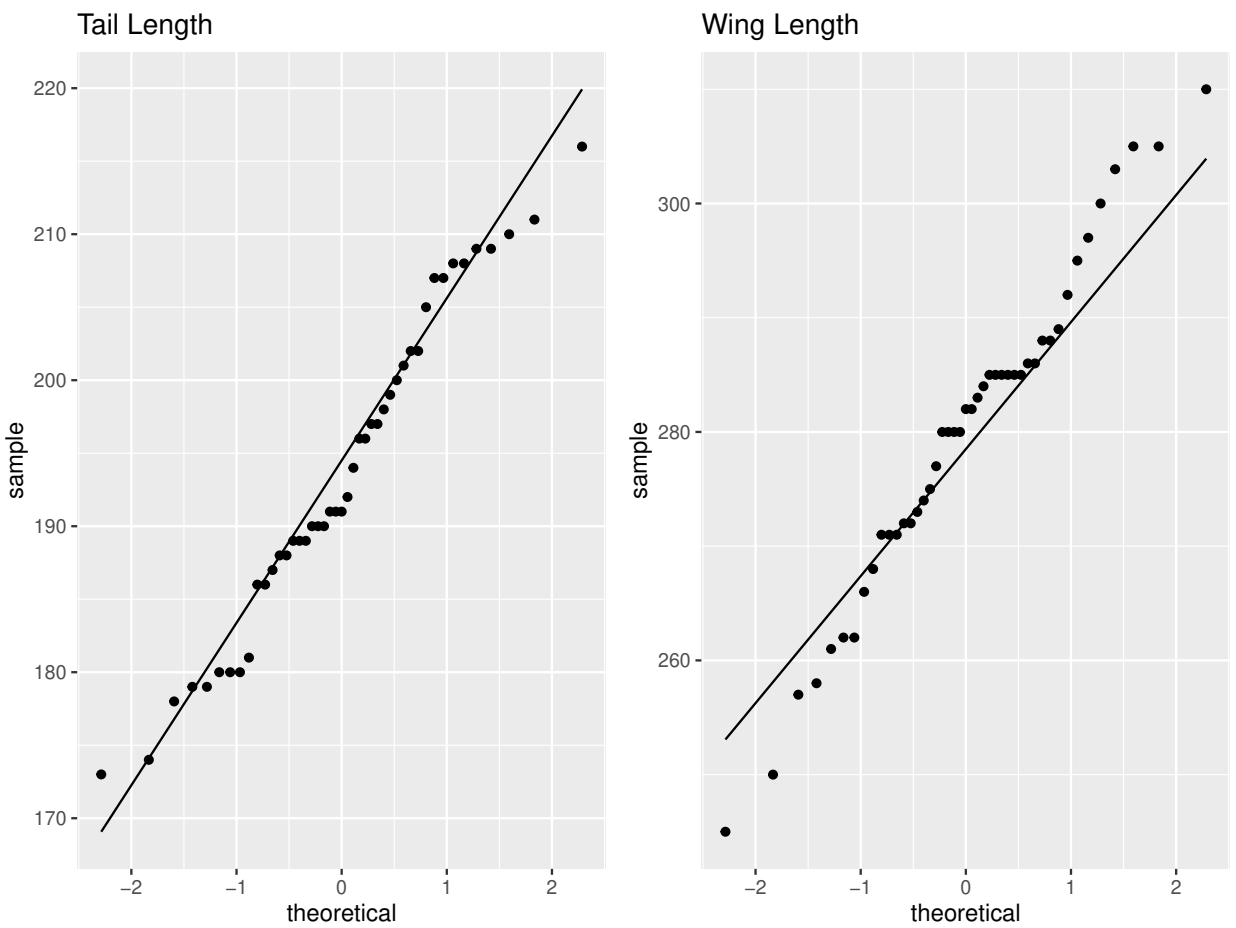
The only advantage we could find for T-squared intervals over Bonferroni interval is that, T-squared intervals account for the correlation between variates, while Boneferroni does not. Benferroni us useful when we have to find the CI for the individual component.

