Multivariate Statistical Methods 732A97

Lab 2

Albin Västerlund
albva223

Eric Herwin
Erihe068

Balaji Ramkumar
Balra340

Guilherme Barros
Guiba484

Contents

Question 1: Test of outliers

a)

In this task we will use the Mahalanobis distance to test against a chi–square value at 0.1% significance level.

The output without simultainious testing:

```
## [1] "PNG" "SAM"
```

The output with simultainious testing (α/m):

```
## character(0)
```

For $\alpha = 0.1$ we can note that it is a very low level of significance and we might assume that it is already corrected by Bonferroni, since $0.0001 * 54 = 0.054$ and 0.054 is almost the "standard" $alpha = 0.05$. The output without simultanious testing we do the test with α and since we already concluded that it is corrected with Bonferroni. In the output we get PNG and SAM which is the same conclusion from the previous lab. So correcting with Bonferroni on an already assumed correction will be invalid, and we will have a very low significance level due to α/m and the result will be that we do not get any significant outliers at all. So it is not sensible to use Bonferroni correction in this task.

b)

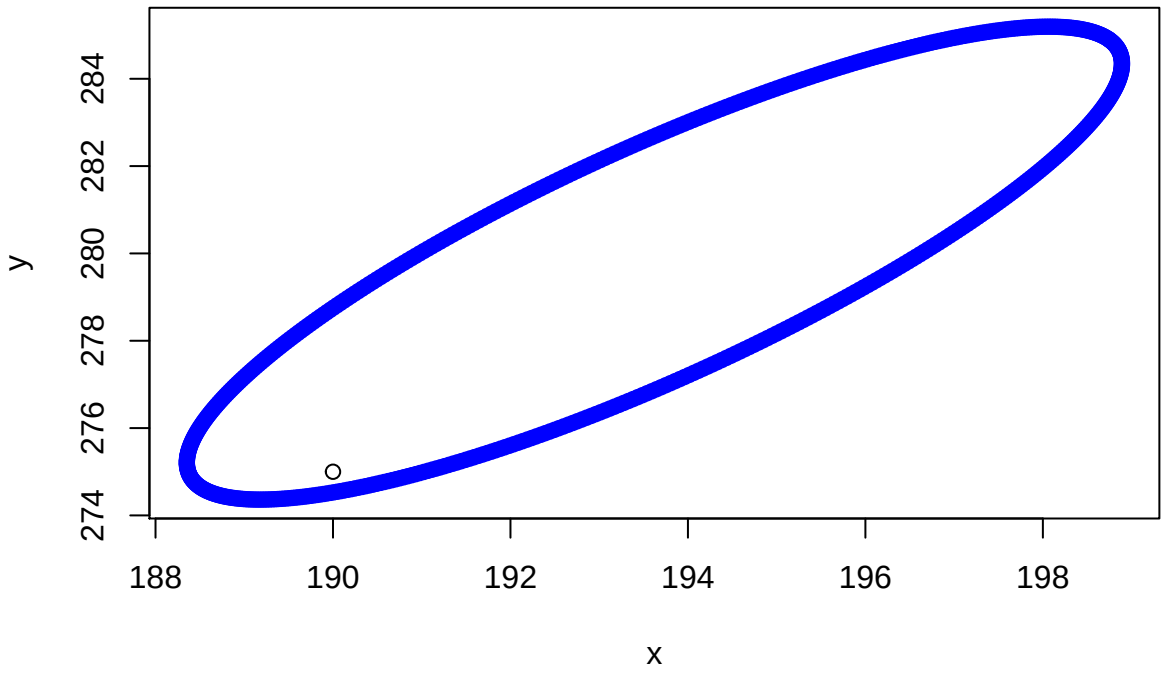
This is because of the variances between the variables are taken in to account in the calculation of the distances. With a country such as North Korea we can assume that they can be better or worse, compared to the other contries, when it comes to Natinonal track record of women. So the country will become an outlier.

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

In this question we will look at bird data and solve the question 5.20.

a)

For tthis step we will find the 95% confidence elipse for the population means μ_1 and mu_2.



We can see that in the mean values for male is inside of the confidence produced elipse. We can conclude that there is no statistical difference between female birds and male birds on 95% confidence level. The amount of observations is enough to make statistcal conclusions, and we are therefore able to say that these values are plausible.

b)

In this task we will construct 95% T^2 and Bonferroni- intervals forμ_1 and μ_2.

The T^2-interval:

```
#for mu_1
print(interval_1)
```

```
## [1] 189.4217 197.8227
```

```
#interval_1[2]-interval_1[1]

#for mu_2
print(interval_2)
```

```
## [1] 274.2564 285.2992
```

```
#interval_2[2]-interval_2[1]
```

The Bonferroni-interval:

```
#for mu_1
print(binterval_1)
```

```
## [1] 189.8216 197.4229
```

```
#binterval_1[2]-binterval_1[1]
```

```r
#for mu_2
print(binterval_2)
```

