Multivariate Statistical Methods

Assignment 2

Allan Gholmi, Emma Wallentinsson, Rasmus Holm

2017-12-08

Question 1

```r
data <- read.table("../data/T1-9.dat")
names(data) <- c("country", "100m", "200m", "400m", "800m", "1500m", "3000m", "marathon")
numeric_data <- data[, -1]

countries <- as.character(data$country)
```

a)

```r
X <- as.matrix(numeric_data)
means <- colMeans(X)
covariances <- cov(X)
X_central <- X - rep(1, nrow(X)) %*% t(means)

mdist_sq <- X_central %*% solve(covariances) %*% t(X_central)
country_mdist <- diag(mdist_sq)

significance_level <- 0.1
p <- ncol(X)
quantile <- qchisq(1 - significance_level, df=p)

outliers <- country_mdist > quantile
print("Outliers without correction")
#> [1] "Outliers without correction"
countries[outliers]
#> [1] "COK"  "KORN" "MEX"  "PNG"  "SAM"
```

No clue what the multiple-testing correction procedure refers to.

b)

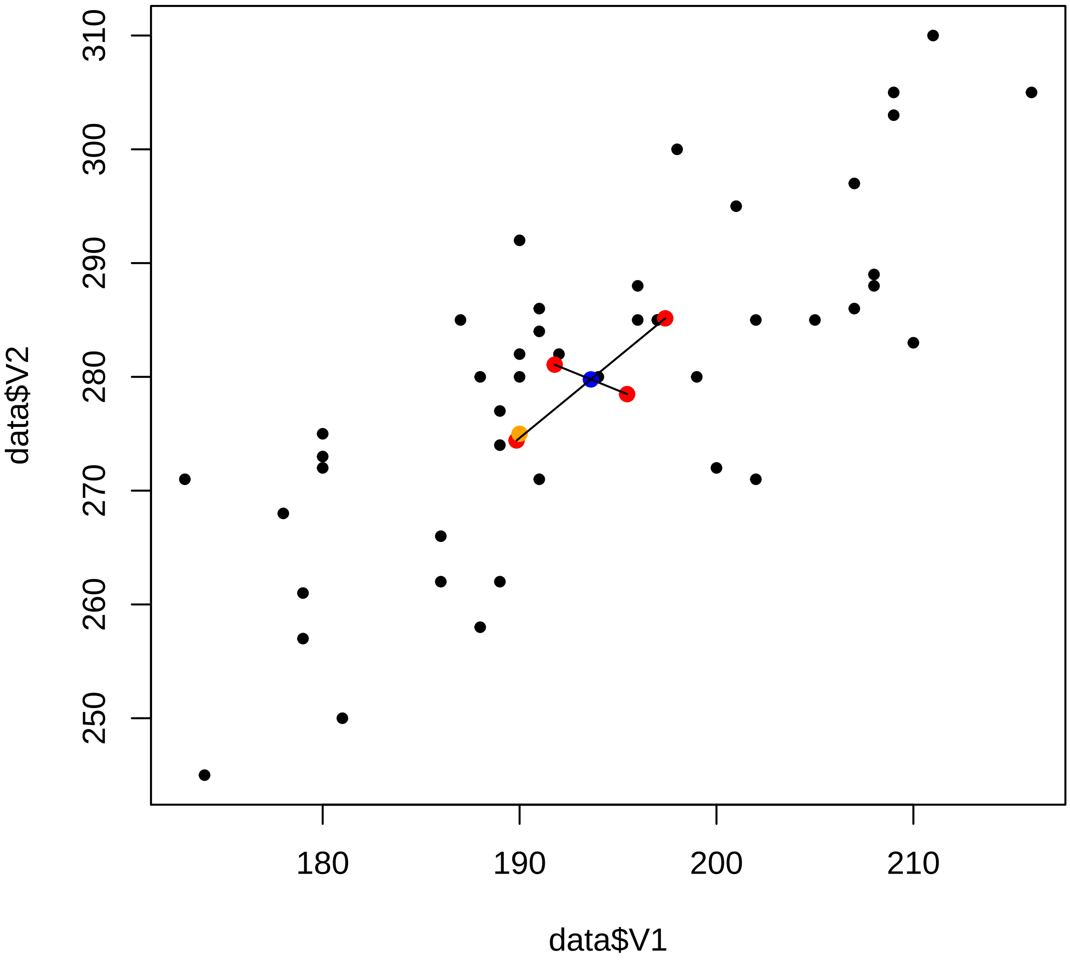
The Mahalanobis takes the covariances into consideration so the distances lead to a elliptic decision boundary as opposed to the circular boundary by Euclidean distance. That indicates that North Korea is an outlier based on the covariances meaning their result do not follow the general trend.

