

MS Lab 2

Aashana Nijhawan

28/11/2019

Contents

1	Question 1: Test of outliers	2
1.1	Conclude which countries can be regarded as outliers	2
1.2	Mahalanobis vs Euclidean	2
2	Question 2: Test, confidence region and confidence inter- vals for a mean vector	3
2.1	Find and sketch 95% confidence for Ellipse	3
2.2	Q-Q plots and scatter plots	5

1 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.

1.1 Conclude which countries can be regarded as outliers

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?

```
trackData = read.table("T1-9.dat")

colnames(trackData) = c("Countries", "100m", "200m", "400m", "800m", "1500m", "3000m", "Marathon")
samplesnames = c("100m", "200m", "400m", "800m", "1500m", "3000m", "Marathon")

Mahalanobis = function(obs){
  IdentityMat = diag(54)
  one_n = matrix(1,54,54) #matrix(1,7,1) %*% matrix(1,1,7)

  V = (IdentityMat - (1/54)*(one_n))

  X = as.matrix(obs)
  sample_variance = (t(X) %*% V %*% X)/54

  # sample mean formula from notes
  one = matrix(1, 54,54)
  sample_mean = (one %*% (obs))/54
  Xbar = obs - sample_mean

  ss = (Xbar) %*% solve(sample_variance) %*% t(Xbar)
  vec3 = diag(ss)
  return(vec3)
}

obs = as.matrix(trackData[,2:8])
Mahalanobis_vec = Mahalanobis(obs)
n = dim(trackData)[2] - 1 #degrees of freedom

alpha = 0.001 # 0.1% significance level

indices = 1-pchisq(Mahalanobis_vec, df=n) < alpha
trackData$Countries[indices]

## [1] KORN PNG SAM
## 54 Levels: ARG AUS AUT BEL BER BRA CAN CHI CHN COK COL CRC CZE DEN ... USA
```

1.2 Mahalanobis vs Euclidean

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

Answer: Euclidean distance only makes sense when all the dimensions have the same units (like meters or minutes in this case), since it involves adding the squared value of them. Unlike the Euclidean distance though, the Mahalanobis distance accounts for how correlated the variables are to one another. Because of high correlation between values, there is a lot of redundant information in that Euclidean distance calculation. By considering the covariance between the points in the distance calculation, we remove that redundancy. When using the Mahalanobis distance, we don't have to standardize the data like we need to do it for the Euclidean distance(which we did not do). The covariance matrix calculation takes care of this in Mahalanobis. Also, it removes redundant information from correlated variables. Even if your variables aren't very correlated it can't hurt to use Mahalanobis distance, it will just be quite similar to the results you'll get from Euclidean. Thus, Mahalanobis distance is almost always better to use than the Euclidean distance for the multivariate case.

source: <https://waterprogramming.wordpress.com/2018/07/23/multivariate-distances-mahalanobis-vs-euclidean/>