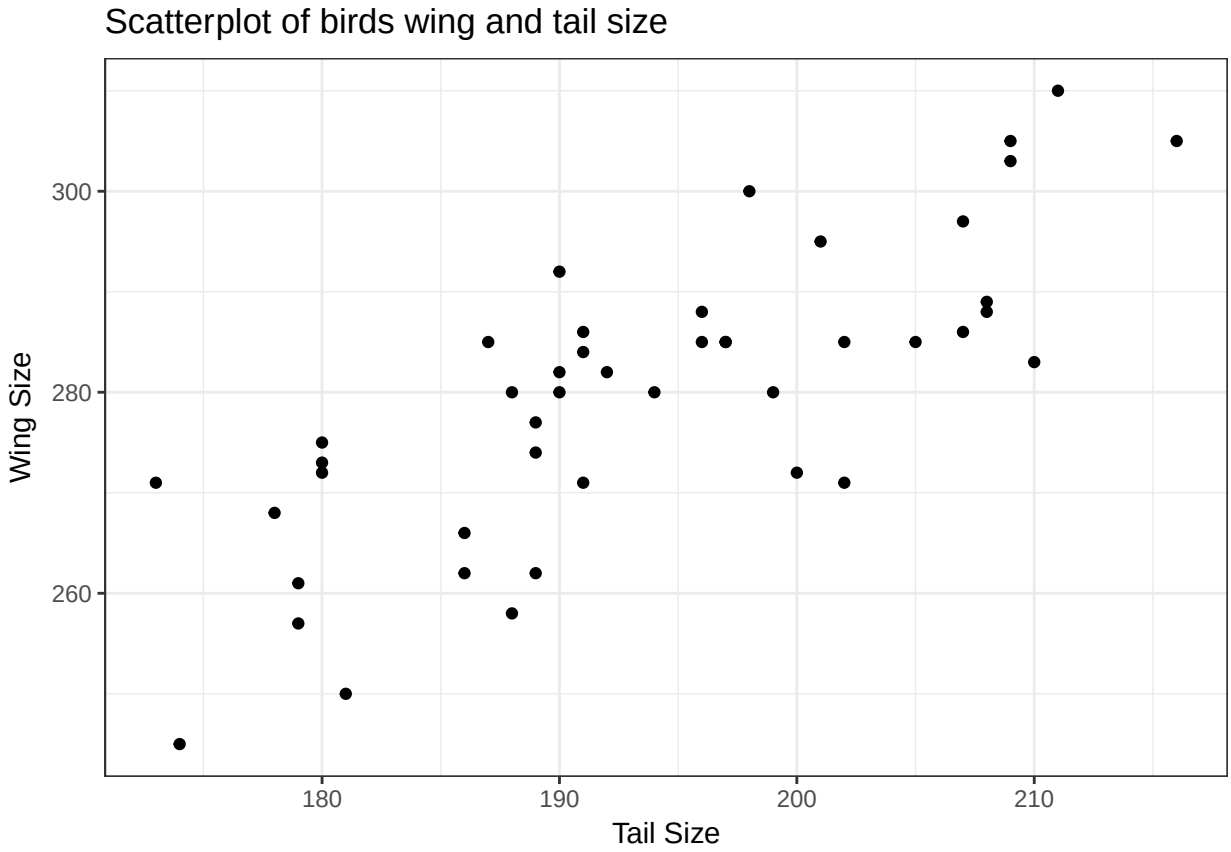
```
## [1] 274.7819 284.7736
```

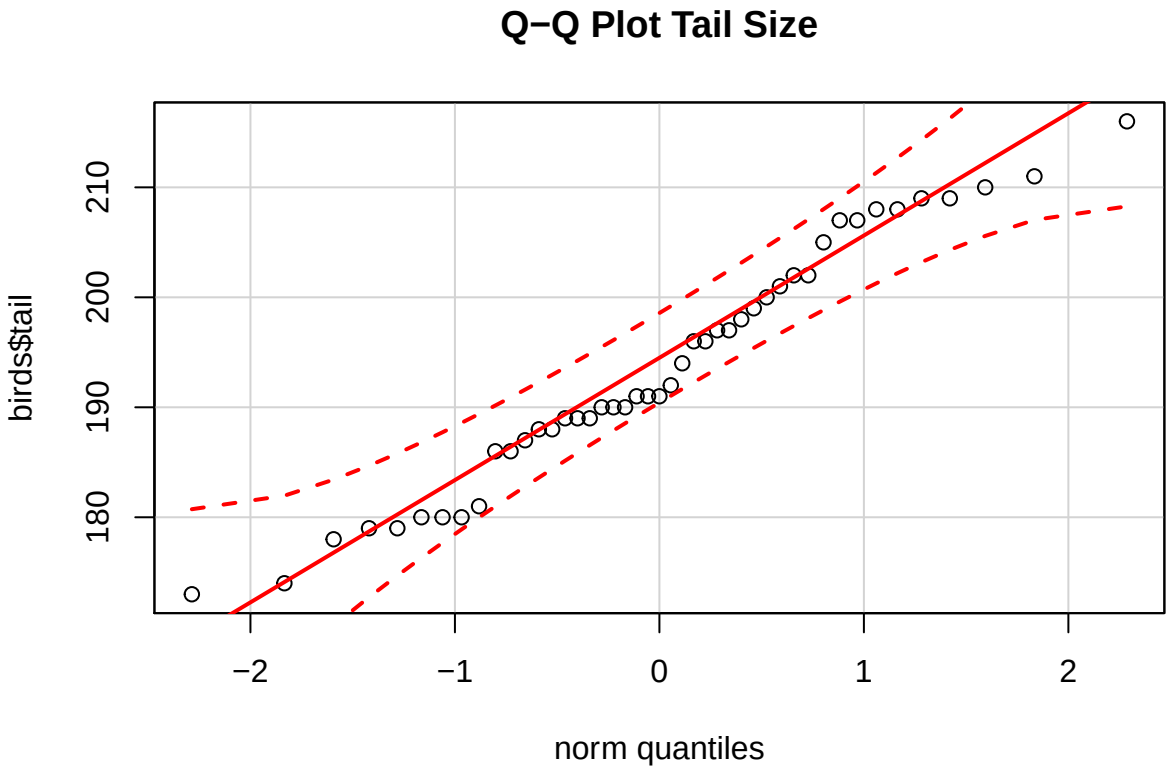
```r
#binterval_2[2]-binterval_2[1]
```

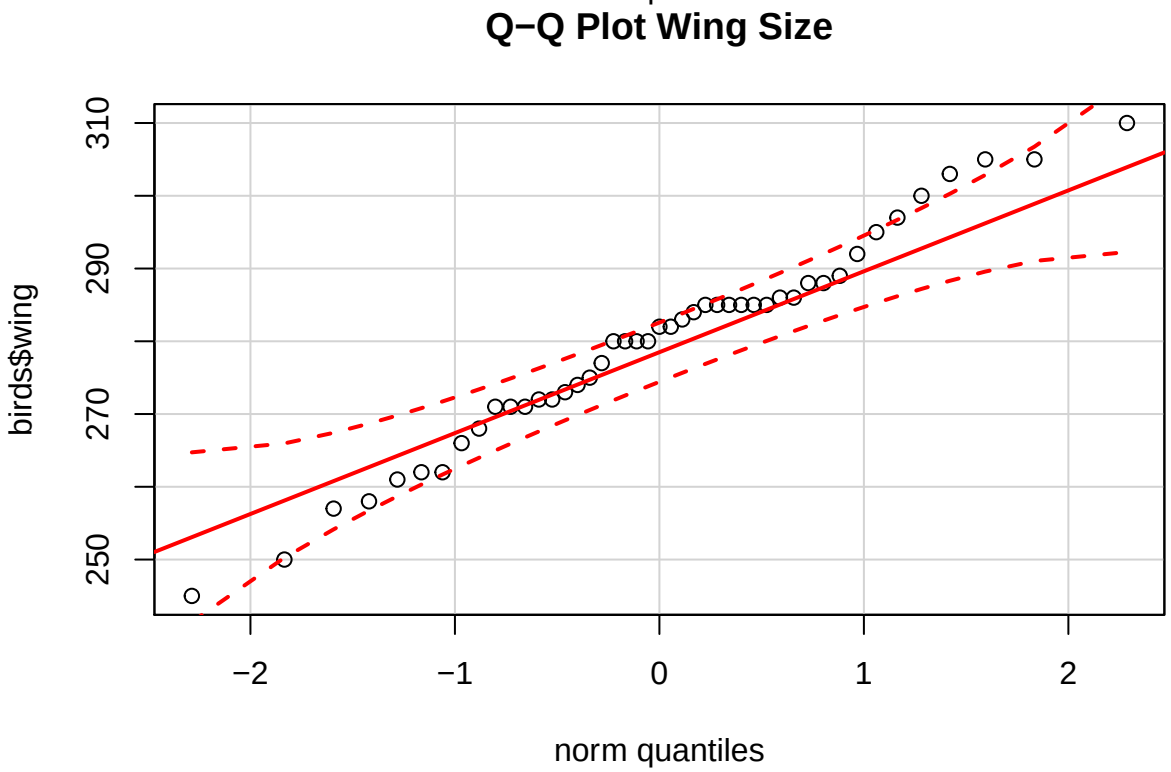
We can see that the Bonferroni-interval is smaller then the T^2-interval. The advantage of T^2 is that you get a more general test of the pair of means you are testing, compared to the Bonferroni. If the T^2 test becomes significant it can be a good idea to look closer on the induvidual pair of means with the Bonferroni.

c)

To investigate normality, we plot both variables and do qqplots:





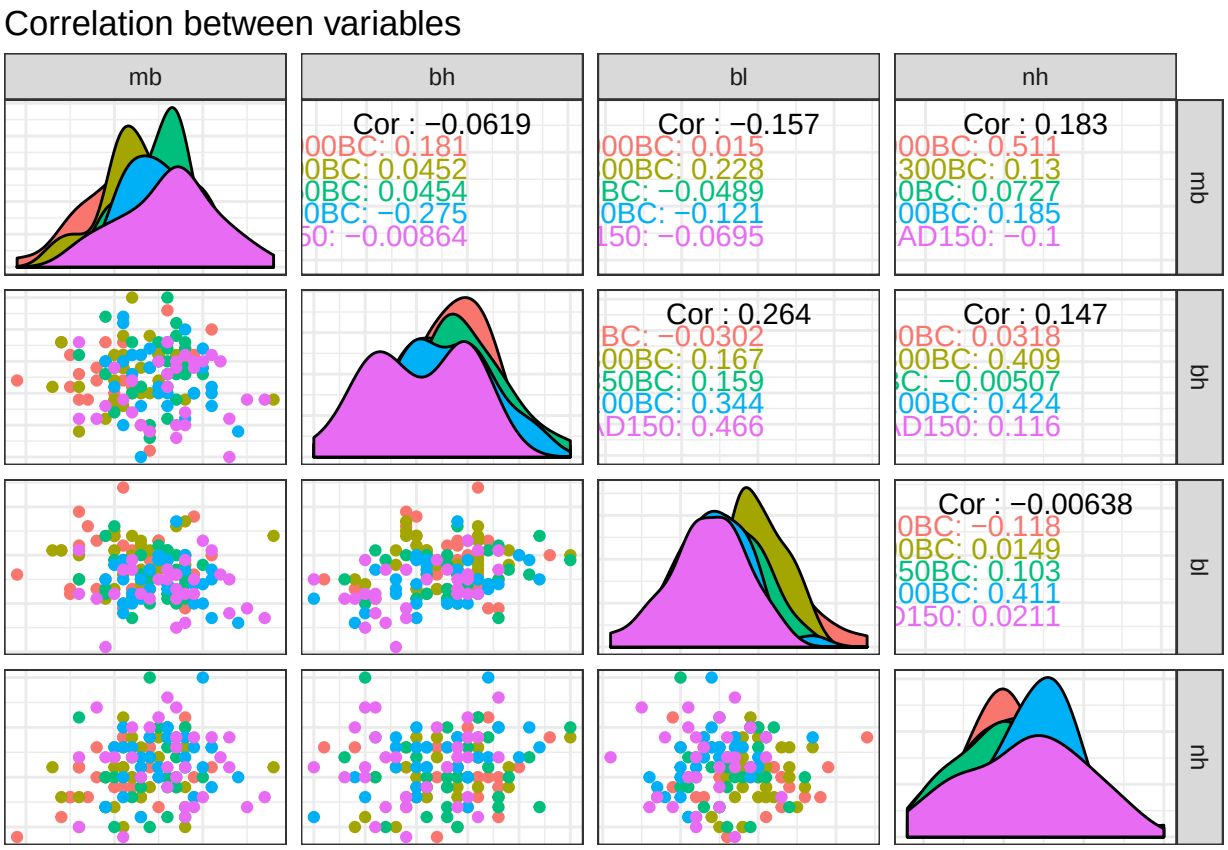


We can see that the Q_Q plots suggest a possible non-normality of the variables, especially in wing size (where some points are outside the confidence band and a lot of the points are far away from the line). In this case, it is possible that we would get better results by applying a transformation to the wing size data.

We can also see that from the scatter plot that it follows an elipse-shape, and therefore indicates that the variables are bivariate normal distributed.

Question 3: Comparison of mean vectors (one-way MANOVA)

a)



By looking at the graph it's hard to tell if the diffrent epoch have differ mean for the diffrent variables.

b)

In this assigment we will test if the mean vector differ between the epochs.

	Overall sample mean
mb	133.97
bh	132.55
bl	96.46
nh	50.93

We start to take out the overall sample mean $\bar{x}$.

variable	epoch	Treatment effect (tau)
mb	c4000BC	-2.61
mb	c3300BC	-1.61
mb	c1850BC	0.49
mb	c200BC	1.53

variable	epoch	Treatment effect (tau)
mb	cAD150	2.19
bh	c4000BC	1.05
bh	c3300BC	0.15
bh	c1850BC	1.25
bh	c200BC	-0.25
bh	cAD150	-2.21
bl	c4000BC	2.71
bl	c3300BC	2.61
bl	c1850BC	-0.43
bl	c200BC	-1.93
bl	cAD150	-2.96
nh	c4000BC	-0.40
nh	c3300BC	-0.70
nh	c1850BC	-0.37
nh	c200BC	1.03
nh	cAD150	0.43

$\tau_l = \bar{x}_l - \bar{x}$ is the mean for each epochs.

We also compute the residuals $\hat{\varepsilon}_{lj} = x_l j - \bar{x}_l$ but when we have 150 of them we dont show them in the report.

B matrix				
	mb	bh	bl	nh
mb	502.83	-228.15	-626.63	135.43
bh	-228.15	229.91	292.28	-66.07
bl	-626.63	292.28	803.29	-180.73
nh	135.43	-66.07	-180.73	61.20

To make the test we need the SST matrix that is computed by this formula: $\mathbf{W} = \sum_{l=1}^{g} n_l(\bar{x}_l - \bar{x})(\bar{x}_l - \bar{x})^t$

W matrix				
	mb	bh	bl	nh
mb	3061.07	5.33	11.47	291.30
bh	5.33	3405.27	754.00	412.53
bl	11.47	754.00	3505.97	164.33
nh	291.30	412.53	164.33	1472.13

We also need the SSE matrix that is computed by this formula: $\mathbf{W} = \sum_{l=1}^{g} \sum_{j=1}^{n_l} (x_{lj} - \bar{x}_l)(x_{lj} - \bar{x}_l)^t$

```
# bartlett ####
g<-5
p<-4
sum_n_l<-dim(skulls)[1]
lambda_star<-det(W_matrix)/det(B_matrix+W_matrix)
```

```
first<-sum_n_l-1
second<-(p+g)/2

bartlett<-(-(first-second)*log(lambda_star))
bartlett # test statistic
```

```
## [1] 59.25903
```

```
bartlett_test<-qchisq(p = 0.95,df = p*(g-1)) #critical value
bartlett_test
```

```
## [1] 26.29623
```

```
bartlett>bartlett_test # if TRUE -> reject H0 (at least one tau is not 0)
```

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

Here we do the test if:

$H_0 : \tau_1 = ...\tau_5 = 0$

H_1 : At least one τ is not 0

The test statistic is $-(n-1-\frac{p+g}{2} * ln(\frac{|\mathbf{W}|}{|\mathbf{B+W}|}) = 59.26$

And the critical value is $\chi^2_{p(g-1)}(\alpha)$ which in our case is 26.30.

