

# IMS Assignment 2

By:

Stephanie Hilders (467249), English group: FEB22003X  
Giuseppe Gianfranceschi (619050), Dutch group: FEB22003

## Exercise 1

### 1.1

Import data and calculate column means, covariance matrix and set variables

```
library(knitr)
data <- read.table("data.txt")

X_bar = colMeans(data)
S <- cov(data)

n <- 100
p <- 3
```

Test  $H_0$  that  $\mu_0 = (0, 0, 0)^T$

```
u0 <- c(0, 0, 0)

alpha <- 0.05

critical_value <- ((n-1)*p)/(n-p) * qf(1-alpha, df1 = p, df2 = n-p)

T2 = as.vector(n * t(X_bar - u0) %*% solve(S) %*% (X_bar - u0))

# If true, reject H_0
(Reject_H_0 <- T2 > critical_value)
```

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

Test statistic exceeds the critical value. Therefore, reject  $H_0$  in favor of  $H_A : \mu_0 \neq (0, 0, 0)^T$

### 1.2

```
# Axes given by sqrt(eigenvalue_i * critical value / n) * eigenvector_i
eig <- eigen(S)
axes <- matrix(0, 3, 3)
colnames(axes) <- c("Axis 1", "Axis 2", "Axis 3")
rownames(axes) <- c("X1", "X2", "X3")
for (i in 1:3) {
  axes[, i] <- sqrt(eig$values[i] * critical_value / n) * eig$vectors[, i]
```

```

}

kable(axes, caption="Answer: Calculated axes for the 95% joint confidence region of mu")

```

Table 1: Answer: Calculated axes for the 95% joint confidence region of mu

	Axis 1	Axis 2	Axis 3
X1	0.2572861	-0.1599494	0.0487260
X2	0.2052942	0.0669720	-0.1330133
X3	0.1556746	0.1760325	0.0948795

### 1.3

```

# mu_i = x_bar +/- sqrt( critical_value * S[i,i] / n)
simultaneously_valid_conf_intervals <- matrix(0, 2, 3)
colnames(simultaneously_valid_conf_intervals) <- c("mu_1", "mu_2", "mu_3")
rownames(simultaneously_valid_conf_intervals) <- c("Lower bound", "Upper bound")
for (i in 1:3) {
  simultaneously_valid_conf_intervals[1,i] <- X_bar[i] - abs(sqrt(critical_value * S[i,i] / n))
  simultaneously_valid_conf_intervals[2,i] <- X_bar[i] + abs(sqrt(critical_value * S[i,i] / n))
}
kable(simultaneously_valid_conf_intervals, caption="Answer: Calculated 95% simultaneously valid confide"

```

Table 2: Answer: Calculated 95% simultaneously valid confidence intervals for components of mu

	mu_1	mu_2	mu_3
Lower bound	-0.7057383	-0.6469275	-0.8142366
Upper bound	-0.0920473	-0.1396859	-0.3073869

### 1.4

```

bonferroni_conf_intervals <- matrix(0, 2, 3)
colnames(bonferroni_conf_intervals) <- c("mu_1", "mu_2", "mu_3")
rownames(bonferroni_conf_intervals) <- c("Lower bound", "Upper bound")
for (i in 1:3) {
  bonferroni_conf_intervals[1,i] <- X_bar[i] - abs(qt(1 - alpha/(2*p), df = n-1) * sqrt(S[i,i] / n))
  bonferroni_conf_intervals[2,i] <- X_bar[i] + abs(qt(1 - alpha/(2*p), df = n-1) * sqrt(S[i,i] / n))
}
kable(bonferroni_conf_intervals, caption="Answer: Calculated 95% bonferroni confidence intervals for co"

