Lab 2 Multivariate Statistical Mathods

Carles Sans Fuentes, Joshua Hudson, Karo Ziomek

5 de diciembre de 2017

R Markdown

Question 1: Test of Outliers

Consider again the data set from the T1-9.dat file, National track records for women. In the first assignment we studied different distance measures between an observation and the sample average vector. The most common multivariate residual is the Mahalanobis distance and we computed this distance for all observations.

Here I write the code from the previous lab

```
require(ggplot2)
link <- "C:/Users/Carles/Desktop/MasterStatistics-MachineLearning/Master_subjects/Multivariate_Statisti
data <- t(read.table(link))
colnames(data) <- c(data[1, ])
data <- data[2:nrow(data), ]
## Preparing data
mydata <- apply(data, 2, as.numeric)
mydata <- t(mydata)
colnames(mydata) <- c("hundred", "twohundred", "fourhundred",
    "eighthundred", "1500", "3000", "marathon")
## preview of mydata
mydata <- as.data.frame(mydata)
mydata$hundred <- mydata$hundred/60
mydata$twohundred <- mydata$twohundred/60
mydata$fourhundred <- mydata$fourhundred/60

CovMat <- cov(mydata)
S <- apply(mydata, 2, FUN = function(x) {
    x - mean(x)
})

dmahal <- S %*% solve(as.matrix(CovMat)) %*% t(S)


countries <- diag(dmahal)
```

a) The Mahalanobis distance is approximately chi-square distributed, if the data comes from a multivariate normal distribution and the number of observations is large. Use this chi-square approximation for testing each observation at the 0.1% significance level and conclude which countries can be regarded as outliers. Should you use a multiple-testing correction procedure? Compare the results with and without one. Why is (or maybe is not) 0.1% a sensible significance level for this task?

```
df <- dim(mydata)[2] - 1


OutlierEvaluation <- function(vector, significance) {
```

```r
    probCountry <- integer(length(vector))
    multTestCorr <- integer(length(vector))
    m <- 7
    for (i in 1:length(vector)) {
        probCountry[i] <- pchisq(vector[i], df = df, ncp = 0,
            lower.tail = FALSE)
        # bonferroni
        multTestCorr[i] <- pchisq(vector[i], df = df, ncp = 0,
            lower.tail = FALSE)
    }
    return(list(normal = (countries[which(probCountry < significance)]),
        bonferroni = countries[which(multTestCorr < significance/m)]))

}
significance <- 0.001

OutlierEvaluation(vector = countries, significance = significance)
```

```
## $normal
##     KORN      PNG      SAM
## 26.16714 30.50725 35.01406
##
## $bonferroni
##      PNG      SAM
## 30.50725 35.01406
```

We should correct for multiple hypothesis because the more variables we are checking at the same time, the more probable it becomes that countries will appear to differ on at least one attribute due to random sampling error alone.

The significance level should be taken on a bigger alpha (e.g. 95% ci) because our data is not large such that impying that outliers are only 0.1% deviation of the dataset might be too restrictive.

b) One outlier is North Korea. This country is not an outlier with the Euclidean distance. Try to explain these seemingly contradictory results.

The euclidean distance is mean to be:

$$d_M^2(\vec{x}, \hat{x}) = (\vec{x} - \hat{x})^2/\sigma^2$$

The Mahalanobis distance is:

$$d_M^2(\vec{x}, \hat{x}) = (\vec{x} - \hat{x})^T C^{-1} (\vec{x} - \hat{x})$$

When using the Euclidean distance, we are supposing that the distance in the covariance matrix is reduced to the diagonal of the data, taking this diagonal variance as the as a general measure. When using the Mahalanobis distance, the covariance matrix (and therefore also the relation between variables) is also taken into account, assigning to our variance measure some probability given the other points. This changes the area of the variance from a square (in the case of the Euclidean distance) to an ellipse, accounting for more information concerning our data. For this, we can say that the Euclidean distance is an special case of the Mahalanobis distance, when the relations between variables are 0 (in the multivariate case).