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

2.1 Find and sketch 95% confidence for Ellipse

```
library(ellipse)

##
## Attaching package: 'ellipse'
## The following object is masked from 'package:graphics':
##
##      pairs

birddata = read.table("T5-12.DAT")
# Male hook-billed kites
n = nrow(birddata)
p = dim(birddata)[2]
mu = NULL
mu = data.frame("mu1" = 190, "mu2" = 275)
meanVec = NULL
x_bar = NULL
meanMu = NULL
# x - x_bar
x_bar = data.frame("X1" = mean(birddata$V1), "X2" = mean(birddata$V2))

sample_var = function(obs){
  X = as.matrix(obs)
  IdentityMat = diag(nrow(obs))
  one_n = matrix(1,nrow(X),nrow(X))
  # sample var formula from notes
  mix = (IdentityMat - (1/nrow(X)) *(one_n))

  sample_variance = (1/nrow(X)) * (t(X) %*% mix %*% X)
  return(sample_variance)
}

S = sample_var(birddata)
S = matrix(S, ncol=2, nrow=2)
```

```

eigenValVec = eigen(S)
meanMu = as.matrix(x_bar - mu)

# To check if mu is on the confidence region we need to check if  $t^2$  is  $< F_{\{n,n-p\}}$ 

Tsqr = n * (meanMu) %*% solve(S) %*% t(meanMu)

F_val = qf(0.95,df1=p, df=n-p) * (p*(n-1)/(n-p)) #5% significance level

Tsqr<F_val

##      [,1]
## [1,] TRUE
# We conclude that it is in the region as Tsqr is less than F_val

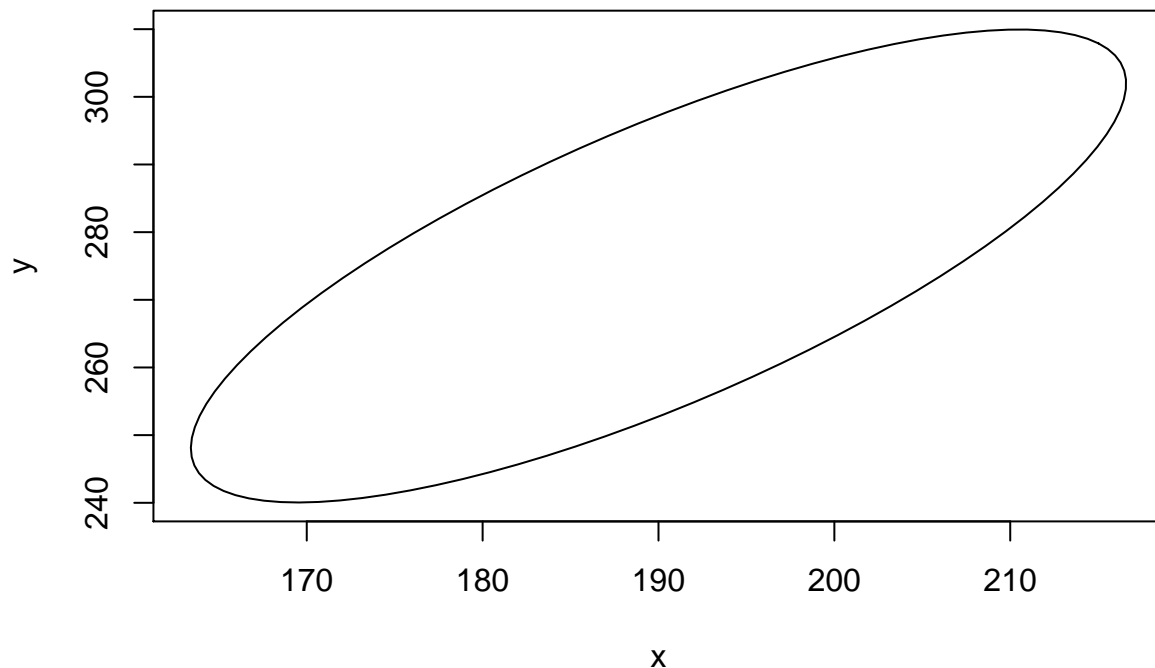
axes_len = function(lam,n,p){
  sqrt(lam)* sqrt(p*(n-1)/ (n*(n-p))) * qf(0.95,df1=p, df=n-p))
}

l1=axes_len(lam=eigenValVec$values[1],n=n,p=p)
l2=axes_len(lam=eigenValVec$values[2],n=n,p=p)

# plot(birddata)
# lines(ellipse(mu, S, npoints = 200))
# dataEllipse(birddata, levels=.95)

plot(ellipse(S,centre = c(mu$mu1,mu$mu2)),type = 'l')

```



Simultaneous T2-intervals for the component means as shadows of the confidence ellipse on the axes

```

fsqrt = function(i){
  return(sqrt(p*(n-1)/ ((n-p))) * qf(0.95,df1=p, df=n-p)) * sqrt(S[i,i]/n))
}

```

```

}

CI = NULL
CI$Lower = (x_bar$X1) - fsqrt(1)
CI$Upper = (x_bar$X1) + fsqrt(1)
CI = as.data.frame(CI)
temp = c((x_bar$X2) - fsqrt(2), (x_bar$X2) + fsqrt(2))
CI = rbind(CI, temp)
rownames(CI) = c("Mu_1_tail", "Mu_2_wing")
kableExtra::kable(CI)

```

	Lower	Upper
Mu_1_tail	189.4687	197.7758
Mu_2_wing	274.3180	285.2375

CI

```

##           Lower      Upper
## Mu_1_tail 189.4687 197.7758
## Mu_2_wing 274.3180 285.2375

```

Benoferroni's Confidence intervals

```

ben_alpha = 0.95
Ber_CI = NULL
Ber_CI$Lower = x_bar$X1 - qt(c(0.95), df=2*p-1) * sqrt(S[1,1]/n)
Ber_CI$Upper = x_bar$X1 + qt(c(0.95), df=2*p-1) * sqrt(S[1,1]/n)
Ber_CI = as.data.frame(Ber_CI)
temp = c(x_bar$X2 - qt(c(0.95), df=2*p-1) * sqrt(S[2,2]/n),
        x_bar$X2 + qt(c(0.95), df=2*p-1) * sqrt(S[2,2]/n))

Ber_CI = rbind(Ber_CI, temp)
rownames(Ber_CI) = c("Mu_1_tail", "Mu_2_wing")
Ber_CI

```

```

##           Lower      Upper
## Mu_1_tail 189.8111 197.4333
## Mu_2_wing 274.7682 284.7873