The test statistic is bigger then the critical value and therefore we reject H_0 and make the conclusion that at least one τ is not 0.

c)

In assigment 3.b we did the conclusion that at least one τ is not 0 and now we want to test which epochs that differ from each other.

We do the intervalls by this formula:

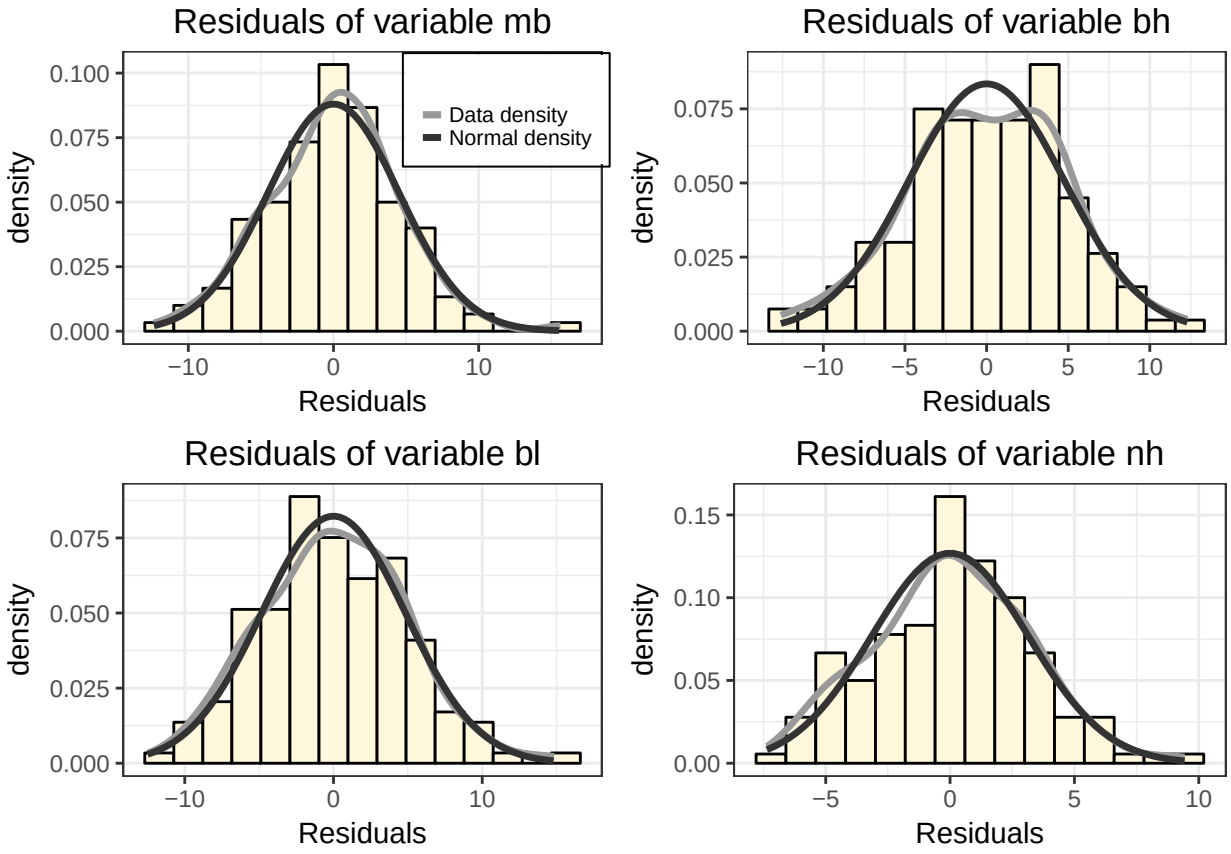
$\bar{x}_{ki} - x_{li} \pm t_{n-g}(\frac{\alpha}{pg(g-1)})\sqrt{\frac{w_{ii}}{n-g}}\sqrt{\frac{1+1}{n_k+n_l}}$

under	over	sign	variable	epoch_1	epoch_2
-4.91	2.91	FALSE	mb	c4000BC	c3300BC
-7.01	0.81	FALSE	mb	c4000BC	c1850BC
-8.04	-0.23	TRUE	mb	c4000BC	c200BC
-8.71	-0.89	TRUE	mb	c4000BC	cAD150
-6.01	1.81	FALSE	mb	c3300BC	c1850BC
-7.04	0.77	FALSE	mb	c3300BC	c200BC
-7.71	0.11	FALSE	mb	c3300BC	cAD150
-4.94	2.87	FALSE	mb	c1850BC	c200BC
-5.61	2.21	FALSE	mb	c1850BC	cAD150
-4.57	3.24	FALSE	mb	c200BC	cAD150
-3.22	5.02	FALSE	bh	c4000BC	c3300BC
-4.32	3.92	FALSE	bh	c4000BC	c1850BC
-2.82	5.42	FALSE	bh	c4000BC	c200BC
-0.85	7.39	FALSE	bh	c4000BC	cAD150
-5.22	3.02	FALSE	bh	c3300BC	c1850BC
-3.72	4.52	FALSE	bh	c3300BC	c200BC