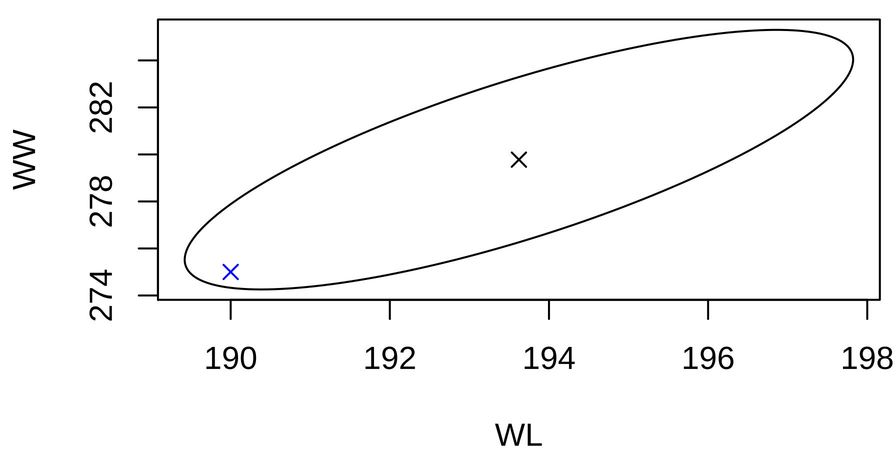
Question 2: Test, confidence region and confidence intervals for a mean vector

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 MAY NOT use loops!

```
link2 <- "C:/Users/Carles/Desktop/MasterStatistics-MachineLearning/Master_subjects/Multivariate_Statist
data2 <- read.table(link2)
```

(a) Find and sketch the 95% confidence ellipse for the population means ILl and IL2' Suppose it is known that iLl = 190 mm and iL2 = 275 mm for female hook-billed kites. Are these plausible values for the mean tail length and mean wing length for the female birds? Explain.

```
bird <- data2
X <- as.matrix(bird)
mu1 <- 190
mu2 <- 275
xbar <- colMeans(X)
n <- dim(X)[1]
p <- dim(X)[2]
S <- cov(X)
angles <- seq(0, 2 * pi, length.out = 200)
eigVal <- eigen(S)$values
eigVec <- eigen(S)$vectors
# eigVec_scaled <- eigVec %*% diag(sqrt(eigVal))
c2 <- p * (n - 1)/(n * (n - p)) * qf(p = 0.95, df1 = p, df2 = n -
    p)
ellBase <- cbind(sqrt(eigVal[1] * c2) * cos(angles), sqrt(eigVal[2] *
    c2) * sin(angles))
ellRot <- eigVec %*% t(ellBase)  #puts in eigenvector coordinates
{
    plot(ellRot[1, ] + xbar[1], ellRot[2, ] + xbar[2], xlab = "WL",
        ylab = "WW", type = "l")
    points(mu1, mu2, pch = 4, col = "blue")
    points(xbar[1], xbar[2], pch = 4)
}
```



Yes, they are plausible since the hypothesized vector is inside the “95% confidence region.

(b) Construct the simultaneous 95% T2_intervals for ILl and IL2 and the 95% Bonferroni intervals for iLl and iL2' Compare the two sets of intervals. What advantage, if any, do the

T2__intervals have over the Bonferroni intervals?

```
f <- p * (n - 1)/(n - p) * qf(0.95, df1 = p, df2 = n - p)
# simultaneous
WL_sim_low <- xbar[1] - sqrt(f) * sqrt(S[1, 1]/n)
WL_sim_upp <- xbar[1] + sqrt(f) * sqrt(S[1, 1]/n)
WW_sim_low <- xbar[2] - sqrt(f) * sqrt(S[2, 2]/n)
WW_sim_upp <- xbar[2] + sqrt(f) * sqrt(S[2, 2]/n)

# bonferroni (1 by 1)
t <- qt(p = (1 - 0.05/2), df = (n - 1))
WL_bon_low <- xbar[1] - t * sqrt(S[1, 1]/n)
WL_bon_upp <- xbar[1] + t * sqrt(S[1, 1]/n)
WW_bon_low <- xbar[2] - t * sqrt(S[2, 2]/n)
WW_bon_upp <- xbar[2] + t * sqrt(S[2, 2]/n)
```

Simultaneous:

$$189.42 \leq \mu_1 \leq 197.82$$

and

$$274.26 \leq \mu_2 \leq 285.3$$

Bonferroni:

$$190.32 \leq \mu_1 \leq 196.92$$

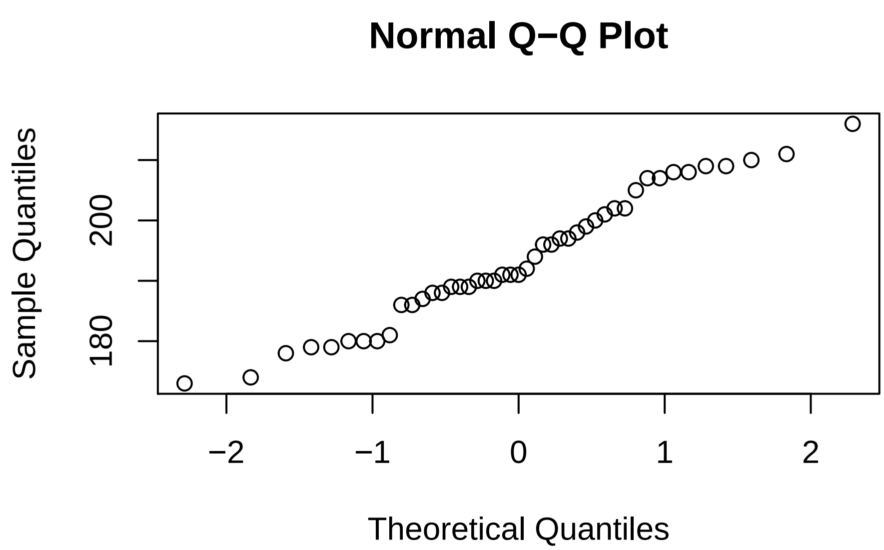
and

$$275.44 \leq \mu_1 \leq 284.12$$

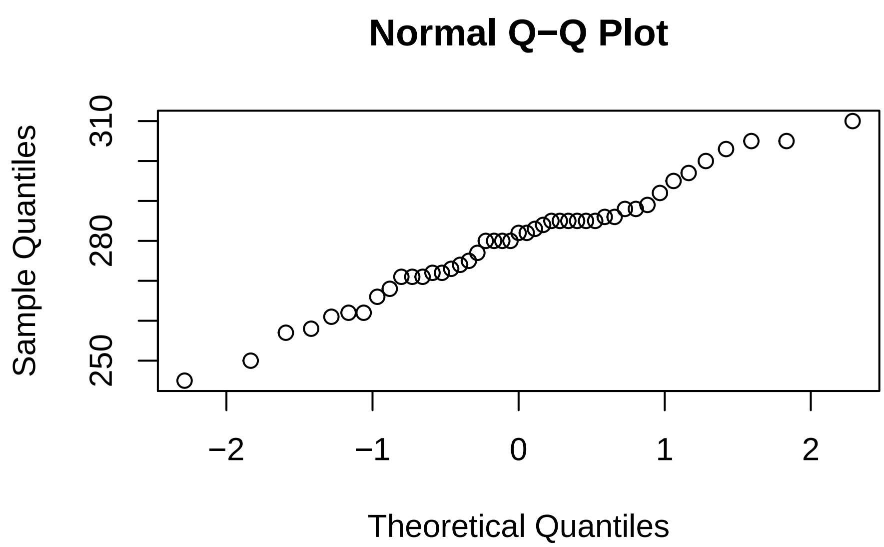
Simultaneous confidence intervals are larger than Bonferroni's confidence intervals. Simultaneous confidence intervals will touch the simultaneous confidence region from outside.

**(c) Is the bivariate normal distribution a viable population model? Explain with reference to Q-Q plots and a scatter diagram.

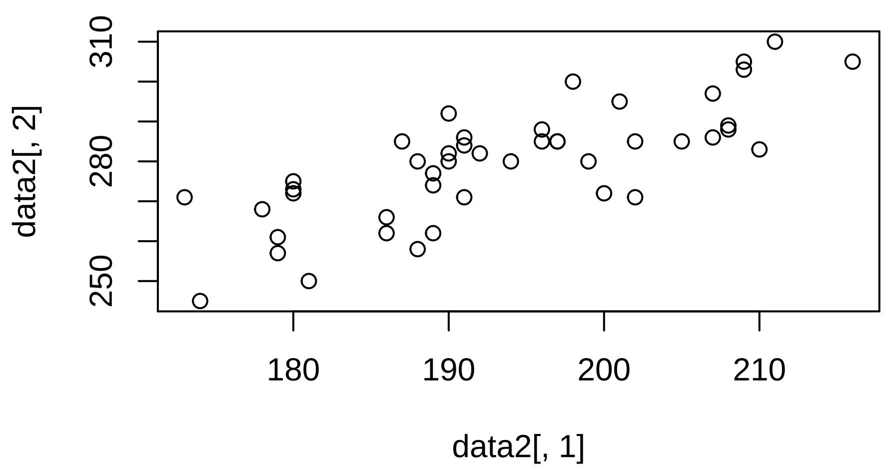
```
qqnorm(data2[, 1])
```



```r
qqnorm(data2[, 2])
```




```r
plot(data2[, 1], data2[, 2])
```



It is not viable since the qq plots are not straight (i.e. totally linear lines)