Question 2

```r
data <- read.table("../data/T5-12.DAT")
```

a)

```r
x_bar <- colMeans(data)
S <- cov(data)
S_inv <- solve(S)

n <- nrow(data)
p <- ncol(data)

eigen_values <- eigen(S)$values
eigen_vectors <- eigen(S)$vectors

true_mean <- c(190, 275)
confidence_level <- 0.05

half_lengths <- sqrt(eigen_values) * sqrt((p * (n - 1)) / (n * (n - p)) *
                                          qf(1 - confidence_level, df1=p, df2=n - p))

p1 <- x_bar + eigen_vectors[, 1] * half_lengths[1]
p2 <- x_bar - eigen_vectors[, 1] * half_lengths[1]

p3 <- x_bar + eigen_vectors[, 2] * half_lengths[2]
p4 <- x_bar - eigen_vectors[, 2] * half_lengths[2]

x <- c(p1[1], p2[1], p3[1], p4[1])
y <- c(p1[2], p2[2], p3[2], p4[2])

plot(data$V1, data$V2, pch=20)
points(x, y, col="red", pch=20, cex=1.5)
points(true_mean[1], true_mean[2], col="orange", pch=20, cex=1.5)
points(x_bar[1], x_bar[2], col="blue", pch=20, cex=1.5)
segments(rep(x_bar[1], 4), rep(x_bar[2], 4), x, y)
```



b)

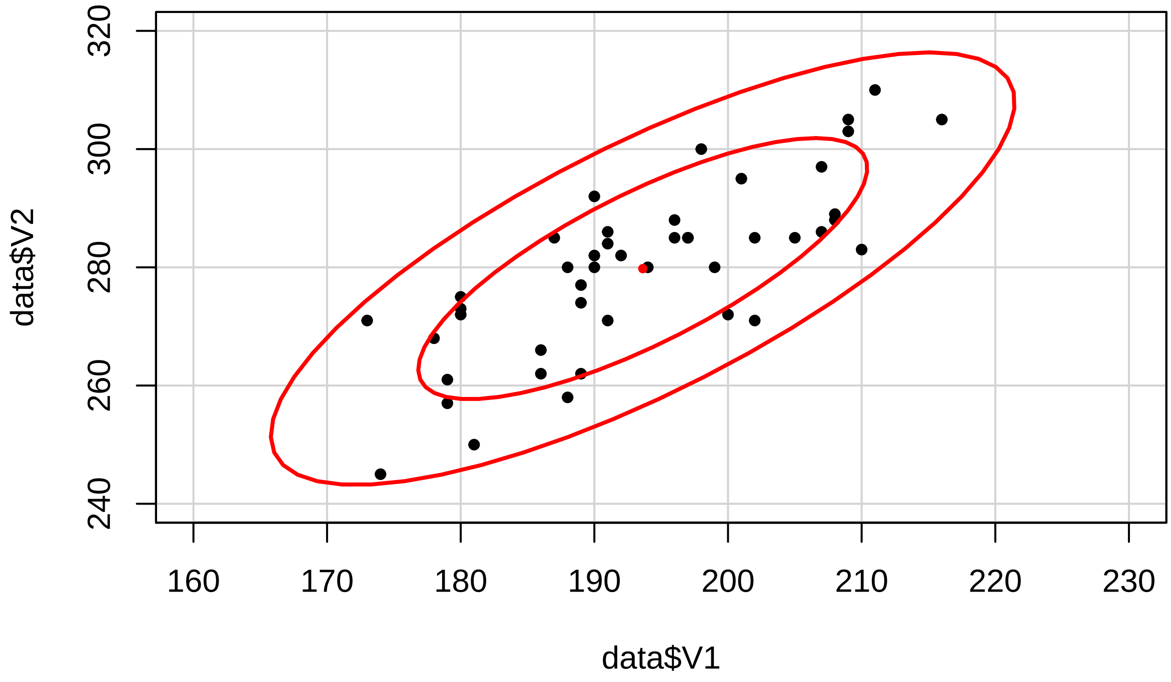
```
Tsq_offset <- sqrt(p * (n - 1) * qf(1 - confidence_level, df1=p, df2=n - p) / (n - p) * diag(S) / n)
Tsq_confidence_interval <- rbind(x_bar - Tsq_offset, x_bar + Tsq_offset)

bonferroni_offset <- sqrt(diag(S) / n) * qt(1 - confidence_level / (2 * p), df=n - 1)
bonferroni_confidence_interval <- rbind(x_bar - bonferroni_offset, x_bar + bonferroni_offset)

print("T-square Intervals")
#> [1] "T-square Intervals"
Tsq_confidence_interval
#>            V1       V2
#> [1,] 189.4217 274.2564
#> [2,] 197.8227 285.2992
```

```
print("Bonferroni Intervals")
#> [1] "Bonferroni Intervals"
bonferroni_confidence_interval
#>            V1       V2
#> [1,] 189.8216 274.7819
#> [2,] 197.4229 284.7736
```