under	over	sign	variable	epoch_1	epoch_2
-1.75	6.49	FALSE	bh	c3300BC	cAD150
-2.62	5.62	FALSE	bh	c1850BC	c200BC
-0.65	7.59	FALSE	bh	c1850BC	cAD150
-2.15	6.09	FALSE	bh	c200BC	cAD150
-4.08	4.28	FALSE	bl	c4000BC	c3300BC
-1.05	7.31	FALSE	bl	c4000BC	c1850BC
0.45	8.81	TRUE	bl	c4000BC	c200BC
1.49	9.85	TRUE	bl	c4000BC	cAD150
-1.15	7.21	FALSE	bl	c3300BC	c1850BC
0.35	8.71	TRUE	bl	c3300BC	c200BC
1.39	9.75	TRUE	bl	c3300BC	cAD150
-2.68	5.68	FALSE	bl	c1850BC	c200BC
-1.65	6.71	FALSE	bl	c1850BC	cAD150
-3.15	5.21	FALSE	bl	c200BC	cAD150
-2.41	3.01	FALSE	nh	c4000BC	c3300BC
-2.74	2.67	FALSE	nh	c4000BC	c1850BC
-4.14	1.27	FALSE	nh	c4000BC	c200BC
-3.54	1.87	FALSE	nh	c4000BC	cAD150
-3.04	2.37	FALSE	nh	c3300BC	c1850BC
-4.44	0.97	FALSE	nh	c3300BC	c200BC
-3.84	1.57	FALSE	nh	c3300BC	cAD150
-4.11	1.31	FALSE	nh	c1850BC	c200BC
-3.51	1.91	FALSE	nh	c1850BC	cAD150
-2.11	3.31	FALSE	nh	c200BC	cAD150

Six intervalls gets significant. For example epoch c4000BC differ from c200BC and cAD150 in variable mb and bl.

To trust the intervalls the residuals need to be normal distributed. So we make histogram of the residuals.



It looks like all four variables got normal distributed residuals. Which means that we can trust the intervalls.