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

We will look at a data set on Egyptian skull measurements (published in 1905 and now in heplots R package as the object Skulls). Here observations are made from five epochs and on each object the maximum breadth (mb), basibregmatic height (bh), basialiveolar length (bl) and nasal height (nh) were measured.

```r
library("heplots")
data(Skulls)
Skulls
```

```
##       epoch  mb  bh  bl nh
## 1   c4000BC 131 138  89 49
## 2   c4000BC 125 131  92 48
## 3   c4000BC 131 132  99 50
## 4   c4000BC 119 132  96 44
## 5   c4000BC 136 143 100 54
## 6   c4000BC 138 137  89 56
## 7   c4000BC 139 130 108 48
## 8   c4000BC 125 136  93 48
## 9   c4000BC 131 134 102 51
## 10  c4000BC 134 134  99 51
## 11  c4000BC 129 138  95 50
## 12  c4000BC 134 121  95 53
## 13  c4000BC 126 129 109 51
## 14  c4000BC 132 136 100 50
## 15  c4000BC 141 140 100 51
## 16  c4000BC 131 134  97 54
## 17  c4000BC 135 137 103 50
## 18  c4000BC 132 133  93 53
## 19  c4000BC 139 136  96 50
## 20  c4000BC 132 131 101 49
## 21  c4000BC 126 133 102 51
## 22  c4000BC 135 135 103 47
## 23  c4000BC 134 124  93 53
## 24  c4000BC 128 134 103 50
## 25  c4000BC 130 130 104 49
## 26  c4000BC 138 135 100 55
## 27  c4000BC 128 132  93 53
## 28  c4000BC 127 129 106 48
## 29  c4000BC 131 136 114 54
## 30  c4000BC 124 138 101 46
## 31  c3300BC 124 138 101 48
## 32  c3300BC 133 134  97 48
## 33  c3300BC 138 134  98 45
## 34  c3300BC 148 129 104 51
## 35  c3300BC 126 124  95 45
## 36  c3300BC 135 136  98 52
## 37  c3300BC 132 145 100 54
## 38  c3300BC 133 130 102 48
## 39  c3300BC 131 134  96 50
## 40  c3300BC 133 125  94 46
## 41  c3300BC 133 136 103 53
## 42  c3300BC 131 139  98 51
## 43  c3300BC 131 136  99 56
## 44  c3300BC 138 134  98 49
## 45  c3300BC 130 136 104 53
## 46  c3300BC 131 128  98 45
## 47  c3300BC 138 129 107 53
## 48  c3300BC 123 131 101 51
## 49  c3300BC 130 129 105 47
```

```
## 50   c3300BC 134 130  93 54
## 51   c3300BC 137 136 106 49
## 52   c3300BC 126 131 100 48
## 53   c3300BC 135 136  97 52
## 54   c3300BC 129 126  91 50
## 55   c3300BC 134 139 101 49
## 56   c3300BC 131 134  90 53
## 57   c3300BC 132 130 104 50
## 58   c3300BC 130 132  93 52
## 59   c3300BC 135 132  98 54
## 60   c3300BC 130 128 101 51
## 61   c1850BC 137 141  96 52
## 62   c1850BC 129 133  93 47
## 63   c1850BC 132 138  87 48
## 64   c1850BC 130 134 106 50
## 65   c1850BC 134 134  96 45
## 66   c1850BC 140 133  98 50
## 67   c1850BC 138 138  95 47
## 68   c1850BC 136 145  99 55
## 69   c1850BC 136 131  92 46
## 70   c1850BC 126 136  95 56
## 71   c1850BC 137 129 100 53
## 72   c1850BC 137 139  97 50
## 73   c1850BC 136 126 101 50
## 74   c1850BC 137 133  90 49
## 75   c1850BC 129 142 104 47
## 76   c1850BC 135 138 102 55
## 77   c1850BC 129 135  92 50
## 78   c1850BC 134 125  90 60
## 79   c1850BC 138 134  96 51
## 80   c1850BC 136 135  94 53
## 81   c1850BC 132 130  91 52
## 82   c1850BC 133 131 100 50
## 83   c1850BC 138 137  94 51
## 84   c1850BC 130 127  99 45
## 85   c1850BC 136 133  91 49
## 86   c1850BC 134 123  95 52
## 87   c1850BC 136 137 101 54
## 88   c1850BC 133 131  96 49
## 89   c1850BC 138 133 100 55
## 90   c1850BC 138 133  91 46
## 91    c200BC 137 134 107 54
## 92    c200BC 141 128  95 53
## 93    c200BC 141 130  87 49
## 94    c200BC 135 131  99 51
## 95    c200BC 133 120  91 46
## 96    c200BC 131 135  90 50
## 97    c200BC 140 137  94 60
## 98    c200BC 139 130  90 48
## 99    c200BC 140 134  90 51
## 100   c200BC 138 140 100 52
## 101   c200BC 132 133  90 53
## 102   c200BC 134 134  97 54
## 103   c200BC 135 135  99 50
```

```
## 104  c200BC 133 136  95 52
## 105  c200BC 136 130  99 55
## 106  c200BC 134 137  93 52
## 107  c200BC 131 141  99 55
## 108  c200BC 129 135  95 47
## 109  c200BC 136 128  93 54
## 110  c200BC 131 125  88 48
## 111  c200BC 139 130  94 53
## 112  c200BC 144 124  86 50
## 113  c200BC 141 131  97 53
## 114  c200BC 130 131  98 53
## 115  c200BC 133 128  92 51
## 116  c200BC 138 126  97 54
## 117  c200BC 131 142  95 53
## 118  c200BC 136 138  94 55
## 119  c200BC 132 136  92 52
## 120  c200BC 135 130 100 51
## 121  cAD150 137 123  91 50
## 122  cAD150 136 131  95 49
## 123  cAD150 128 126  91 57
## 124  cAD150 130 134  92 52
## 125  cAD150 138 127  86 47
## 126  cAD150 126 138 101 52
## 127  cAD150 136 138  97 58
## 128  cAD150 126 126  92 45
## 129  cAD150 132 132  99 55
## 130  cAD150 139 135  92 54
## 131  cAD150 143 120  95 51
## 132  cAD150 141 136 101 54
## 133  cAD150 135 135  95 56
## 134  cAD150 137 134  93 53
## 135  cAD150 142 135  96 52
## 136  cAD150 139 134  95 47
## 137  cAD150 138 125  99 51
## 138  cAD150 137 135  96 54
## 139  cAD150 133 125  92 50
## 140  cAD150 145 129  89 47
## 141  cAD150 138 136  92 46
## 142  cAD150 131 129  97 44
## 143  cAD150 143 126  88 54
## 144  cAD150 134 124  91 55
## 145  cAD150 132 127  97 52
## 146  cAD150 137 125  85 57
## 147  cAD150 129 128  81 52
## 148  cAD150 140 135 103 48
## 149  cAD150 147 129  87 48
## 150  cAD150 136 133  97 51
```