```

2.2 Q-Q plots and scatter plots

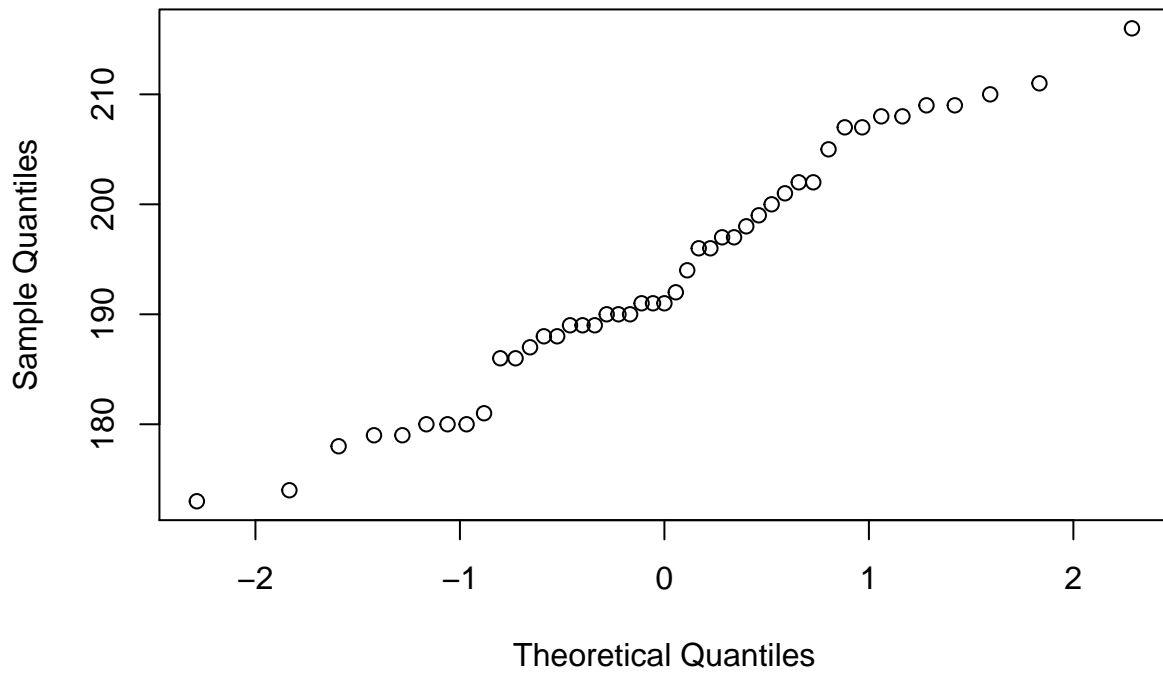
```

Scat = ggplot()+
  geom_point(aes(x=birddata$V1,y=birddata$V2))+
  ggtitle(label = "Scatter plot X1 vs X2")

qqnorm(birddata$V1,main="QQ plot for X1")

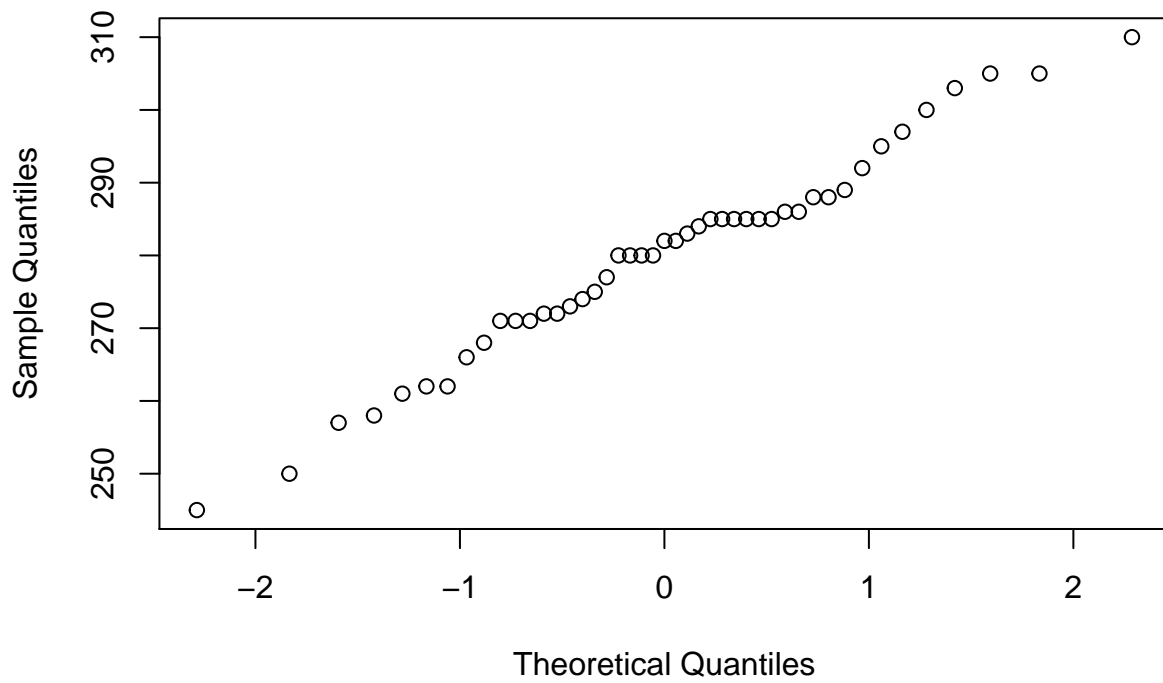
```

QQ plot for X1



```
qqnorm(birddata$V2,main="QQ plot for X2")
```

QQ plot for X2



```
plot(Scat)
```