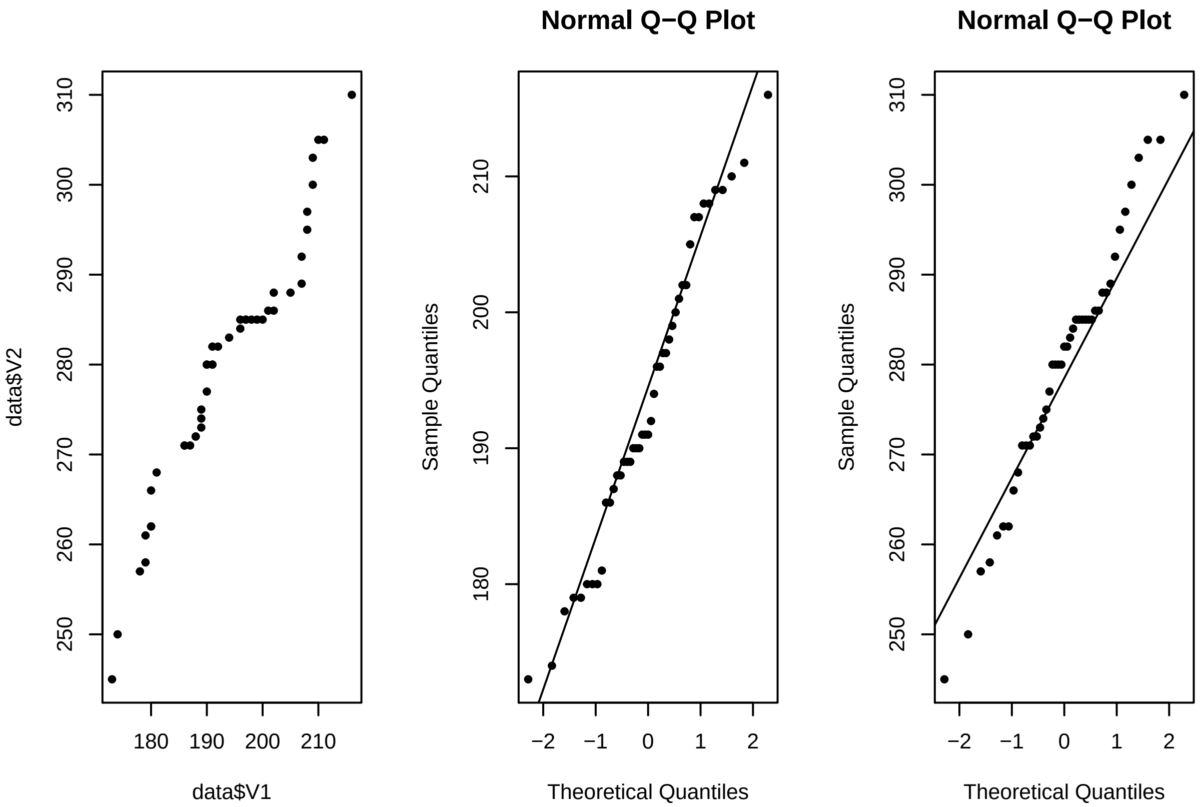
T-square test always gives wider confidence intervals since it takes the correlation between the measured variables into account. Bonferroni intervals are more precise if you are interested in the individual component means, but if you are interested in the overall data mean you should consider the T-square intervals.