```

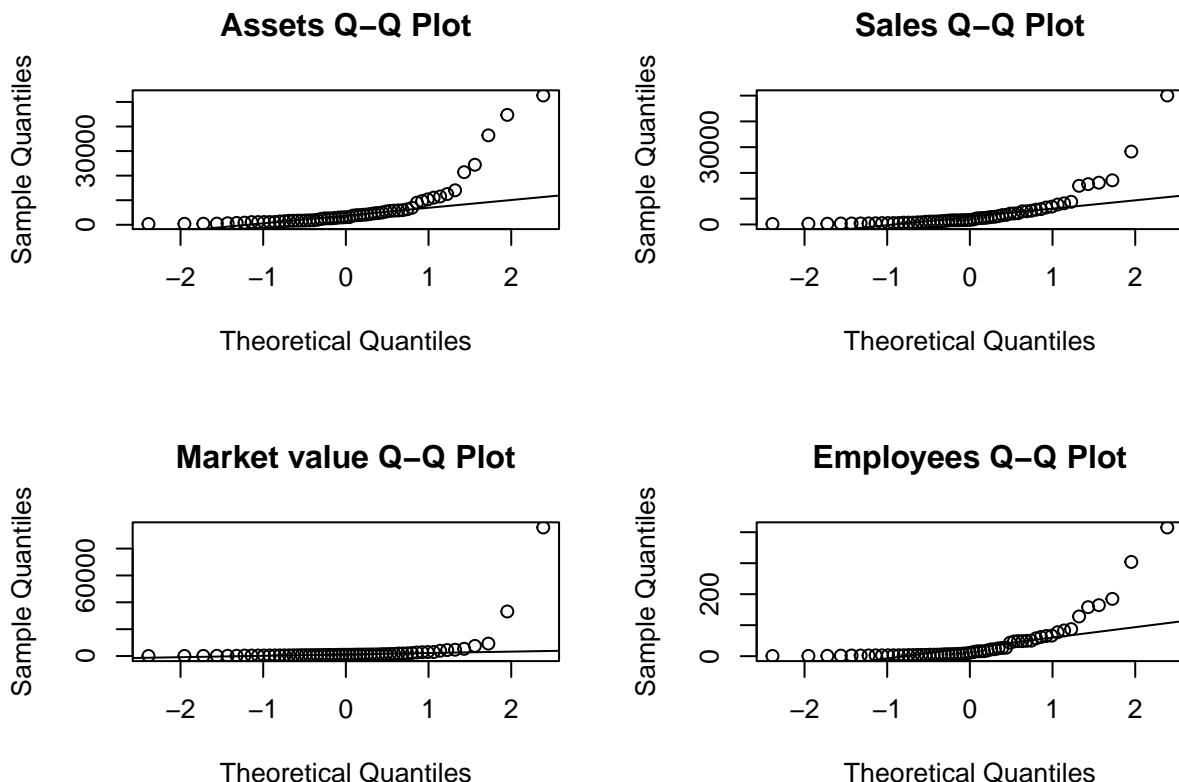
Table 3: Answer: Calculated 95% bonferroni confidence intervals for components of mu

	mu_1	mu_2	mu_3
Lower bound	-0.6588692	-0.6081882	-0.7755272
Upper bound	-0.1389164	-0.1784252	-0.3460963

## Exercise 2

### 2.1

```
MarkVal <- read.delim("MarkVal.txt")
par(mfrow = c(2,2))
qqnorm(MarkVal$Assets, main = "Assets Q-Q Plot")
qqline(MarkVal$Assets)
qqnorm(MarkVal$Sales, main = "Sales Q-Q Plot")
qqline(MarkVal$Sales)
qqnorm(MarkVal$Market_Value, main = "Market value Q-Q Plot")
qqline(MarkVal$Market_Value)
qqnorm(MarkVal$Employees, main = "Employees Q-Q Plot")
qqline(MarkVal$Employees)
```



```
# Assess joint multivariate normality
p <- 4
n <- 59
data <- as.matrix(MarkVal[,2:5])
deviations <- data - matrix(1,n,1) %*% colMeans(data)

d2 <- rep(0, n)
for (i in 1:n) {
  d2[i] <- t(deviations[i,]) %*% solve(cov(data)) %*% (deviations[i,]) # Calculate squared distances
}

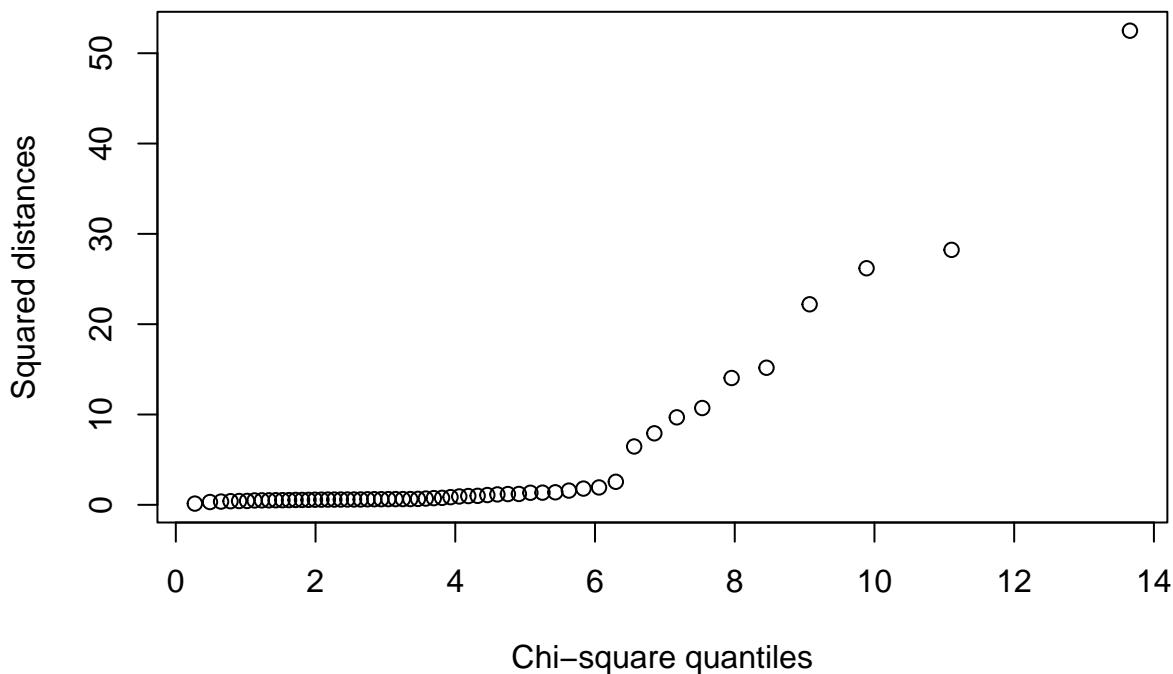
d2 <- d2[order(d2)] # Order squared distances
```

```

theoretical_quantiles <- qchisq(ppoints(n, a=0.5), df = p)
par(mfrow = c(1,1))
qqplot(theoretical_quantiles, d2,
       main = "Chi-square plot of Multivariate MarkVal Data",
       xlab = "Chi-square quantiles",
       ylab = "Squared distances")