c

```r
dataEllipse(x=bird_data$V1, y=bird_data$V2, pch=20, levels=c(0.68, 0.95),
            xlim=c(160, 230), ylim=c(240, 320), center.cex=0.5, main="Contour plot")
```



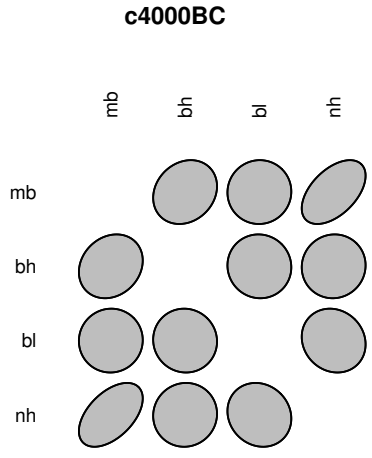

```
grid.arrange(ggplot(data = bird_data, aes(sample = V1)) +
                  stat_qq() + stat_qq_line() + ggtitle("Tail Length"),
                  ggplot(data = bird_data, aes(sample = V2)) +
                  stat_qq() + stat_qq_line() +
                  ggtitle("Wing Length"),ncol=2)
```

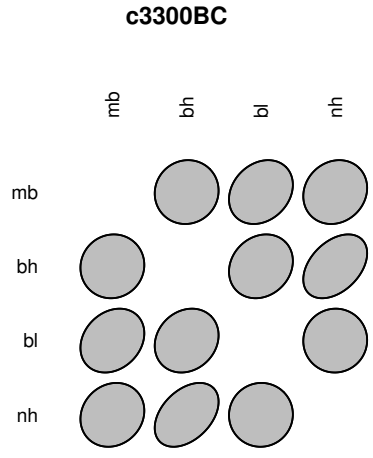


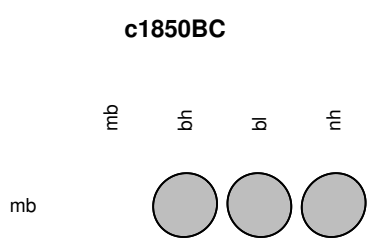
Since the data points go along the line in the qqplots, it shows that the data is normally distributed. A bivariate normal distribution would be a viable population model.

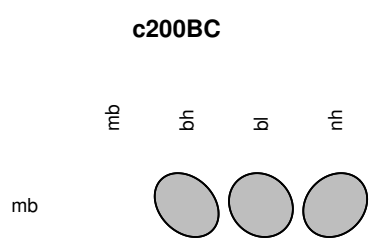
Question 3. Comparison of mean vectors (one{way MANOVA)