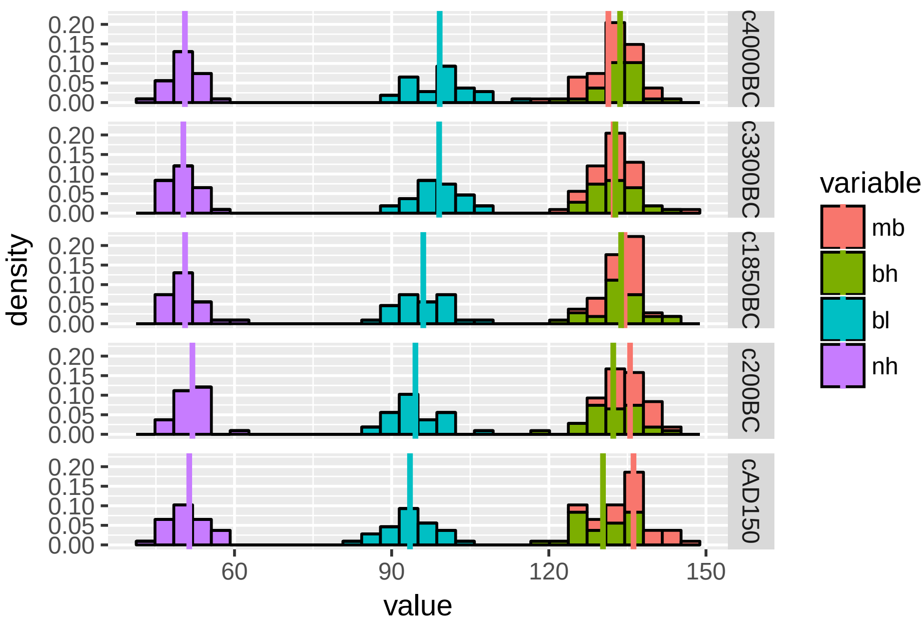
a) Explore the data first and present plots that you find informative.

```r
dim(Skulls)
```