c)

```
dataEllipse(x=data$V1, y=data$V2, pch=20, levels=c(0.68, 0.95),
            xlim=c(160, 230), ylim=c(240, 320), center.cex=0.5)
```



```
old <- par(mfrow=c(1, 3))
qqplot(data$V1, data$V2, pch=20)
?qqplot
qqnorm(data$V1, pch=20)
qqline(data$V1)
qqnorm(data$V2, pch=20)
qqline(data$V2)
```



```r
par(old)
```

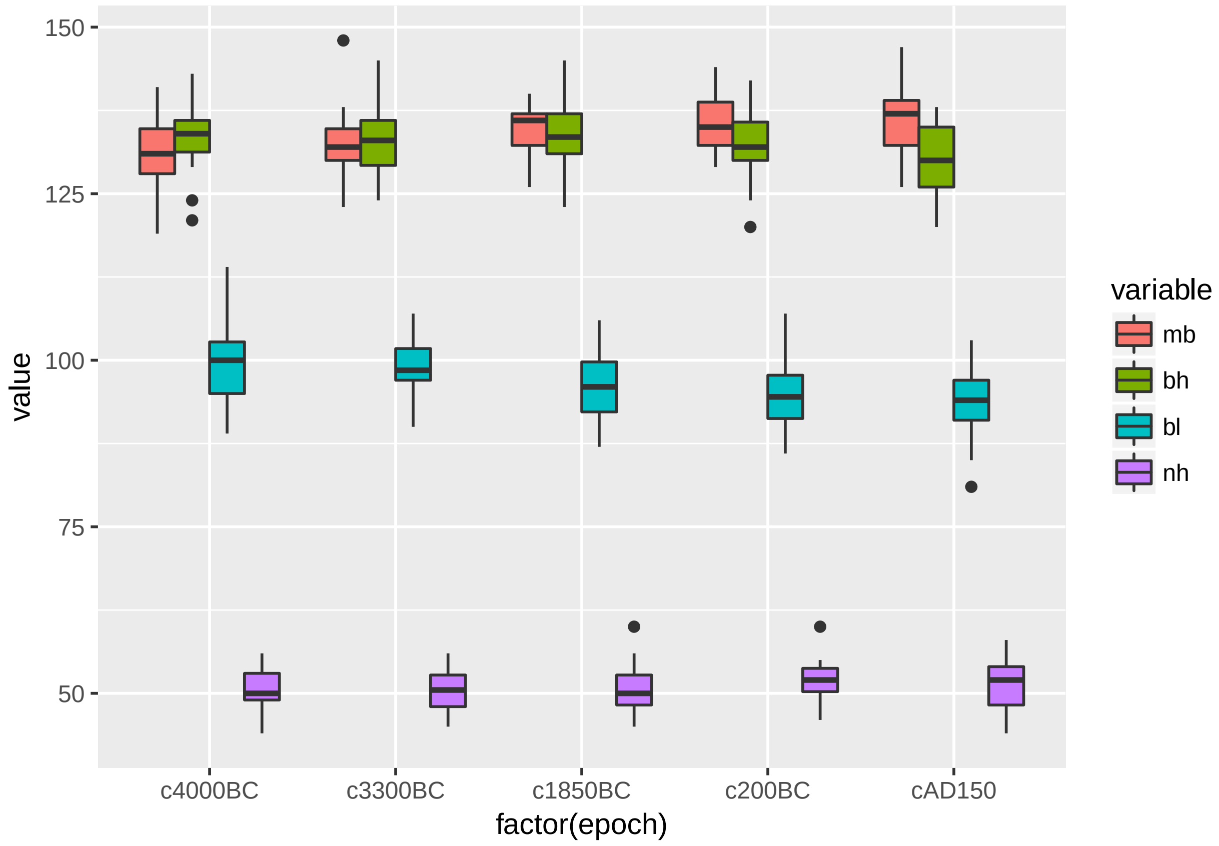
A bivariate normal distribution would be a viable population model. The qqplots do not deviate to much from the straight lines and the scatter plot shows that the points could very well have been generated from a bivariate normal distribution.