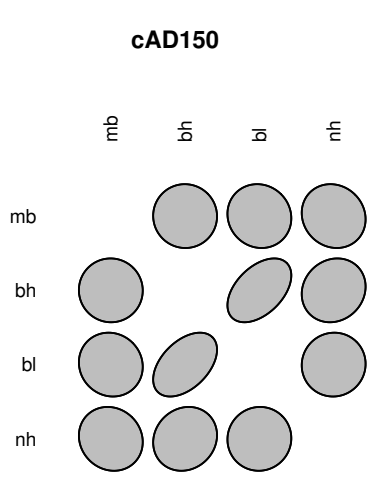
a



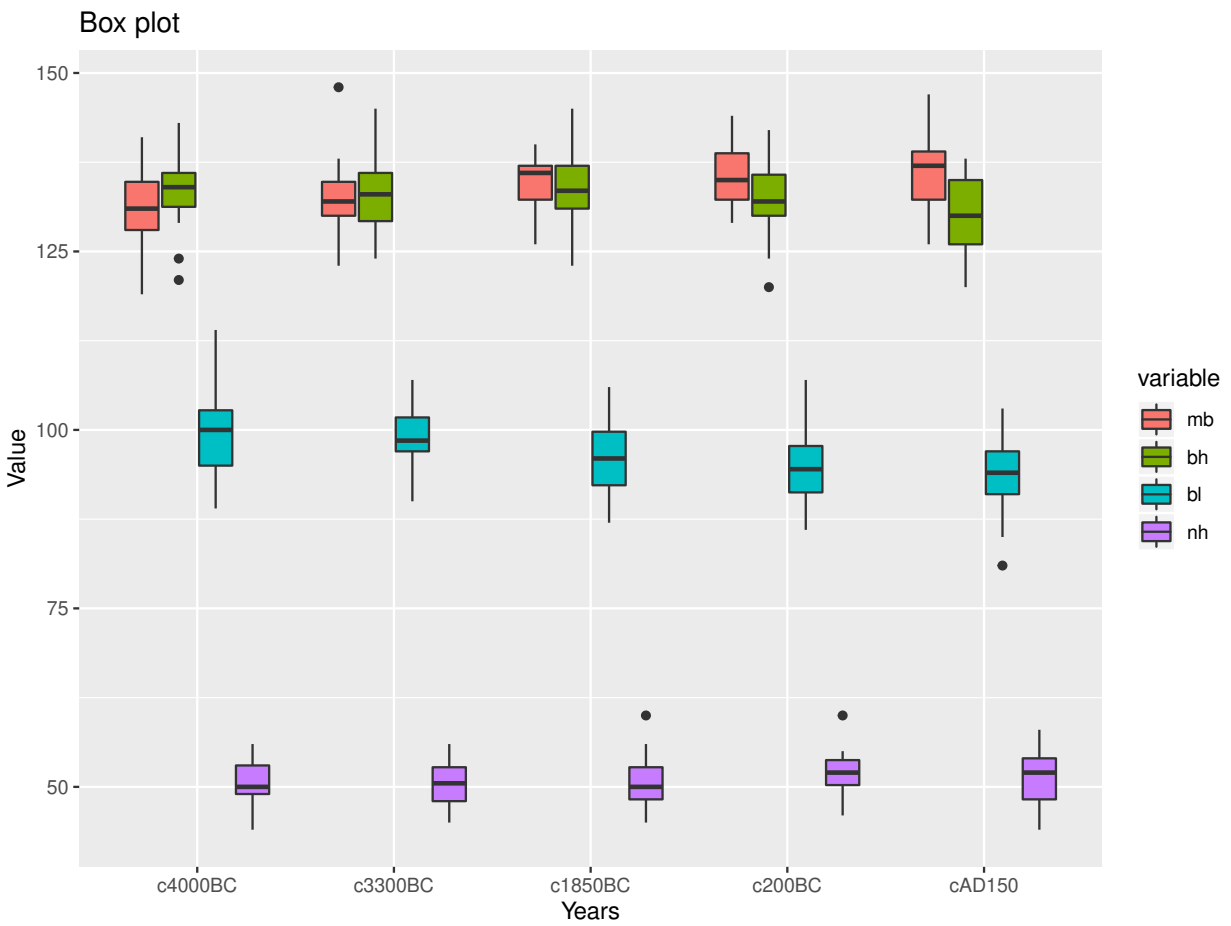








From above correlation plot over 5 time period the common thing which we noticed is the postive correlation between basibregmatic height and basialiveolar length. Moreover, the correlation become more stronger between them over the period. In AD 150 the correlation between nh and mb was negative which was unprecedented.



From this the only thing which is informative is that the mean bl length decreased over the years while rest of the mean measurement did not follow any pattern. Also the distance between mean bh and mb increased towards the end of AD150. As maximal breadth of the skull increased , the height of the skull increased as well. Were brains were increasing in size ?

b

Table 1: Mean of covariates in different groups

Group	mb	bh	bl	nh
c4000BC	131.3667	133.6000	99.16667	50.53333
c3300BC	132.3667	132.7000	99.06667	50.23333
c1850BC	134.4667	133.8000	96.03333	50.56667
c200BC	135.5000	132.3000	94.53333	51.96667
cAD150	136.1667	130.3333	93.50000	51.36667

```
manova_fit <- manova(cbind(mb, bh, bl, nh) ~ sk$epoch, sk)
summary(manova_fit,test="Pillai")
```