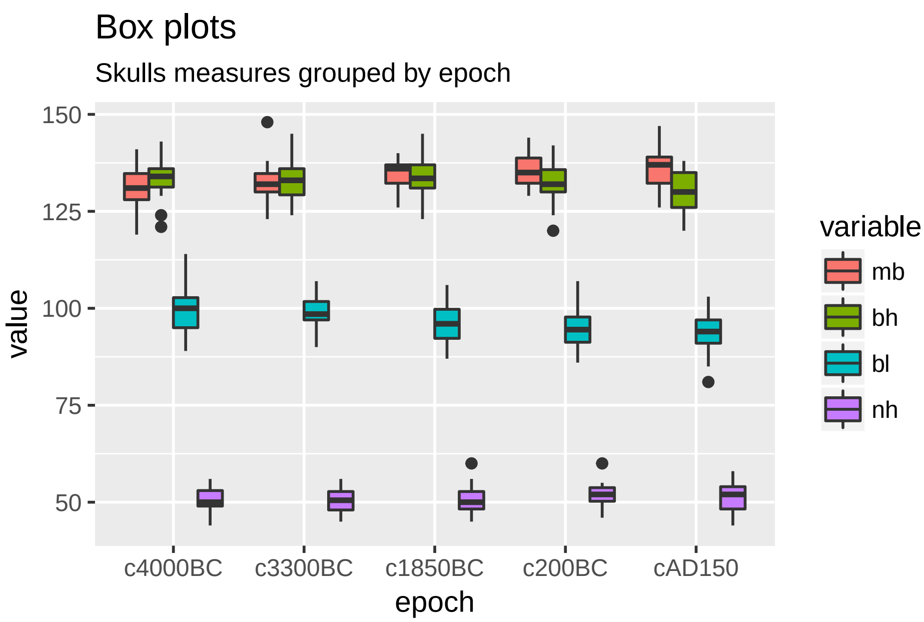
```
## [1] 150   5
```

```r
library(ggplot2)
library(reshape2)
```

```r
Skullshape <- Skulls
xymelt <- melt(Skullshape)


# Getting the mean
library(dplyr)

DataMeanbyEpoch <- Skulls %>% group_by(epoch) %>% summarise_all(funs(mean(.,
    na.rm = TRUE)))

Meltmean <- melt(DataMeanbyEpoch)
## histogram by age and variables
sp <- ggplot(xymelt, aes(x = value, fill = variable, group = variable)) +
    geom_histogram(aes(y = ..density..), col = "black")

# Divide by levels of 'sex', in the vertical direction
sp + facet_grid(epoch ~ ., scales = "free_x") + geom_vline(data = Meltmean,
    aes(xintercept = value, col = variable), size = 1)
```




```r
dfmelt <- melt(Skulls, measure.vars = 2:5)

g <- ggplot(dfmelt) + geom_boxplot(aes(x = epoch, y = value,
    fill = variable)) +
labs(title = "Box plots", subtitle = "Skulls measures grouped by epoch")
g
```



In the previous plots, the distribution of each variable by variable and epoch has been performed. The vertical line corresponds to the mean value of that particular distribution in the histogram and the boxplot's horizontal line inside is the median of it.

b) Now we are interested whether there are differences between the epochs. Do the mean vectors differ? Study this question and justify your conclusions.

Given the boxplots in 3a), the means for each variables seems to differ between the epochs. The particular means are:

```
# aggregate
library(knitr)
means <- aggregate(Skulls[, 2:5], list(Skulls$epoch), mean)
kable(means, caption = "Variable means for each epoch", col.names = c("epoch",
    "mb", "bh", "bl", "nh"))
```

Table 1: Variable means for each epoch

epoch	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

However, having a look at the specific means displayed in the table above only points out small differences between the means of each variable for the epochs. The result of a manova is:

```
res <- manova(cbind(mb, bh, bl, nh) ~ epoch, data = Skulls)
summary.aov(res)
```

```
##  Response mb :
##              Df  Sum Sq Mean Sq F value    Pr(>F)
## epoch         4  502.83 125.707  5.9546 0.0001826 ***
## Residuals   145 3061.07  21.111
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response bh :
##              Df Sum Sq Mean Sq F value  Pr(>F)
## epoch        4  229.9  57.477  2.4474 0.04897 *
## Residuals  145 3405.3  23.485
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response bl :
##              Df Sum Sq Mean Sq F value    Pr(>F)
## epoch        4  803.3 200.823  8.3057 4.636e-06 ***
## Residuals  145 3506.0  24.179
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response nh :
##              Df Sum Sq Mean Sq F value Pr(>F)
## epoch        4   61.2  15.300   1.507 0.2032
## Residuals  145 1472.1  10.153
```

c) If the means differ between epochs compute and report simultaneous confidence intervals. Inspect the residuals whether they have mean 0 and if they deviate from normality (graphically). Tip: It might be helpful for you to read Exercise 6.24 of Johnson, Wichern. The function manova() can be useful for this question and the residuals can be found in the $res field.