```

## Chi-square plot of Multivariate MarkVal Data



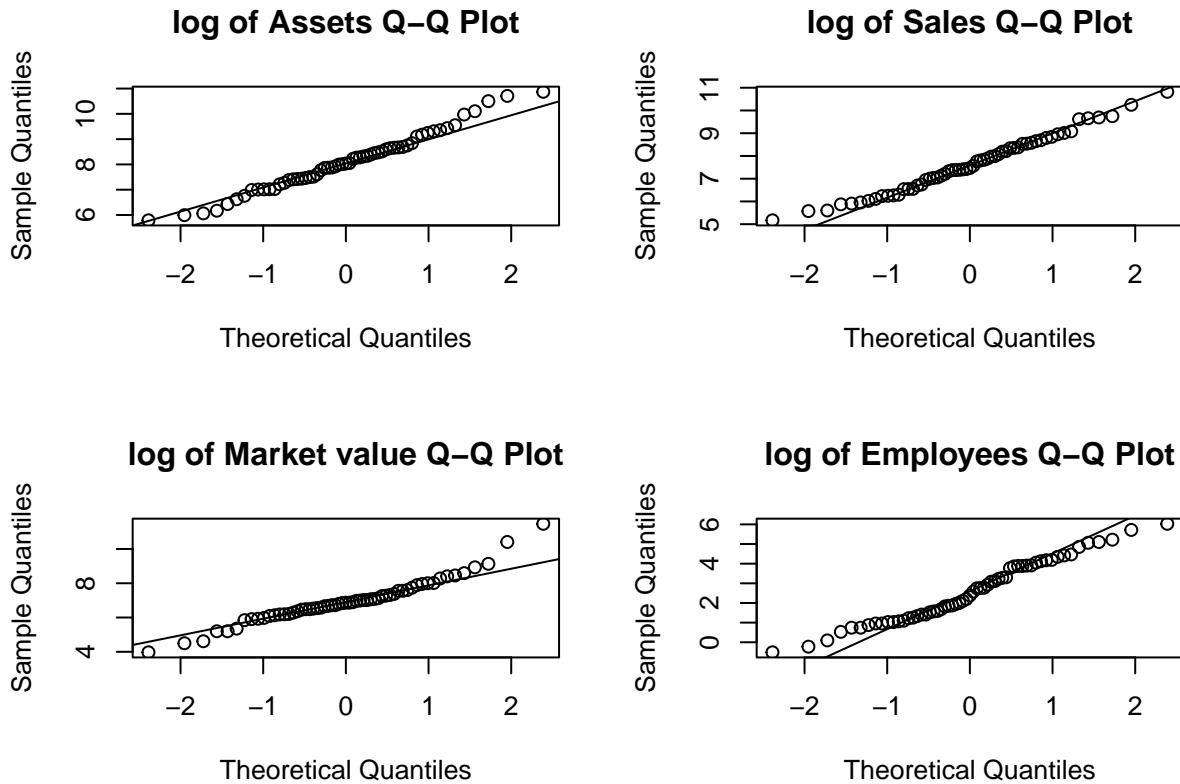
It is clear that both the individual Q-Q plots of the data and the multivariate Q-Q plot do not form a linear relation with the theoretical quantiles. Therefore, it is reasonable to assume that the data is not normally distributed.