Alright.

```
           Df  Pillai approx F num Df den Df    Pr(>F)
sk$epoch    4 0.35331    3.512     16    580 4.675e-06 ***
Residuals 145
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Pillai's trace is used as a test statistic in MANOVA. It is a positive valued statistic ranging from 0 to 1. Increasing values means that effects are contributing more to the model; you should reject the null hypothesis for large values.

Pillai's trace is considered to be the most powerful and robust statistic for general use, especially for departures from assumptions – Wikipedia

Here the p value is less than 0.05 so we reject the null hypothesis which is ,that the vector means are same across group.

c

```r
t2_interval <- function(data, conf) {
n <- nrow(data)
p <- ncol(data)
x_ <- colMeans(data)
S <- cov(data)
offset <- sqrt(p * (n - 1) * qf(1 - conf, df1=p, df2=n - p) / (n - p) * diag(S) / n)
rbind(x_ - offset, x_ + offset)
}

conf_interval <-  t2_interval(sk[,2:5],0.05)
```

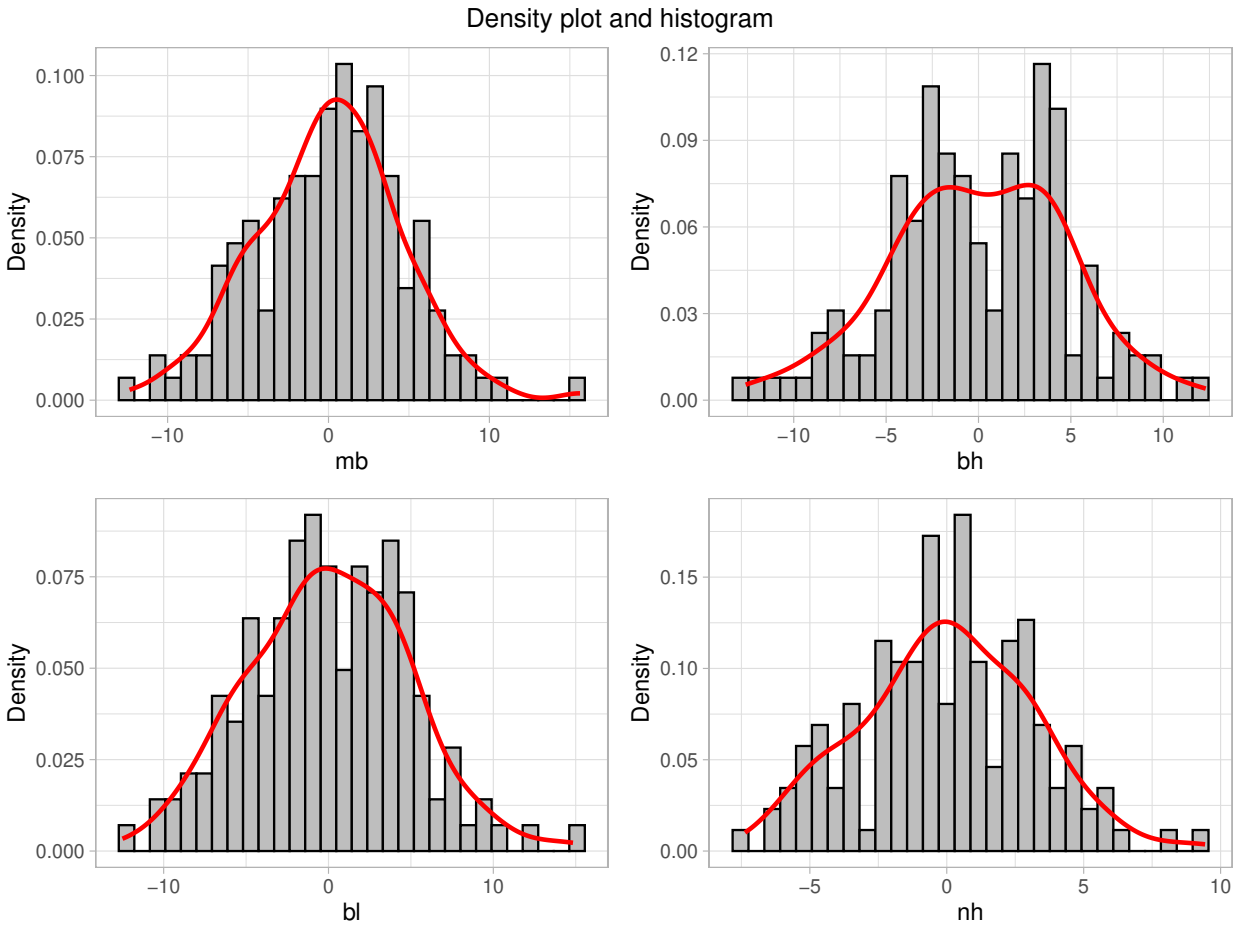
Table 2: 95% CI

mb	bh	bl	nh
132.7147	131.2755	95.076	50.10776
135.2320	133.8178	97.844	51.75890

You have misunderstood the question.

The question means that you should compute the interval described in Result 6.5.

Hint: The diagonal of the W matrix is the same as the "Residuals" column given by `print(manova(...))`



The variables are nearly normally distributed.

Appendix