Question 3

```
library(heplots)
library(dplyr)
library(ggplot2)
library(reshape2)

data <- Skulls
numeric_data <- data[, -1]
colors <- as.numeric(data$epoch)
```

a)

```
# pairs(numeric_data, col=colors)
mm <- melt(data, id="epoch")
ggplot(mm) +
    geom_boxplot(aes(x=factor(epoch), y=value, fill=variable))
```

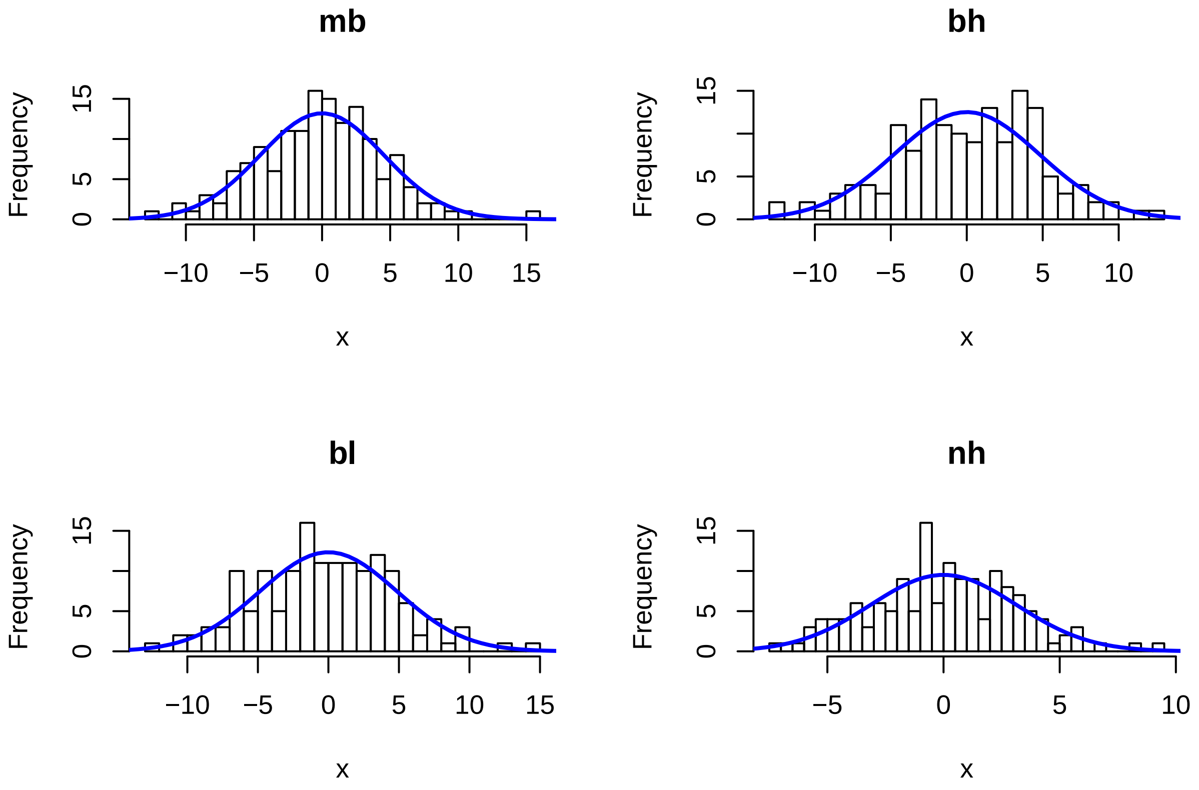


b)

```
group_means <- data %>%
    group_by(epoch) %>%
    summarise_all(funs(mean(., na.rm=TRUE)))


fit <- manova(cbind(mb, bh, bl, nh) ~ data$epoch, data)
```