The confidence intervals are provided below:

```r
## c number of groups
groups <- ncol(Skulls)
# number of variables
p <- ncol(Skulls[, -which("epoch" %in% colnames(Skulls))])

# number of observations per group
n_k <- Skulls %>% group_by(epoch) %>% summarise(number = n())

# n observations
n <- nrow(Skulls)

c <- sqrt(1/(30 + 30))
df_hej <- Skulls %>% group_by(epoch)

# cov(df_hej[df_hej$epoch == 'c4000BC',2:5] )

# calculating sum of covariance per epoch
W <- (30 - 1) * cov(df_hej[df_hej$epoch == "c4000BC", 2:5]) +
    (30 - 1) * cov(df_hej[df_hej$epoch == "c3300BC", 2:5]) +
    (30 - 1) * cov(df_hej[df_hej$epoch == "c1850BC", 2:5]) +
    (30 - 1) * cov(df_hej[df_hej$epoch == "c200BC", 2:5]) + (30 -
    1) * cov(df_hej[df_hej$epoch == "cAD150", 2:5])

# comparision of two mean at the same time
res_intervals <- function(k, l) {
```

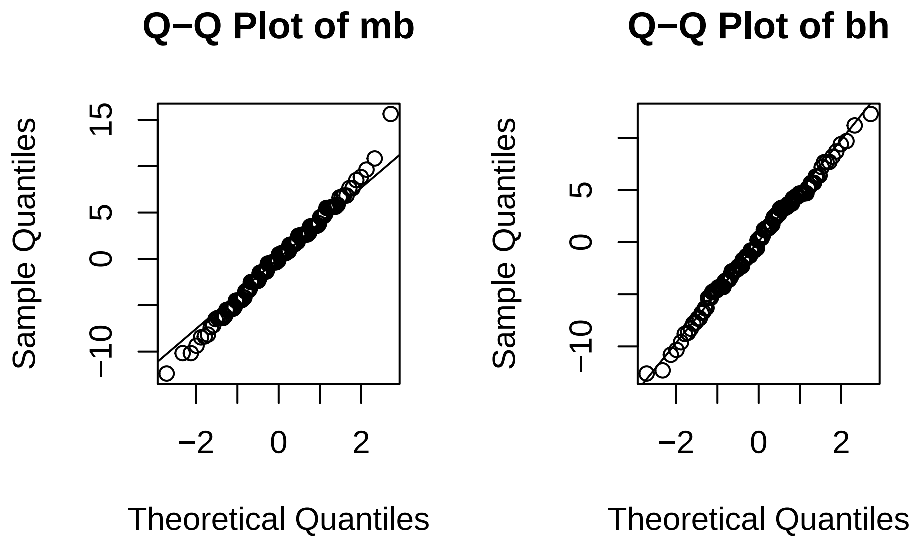
```
    alpha <- 0.05
    df_ci <- data.frame(low_lim = numeric(0), upper_limit = numeric(0))

    for (i in 1:p) {
        up <- means[k, i + 1] - means[l, i + 1] + qt(1 - (alpha/(p *
            groups * (groups - 1))), n - groups) * c * sqrt(W[i,
            i]/(n - groups))
        low <- means[k, i + 1] - means[l, i + 1] - qt(1 - (alpha/(p *
            groups * (groups - 1))), n - groups) * c * sqrt(W[i,
            i]/(n - groups))
        df_ci[i, c("low_lim", "upper_limit")] <- c(low, up)
    }
    row.names(df_ci) <- c("mb", "bh", "bl", "nh")
    df_ci$groups <- paste(k, l, sep = ",")
    df_ci <- df_ci[, c(3, 1, 2)]
    return(df_ci)
}

df_epoch12 <- res_intervals(1, 2)
df_epoch13 <- res_intervals(1, 3)
df_epoch14 <- res_intervals(1, 4)
df_epoch15 <- res_intervals(1, 5)

df_epoch23 <- res_intervals(2, 3)
df_epoch24 <- res_intervals(2, 4)
df_epoch25 <- res_intervals(2, 5)

df_epoch34 <- res_intervals(3, 4)
df_epoch35 <- res_intervals(3, 5)

df_epoch45 <- res_intervals(4, 5)

table1 <- rbind(df_epoch12, df_epoch13, df_epoch14, df_epoch15,
    df_epoch23, df_epoch24, df_epoch25, df_epoch34, df_epoch35,
    df_epoch45)

kable(table1, caption = "Upper and lower limits for the confidence intervals in epoch wise comparison")
```