```r
library(outliers)
library(heplots)
library(kableExtra)
library(gridExtra)
library(ggplot2)
library(reshape2)
library(ellipse)
library(car)
knitr::opts_chunk$set(
    message = FALSE,
    warning = FALSE,
    comment = NA,
    fig.width=8,
    fig.height=6
)
dataset <- read.table("T1-9.dat")

mean <- colMeans(dataset[, 2:8])
Sx <- cov(dataset[, 2:8])
D2M <- mahalanobis(dataset[, 2:8], mean, Sx)
setNames(D2M, dataset[, 1])
chisq.test(D2M)
setNames(scores(D2M, type="chisq", prob = 0.999), dataset[, 1])
setNames(scores(D2M, type="chisq", prob = 0.761), dataset[, 1])

setNames(scores(D2M, type="chisq", prob = 0.65), dataset[, 1])
bird_data <- read.table("T5-12.dat")
mu <- c(190, 275) #given mu values
x_bar <- colMeans(bird_data)

calcReqVals = function(data){
  n <- nrow(data)
  p <- ncol(data)
  conf <- 0.05
  S <- cov(data)
  eig <- eigen(S)
  x_ <- colMeans(data)

  quantile <- qf(1 - conf, df1=p, df2=n - p)
  scale <- sqrt(eig$values * p * (n - 1) * quantile / (n * (n - p)))
  scaled <- eig$vector %*% diag(scale) # scale eigenvectors to length = square-root
  xMat <- rbind(x_[1] + scaled[1, ], x_[1] - scaled[1, ])
  yMat <- rbind(x_[2] + scaled[2, ], x_[2] - scaled[2, ])

  angles <- seq(0, 2 * pi, length.out=200)
  ellBase <- cbind(scale[1]*cos(angles), scale[2]*sin(angles)) # making a circle base...
  ellax <- eig$vector %*% t(ellBase)

  return(list("ellax"=ellax, "xMat"=xMat, "yMat"=yMat))
}
```

```r
out = calcReqVals(bird_data)

#Plotting the confidence region
plot(bird_data,pch=19, xlab="Tail length",
     ylab="Wing length", main="95% Confidence Region")
lines((out$ellax + x_bar)[1, ], (out$ellax + x_bar)[2, ], asp=1, type="l", lwd=2, col="red")
matlines(out$xMat, out$yMat, lty=1, lwd=2, col="cyan") #
points(mu[1], mu[2], pch=4, col="black", lwd=3)
grid()
points(x_bar[1],x_bar[2], type="p", col="blue", pch=15)
```