c)

```
residuals <- fit$res
col_names <- c("mb", "bh", "bl", "nh")

old <- par(mfrow=c(2, 2))

for (col in 1:ncol(residuals)) {
    x <- residuals[, col]
    main <- col_names[col]
    h <- hist(x, breaks=25, main=main)
    offset <- (max(x) - min(x)) / 2
    xfit <- seq(min(x) - offset, max(x) + offset, length = 100)
    yfit <- dnorm(xfit, mean = mean(x), sd = sd(x))
    yfit <- yfit * diff(h$mids[1:2]) * length(x)
    lines(xfit, yfit, col="blue", lwd=2)
}
```


mb
Frequency
0
5
15
−10
−5
0
5
10
15
x
bh
Frequency
0
5
15
−10
−5
0
5
10
x
bl
Frequency
0
5
15
−10
−5
0
5
10
15
x
nh
Frequency
0
5
15
−5
0
5
10
x

```
par(old)
```

Appendix

Code

```
# Question 1
data <- read.table("../data/T1-9.dat")
names(data) <- c("country", "100m", "200m", "400m", "800m", "1500m", "3000m", "marathon")
numeric_data <- data[, -1]

countries <- as.character(data$country)
X <- as.matrix(numeric_data)
means <- colMeans(X)
covariances <- cov(X)
X_central <- X - rep(1, nrow(X)) %*% t(means)

mdist_sq <- X_central %*% solve(covariances) %*% t(X_central)
country_mdist <- diag(mdist_sq)

significance_level <- 0.1
p <- ncol(X)
quantile <- qchisq(1 - significance_level, df=p)

outliers <- country_mdist > quantile
print("Outliers without correction")
countries[outliers]


# Question 2
data <- read.table("../data/T5-12.DAT")
x_bar <- colMeans(data)
S <- cov(data)
S_inv <- solve(S)

n <- nrow(data)
p <- ncol(data)

eigen_values <- eigen(S)$values
eigen_vectors <- eigen(S)$vectors

true_mean <- c(190, 275)
confidence_level <- 0.05

half_lengths <- sqrt(eigen_values) * sqrt((p * (n - 1)) / (n * (n - p)) *
                                          qf(1 - confidence_level, df1=p, df2=n - p))

p1 <- x_bar + eigen_vectors[, 1] * half_lengths[1]
p2 <- x_bar - eigen_vectors[, 1] * half_lengths[1]

p3 <- x_bar + eigen_vectors[, 2] * half_lengths[2]
p4 <- x_bar - eigen_vectors[, 2] * half_lengths[2]

x <- c(p1[1], p2[1], p3[1], p4[1])
y <- c(p1[2], p2[2], p3[2], p4[2])
```

```r
plot(data$V1, data$V2, pch=20)
points(x, y, col="red", pch=20, cex=1.5)
points(true_mean[1], true_mean[2], col="orange", pch=20, cex=1.5)
points(x_bar[1], x_bar[2], col="blue", pch=20, cex=1.5)
segments(rep(x_bar[1], 4), rep(x_bar[2], 4), x, y)
Tsq_offset <- sqrt(p * (n - 1) * qf(1 - confidence_level, df1=p, df2=n - p) / (n - p) * diag(S) / n)
Tsq_confidence_interval <- rbind(x_bar - Tsq_offset, x_bar + Tsq_offset)

bonferroni_offset <- sqrt(diag(S) / n) * qt(1 - confidence_level / (2 * p), df=n - 1)
bonferroni_confidence_interval <- rbind(x_bar - bonferroni_offset, x_bar + bonferroni_offset)

print("T-square Intervals")
Tsq_confidence_interval

print("Bonferroni Intervals")
bonferroni_confidence_interval


# Question 3
library(heplots)
library(dplyr)
library(ggplot2)
library(reshape2)

data <- Skulls
numeric_data <- data[, -1]
colors <- as.numeric(data$epoch)
# pairs(numeric_data, col=colors)
mm <- melt(data, id="epoch")
ggplot(mm) +
    geom_boxplot(aes(x=factor(epoch), y=value, fill=variable))
group_means <- data %>%
    group_by(epoch) %>%
    summarise_all(funs(mean(., na.rm=TRUE)))


fit <- manova(cbind(mb, bh, bl, nh) ~ data$epoch, data)
```