Appendix

R-code

```
## ---- echo=FALSE, message=FALSE-----------------------------------------
library(ggplot2)
library(knitr)
library(car)

### Assigment 3 ####
library(heplots)
library(tidyverse)
library(GGally)
library(reshape2)
library(ggpubr)
library(kableExtra)

## ---- echo=FALSE-------------------------------------------------------
T1.9 <- read.delim("T1-9.dat", header=FALSE)

T1.9_dat <- as.matrix(T1.9[,2:8])
#due to high variability in data we center:
T1.9_dat <- scale(T1.9_dat, scale = FALSE)
```

```r
C <- cov(T1.9_dat)
C_inv <- solve(C)
mahl <- diag(T1.9_dat%*%C_inv%*%t(T1.9_dat))


## ---- echo=FALSE-----------------------------------------------------------
as.character(T1.9[which(mahl > qchisq(0.99, ncol(T1.9_dat))), 1])

## ----echo=FALSE------------------------------------------------------------
as.character(T1.9[which(mahl > qchisq(1- (0.01/nrow(T1.9_dat)), ncol(T1.9_dat))), 1])

## ----Q2a, warning=FALSE, echo=FALSE----------------------------------------


birds<-read.table("T5-12.DAT")
names(birds)<-c("tail","wing")

n = length(birds$tail)
p = ncol(birds)

S = cov(birds)
S_inv = solve(S)

means = colMeans(birds)
means_0 = c(190,275)

eigen_values = eigen(S)$values
eigen_vectors = eigen(S)$vectors

qchis = qchisq(0.95,2)
c_2 = sqrt(qchis) #(5-32)#pg241

length1 = c_2*sqrt(eigen_values[1]/n)
length2 = c_2*sqrt(eigen_values[2]/n)

xc <- means[1] # center x_c or h
yc <- means[2] # y_c or k
a <- length1 # major axis length
b <- length2 # minor axis length
phi = acos(t(c(0,1))%*%eigen_vectors[,1])

t <- seq(0, 2*pi, 0.01)
x <- xc + a*cos(t)*cos(phi) - b*sin(t)*sin(phi)
y <- yc + a*cos(t)*cos(phi) + b*sin(t)*cos(phi)
plot(x,y,pch=19, col='blue')
points(190,275) #See if the point is in the graph



## ----Q2b, echo=FALSE-------------------------------------------------------
# Simultaneos T2 intervals for mu_1 and mu_2
```

```r
#T_2 intervals
interval_1 = c(0,0)
interval_1[1] = means[1] - (sqrt(p*(n-1)*qf(0.95,p,n-p)/(n-p))) * sqrt(S[1,1]/n)
interval_1[2] = means[1] + (sqrt(p*(n-1)*qf(0.95,p,n-p)/(n-p))) * sqrt(S[1,1]/n)

interval_2 = c(0,0)
interval_2[1] = means[2] - (sqrt(p*(n-1)*qf(0.95,p,n-p)/(n-p))) * sqrt(S[2,2]/n)
interval_2[2] = means[2] + (sqrt(p*(n-1)*qf(0.95,p,n-p)/(n-p))) * sqrt(S[2,2]/n)

# Bonferoni'sintervals

binterval_1 = c(0,0)

binterval_1[1] = means[1] - qt(1-(0.05/(2*p)),n-1)*sqrt(S[1,1]/n)
binterval_1[2] = means[1] + qt(1-(0.05/(2*p)),n-1)*sqrt(S[1,1]/n)


binterval_2 = c(0,0)

binterval_2[1] = means[2] - qt(0.95/(2*p),n-1)*sqrt(S[2,2]/n)
binterval_2[2] = means[2] + qt(0.95/(2*p),n-1)*sqrt(S[2,2]/n)

## ------------------------------------------------------------------------
#for mu_1
print(interval_1)

#for mu_2
print(interval_2)

## ------------------------------------------------------------------------
#for mu_1
print(binterval_1)

#for mu_2
print(binterval_2)

## ----Q2c, echo=FALSE-----------------------------------------------------

library(ggplot2)


qplot(birds$tail, birds$wing) + ggtitle("Scatterplot of birds wing and tail size") +
  ylab("Wing Size") + xlab("Tail Size") +
  theme_bw()

qqPlot(birds$tail, main = "Q-Q Plot Tail Size")

qqPlot(birds$wing, main = "Q-Q Plot Wing Size")


## ----echo=FALSE----------------------------------------------------------
# Import data #####
skulls<-Skulls
```

```r
# ggpairs - Plot data ####
ggpairs(data = skulls,
        columns = c(2:5),#,
        title = "Correlation between variables",
        axisLabels = "none",
        mapping = aes(color = epoch))+
  theme(plot.title = element_text(hjust = 1))+
  theme_bw()


## ----grand_mean,echo=FALSE-----------------------------------------------
# grand_mean ####
statistcs<-skulls %>%
  melt(id.vars ="epoch" ) %>%
  group_by(variable) %>%
  summarise_if(is.numeric,.funs = c(n=length,
                                    var=function(x)sd(x)^2,
                                    mean=mean))


grand_mean<-statistcs$mean
names(grand_mean)<-as.character(statistcs$variable)
kable(data.frame(grand_mean),col.names = "Overall sample mean",align = "c",digits = 2)

## ----tau,echo=FALSE------------------------------------------------------
# tau ####
statistcs_epoch<-skulls %>%
  melt(id.vars ="epoch" ) %>%
  group_by(variable,epoch) %>%
  summarise_if(is.numeric,.funs = c(n=length,
                                    var=function(x)sd(x)^2,
                                    mean=mean))

statistcs_epoch$grand_mean<-rep(grand_mean,each=5)
statistcs_epoch$tau<-statistcs_epoch$mean-statistcs_epoch$grand_mean



kable(data.frame(statistcs_epoch$variable,statistcs_epoch$epoch,statistcs_epoch$tau)
      ,col.names = c("variable","epoch","Treatment effect (tau)"),align = "c",digits = 2)


skulls2<-skulls %>%
  melt(id.vars="epoch")

## ----residuals,echo=FALSE------------------------------------------------
# residuals ####
skulls2$factor_mean<-rep(statistcs_epoch$mean,each=30)
skulls2$right<-rep(paste(statistcs_epoch$variable,"_",statistcs_epoch$epoch,sep=""),each=30)
skulls2$residuals<-skulls2$value-skulls2$factor_mean

## ----B_matrix,echo=FALSE-------------------------------------------------
# B matrix ####
```

```r
statistcs_epoch_arrange<-statistcs_epoch %>% arrange(epoch)

statistcs_epoch_arrange$for_B<-statistcs_epoch_arrange$mean-statistcs_epoch_arrange$grand_mean

for_B<-as.matrix(statistcs_epoch_arrange$for_B)

lista<-list()
j<-1
lista[[j]]<-matrix(0,ncol=4,nrow=4)
for(i in c(1,5,9,13,17)){
  j<-j+1