## 2.2

```

LnMarkVal <- log(MarkVal[,2:5])
par(mfrow = c(2,2))
qqnorm(LnMarkVal$Assets, main = "log of Assets Q-Q Plot")
qqline(LnMarkVal$Assets)
qqnorm(LnMarkVal$Sales, main = "log of Sales Q-Q Plot")
qqline(LnMarkVal$Sales)
qqnorm(LnMarkVal$Market_Value, main = "log of Market value Q-Q Plot")
qqline(LnMarkVal$Market_Value)
qqnorm(LnMarkVal$Employees, main = "log of Employees Q-Q Plot")
qqline(LnMarkVal$Employees)

```



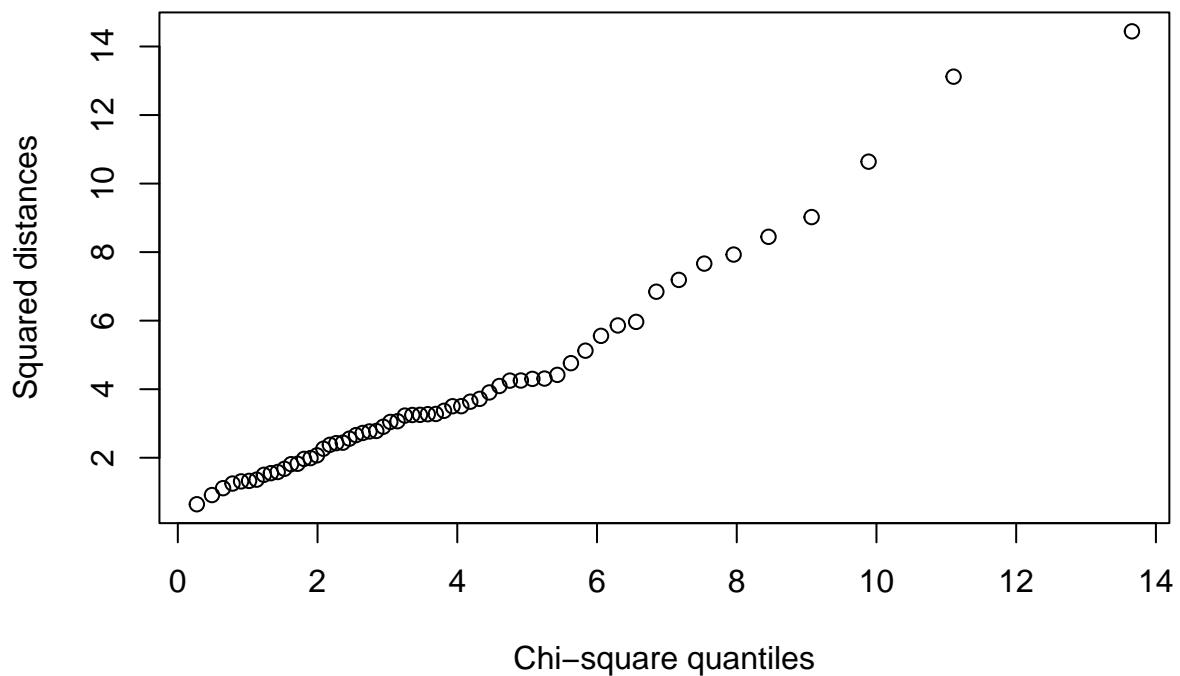
```
# Asses joint multivariate normality
p <- 4
n <- 59
lnData <- as.matrix(LnMarkVal)
lnDeviations <- lnData - matrix(1,n,1) %*% colMeans(lnData)

lnD2 <- rep(0, n)
for (i in 1:n) {
  lnD2[i] <- t(lnDeviations[i,]) %*% solve(cov(lnData)) %*% (lnDeviations[i,]) # Calculate squared distances
}

lnD2 <- lnD2[order(lnD2)] # Order squared distances

theoretical_quantiles <- qchisq(ppoints(n, a=0.5), df = p)
par(mfrow = c(1,1))
qqplot(theoretical_quantiles, lnD2,
       main = "Chi-square plot of Log of Multivariate MarkVal Data",
       xlab = "Chi-square quantiles",
       ylab = "Squared distances")
```

## Chi-square plot of Log of Multivariate MarkVal Data



From the Q-Q plots of the transformed data, a clearer linear relationship with the theoretical quantiles is evident. Both the univariate and multivariate Q-Q plots follow mostly a straight line with the theoretical quantiles. Thus, it can be assumed that the tranformed data by the natural logarithm is normally distributed.