```r
tsq95_intervals <- function(data) {
  conf = 0.05
  n <- nrow(data)
  p <- ncol(data)
  x_bar <- colMeans(data)
  S <- cov(data)
  offset <- sqrt(p * (n - 1) * qf(1 - conf, df1=p, df2=n - p) / (n - p) * diag(S) / n)
  rbind(x_bar - offset, x_bar + offset)
}

bon95_intervals <- function(data) {
  conf = 0.05
  n <- nrow(data)
  p <- ncol(data)
  x_bar <- colMeans(data)
  S <- cov(data)
  offset <- sqrt(diag(S) / n) * qt(1 - conf / (2 * p), df=n - 1)
  rbind(x_bar - offset, x_bar + offset)
}

cat("\n95% T-square interval\n")
tsq95_intervals(bird_data)

cat("\n95% Bonferroni interval\n")
bon95_intervals(bird_data)

dataEllipse(x=bird_data$V1, y=bird_data$V2, pch=20, levels=c(0.68, 0.95),
            xlim=c(160, 230), ylim=c(240, 320), center.cex=0.5, main="Contour plot")


grid.arrange(ggplot(data = bird_data, aes(sample = V1)) +
                 stat_qq() + stat_qq_line() + ggtitle("Tail Length"),
                 ggplot(data = bird_data, aes(sample = V2)) +
                 stat_qq() + stat_qq_line() +
                 ggtitle("Wing Length"),ncol=2)
sk <- Skulls
#aggregate(sk["c40,2:5],by=list(sk$epoch), FUN =sd)


corr_mat <- cor(sk[sk$epoch=="c4000BC",2:5])
corr_mat2 <- cor(sk[sk$epoch=="c3300BC",2:5])
corr_mat3 <- cor(sk[sk$epoch=="c1850BC",2:5])
corr_mat4 <- cor(sk[sk$epoch=="c200BC",2:5])
```

```r
corr_mat5 <- cor(sk[sk$epoch=="cAD150",2:5])

par(mfrow=c(3,2))

plotcorr(corr_mat,outline=TRUE,diag=FALSE,main="c4000BC")
plotcorr(corr_mat2, diag=FALSE,main="c3300BC")
plotcorr(corr_mat3, diag=FALSE,main ="c1850BC")
plotcorr(corr_mat4, diag=FALSE,main ="c200BC")
plotcorr(corr_mat5, diag=FALSE,main ="cAD150")
```

```r
tall_sk <- reshape2::melt(sk, id="epoch")

ggplot(tall_sk) +
geom_boxplot(aes(x=factor(epoch), y=value, fill=variable)) +
ggtitle("Box plot") + xlab("Years") + ylab("Value")
mean_group <- aggregate(sk[,2:5],by=list(sk$epoch), FUN =mean)
colnames(mean_group)[1] <- "Group"

kable(mean_group,caption = "Mean of covariates in different groups") %>% kable_styling(latex_options = '
manova_fit <- manova(cbind(mb, bh, bl, nh) ~ sk$epoch, sk)
summary(manova_fit,test="Pillai")
t2_interval <- function(data, conf) {
n <- nrow(data)
p <- ncol(data)
x_ <- colMeans(data)
S <- cov(data)
offset <- sqrt(p * (n - 1) * qf(1 - conf, df1=p, df2=n - p) / (n - p) * diag(S) / n)
rbind(x_ - offset, x_ + offset)
}

conf_interval <-  t2_interval(sk[,2:5],0.05)
kable(conf_interval,caption = "95\\% CI") %>%
    kable_styling(latex_options = "hold")
residual <- manova_fit$res %>% as.data.frame()


plot_hist_dens <- function (col_name){
    ggplot(residual,aes_string(col_name) ) +
    geom_histogram(aes(y=..density..),colour="black",fill="gray",bin=10) +
    geom_line(stat="density",colour="red",alpha=1,size=1) + theme_light() +
    xlab(col_name) + ylab("Density")
}
```

```r
p1 <- plot_hist_dens("mb")
p2 <- plot_hist_dens("bh")
p3 <- plot_hist_dens("bl")
p4 <- plot_hist_dens("nh")

grid.arrange(p1, p2, p3, p4, ncol=2,nrow=2,top = "Density plot and histogram")
```