  lista[[j]]<-(as.matrix(for_B[i:(i+3)])%*%t(as.matrix(for_B[i:(i+3)])))*30)
  lista[[1]]<-lista[[1]]+lista[[j]]
}
B_matrix<-lista[[1]]
colnames(B_matrix)<-c("mb","bh","bl","nh")
rownames(B_matrix)<-c("mb","bh","bl","nh")

kable(as.data.frame(B_matrix),digits = 2,align = "c",format = "latex") %>%
   kableExtra::kable_styling("striped") %>%
  kableExtra::add_header_above(c("B matrix"=5))

## ----W_matrix,echo=FALSE-----------------------------------------------

skulls3<-skulls2 %>%
  select(epoch,variable,residuals) %>%
  arrange(epoch) %>%
  group_by(variable) %>%
  mutate(id=seq_along(variable)) %>%
  spread(variable,residuals) %>%
  select(-id)


for_W<-skulls3[,-1]
lista2<-list()
j<-1
lista2[[j]]<-matrix(0,ncol=4,nrow=4)
for(i in c(1,31,61,91,121)){
  j<-j+1
  lista2[[j]]<-t(as.matrix(for_W[i:(i+29),]))%*%(as.matrix(for_W[i:(i+29),]))
  lista2[[1]]<-lista2[[1]]+lista2[[j]]
}
W_matrix<-lista2[[1]]

kable(as.data.frame(W_matrix),digits = 2,align = "c",format = "latex") %>%
   kableExtra::kable_styling("striped") %>%
  kableExtra::add_header_above(c("W matrix"=5))


## ----Wilks lambda (bartlett),echo=TRUE-----------------------------------
# bartlett ####
g<-5
```

```r
p<-4
sum_n_l<-dim(skulls)[1]
lambda_star<-det(W_matrix)/det(B_matrix+W_matrix)

first<-sum_n_l-1
second<-(p+g)/2

bartlett<-(-(first-second)*log(lambda_star))
bartlett # test statistic
bartlett_test<-qchisq(p = 0.95,df = p*(g-1)) #critical value
bartlett_test

bartlett>bartlett_test # if TRUE -> reject H0 (at least one tau is not 0)


## ----intervalls function, echo=FALSE-------------------------------------
intervalls<-function(data_analys,t_value=3.291887,sum_n_l=150){

  data_id<-data.frame(i=c(1,1,1,1,2,2,2,3,3,4),j=c(2,3,4,5,3,4,5,4,5,5))
  Ci_intervall<-data.frame(under=0,over=0,variable1=0,variable2=0,sign=TRUE,used_W=0,used_t=0,used_last=

  for(k in 1:dim(data_id)[1]){
    i<-data_id$i[k]
    j<-data_id$j[k]



    differ<-data_analys$mean[i]-data_analys$mean[j]

    first<-diag(W_matrix)[names(diag(W_matrix))==as.character(data_analys$variable[i])]/(sum_n_l-g)
    second<-(1/30)+(1/30) # always 30 obs
    last<-sqrt(first*second)

    under<-differ-t_value*last
    over<-differ+t_value*last
    Ci_intervall[k,1:2]<-c(under,over)

    Ci_intervall[k,5]<-((under<0) &( over>0))==FALSE
    Ci_intervall[k,6]<-diag(W_matrix)[names(diag(W_matrix))==as.character(data_analys$variable[i])]
    Ci_intervall[k,7]<-t_value
    Ci_intervall[k,8]<-last


    namn1<-paste(data_analys$variable[i],data_analys$epoch[i],sep="_")
    namn2<-paste(data_analys$variable[j],data_analys$epoch[j],sep="_")
    Ci_intervall[k,3:4]<-c(namn1,namn2)
  }

  Ci_intervall
}

## ----make intervalls,echo=FALSE------------------------------------------
g<-5
```

```
p<-4
sum_n_l<-dim(skulls)[1]
m<-(p*g*(g-1))/2
alpha<-0.05/(2*m)
t_value<-abs(qt(p = alpha,df = sum_n_l-g))

# Make the intervalls intervall ####
lista_intervall<-list()
lista_intervall[[1]]<-intervalls(statistcs_epoch %>%  filter(variable=="mb"))
lista_intervall[[2]]<-intervalls(statistcs_epoch %>%  filter(variable=="bh"))
lista_intervall[[3]]<-intervalls(statistcs_epoch %>%  filter(variable=="bl"))
lista_intervall[[4]]<-intervalls(statistcs_epoch %>%  filter(variable=="nh"))

all_intervalls<-rbind(lista_intervall[[1]],
      lista_intervall[[2]],
      lista_intervall[[3]],
      lista_intervall[[4]])


all_intervalls_2<-all_intervalls %>%
  mutate(variable=substr(variable1,1,2),
         epoch_1=substr(variable1,4,20),
         epoch_2=substr(variable2,4,20)) %>%
  select(under,over,sign,variable,epoch_1,epoch_2)

kable(all_intervalls_2,digits = 2,align = "c")



## ----graf_histogram function,echo=FALSE----------------------------------
graf_histogram<-function(my_data,legend=FALSE){

  my_mean<-mean(my_data$residuals)
  my_sd<-sd(my_data$residuals)

  x<-seq(min(my_data$residuals),max(my_data$residuals),by=0.3)
  y<-dnorm(x,mean=my_mean,sd=my_sd)

  data<-my_data$residuals
  titel<-paste("Residuals of variable",as.character(my_data$variable[1]))

  histogram<-ggplot() +
    geom_histogram(fill="cornsilk",
                   color="black",
                   aes(y = ..density..,x=data),
                   bins = 15)+
    stat_density(geom = "line", aes(x=data,colour = "nr1"),size=1.2) +
    geom_line(aes(x=x,y=y,colour="nr2"),size=1.2)+
    scale_colour_manual(name = "", values = c("gray60", "gray20"),
                        breaks = c("nr1", "nr2"),
                        labels = c("Data density", "Normal density"))+
    #scale_x_continuous(#breaks = seq(-10,3 , by = 1),
    #limits = min_max)+
```

```
    theme_bw()+
    xlab("Residuals")+
    theme(legend.position="none")+
    ggtitle(titel)+
    theme(plot.title = element_text(hjust = 0.5))+
    theme(legend.position = c(0.8, 0.8),
          legend.box.background = element_rect(),
          legend.box.margin = margin(1, 1, 1, 1),
          legend.key.size =unit(0.3, "cm"),
          legend.text = element_text( size = 8))

  if(legend==FALSE){
    histogram<-histogram+theme(legend.position="none")
  }
  histogram
}

## ----make histograms, echo=FALSE-----------------------------------------
graf_mb<-graf_histogram(skulls2 %>%  filter(variable=="mb"),legend=TRUE)
graf_bh<-graf_histogram(skulls2 %>%  filter(variable=="bh"))
graf_bl<-graf_histogram(skulls2 %>%  filter(variable=="bl"))
graf_nh<-graf_histogram(skulls2 %>%  filter(variable=="nh"))

ggarrange(graf_mb,graf_bh,graf_bl,graf_nh,ncol = 2,nrow = 2)
```