Table 2: Upper and lower limits for the confidence intervals in epoch wise comparison

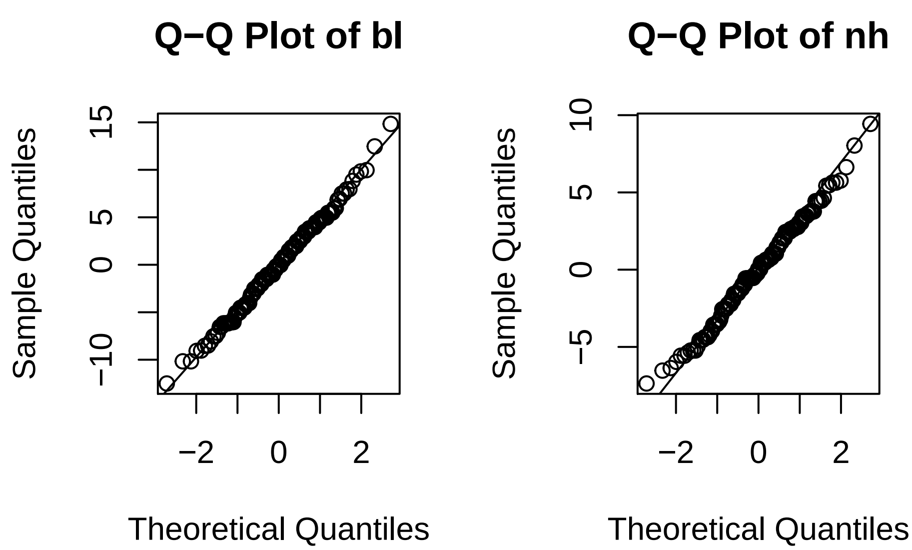
	groups	low_lim	upper_limit
mb	1,2	-2.9526378	0.9526378
bh	1,2	-1.1594956	2.9594956
bl	1,2	-1.9897253	2.1897253
nh	1,2	-1.0541254	1.6541254
mb1	1,3	-5.0526378	-1.1473622
bh1	1,3	-2.2594956	1.8594956
bl1	1,3	1.0436080	5.2230586
nh1	1,3	-1.3874587	1.3207921
mb2	1,4	-6.0859712	-2.1806955
bh2	1,4	-0.7594956	3.3594956
bl2	1,4	2.5436080	6.7230586

	groups	low_lim	upper_limit
nh2	1,4	-2.7874587	-0.0792079
mb3	1,5	-6.7526378	-2.8473622
bh3	1,5	1.2071711	5.3261623
bl3	1,5	3.5769414	7.7563920
nh3	1,5	-2.1874587	0.5207921
mb4	2,3	-4.0526378	-0.1473622
bh4	2,3	-3.1594956	0.9594956
bl4	2,3	0.9436080	5.1230586
nh4	2,3	-1.6874587	1.0207921
mb5	2,4	-5.0859712	-1.1806955
bh5	2,4	-1.6594956	2.4594956
bl5	2,4	2.4436080	6.6230586
nh5	2,4	-3.0874587	-0.3792079
mb6	2,5	-5.7526378	-1.8473622
bh6	2,5	0.3071711	4.4261623
bl6	2,5	3.4769414	7.6563920
nh6	2,5	-2.4874587	0.2207921
mb7	3,4	-2.9859712	0.9193045
bh7	3,4	-0.5594956	3.5594956
bl7	3,4	-0.5897253	3.5897253
nh7	3,4	-2.7541254	-0.0458746
mb8	3,5	-3.6526378	0.2526378
bh8	3,5	1.4071711	5.5261623
bl8	3,5	0.4436080	4.6230586
nh8	3,5	-2.1541254	0.5541254
mb9	4,5	-2.6193045	1.2859712
bh9	4,5	-0.0928289	4.0261623
bl9	4,5	-1.0563920	3.1230586
nh9	4,5	-0.7541254	1.9541254

```r
par(mfrow = c(1, 2))

qqnorm(res$residuals[, 1], main = "Q-Q Plot of mb")
qqline(res$residuals[, 1])

qqnorm(res$residuals[, 2], main = "Q-Q Plot of bh")
qqline(res$residuals[, 2])
```




```r
qqnorm(res$residuals[, 3], main = "Q-Q Plot of bl")
qqline(res$residuals[, 3])

qqnorm(res$residuals[, 4], main = "Q-Q Plot of nh")
qqline(res$residuals[, 4])
```



The residuals from bh and bl looks fairly normal whereas residuals from mb and nh do not.