87/100
Very good job!

Calderoni-HW2

Ivan Calderoni

9/12/2017

Packages Used:

```
library(asbio)
library(reshape2)
library(ggplot2)
library(outliers)
library(MASS)
library(robustbase)
library(outliers)
library(fitdistrplus)
library(Amelia)
library(VIM)
library(HSAUR2)
library(devtools)
install_github("ggbiplot", "vqv")
library(ggbiplot)
library(jpeg)
```

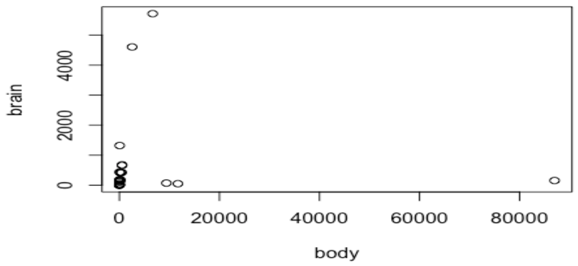
1) CONCORDANCE AND DISCORDANCE

```
x = c(3, 4, 2, 1, 7, 6, 5)
y = c(4, 3, 7, 6, 5, 2, 1)

z <- ConDis.matrix(x, y)
z

##    1  2  3  4  5  6  7
## 1 NA NA NA NA NA NA NA
## 2 -1 NA NA NA NA NA NA
## 3 -1 -1 NA NA NA NA NA
## 4 -1 -1  1 NA NA NA NA
## 5  1  1 -1 -1 NA NA NA
## 6 -1 -1 -1 -1  1 NA NA
## 7 -1 -1 -1 -1  1  1 NA


Concordant <- sum(z == 1, na.rm = TRUE)
Concordant

## [1] 6


Discordant <- sum(z == -1, na.rm = TRUE)
Discordant

## [1] 15

# There are 6 concordant pairs and 15 discordant pairs.
```

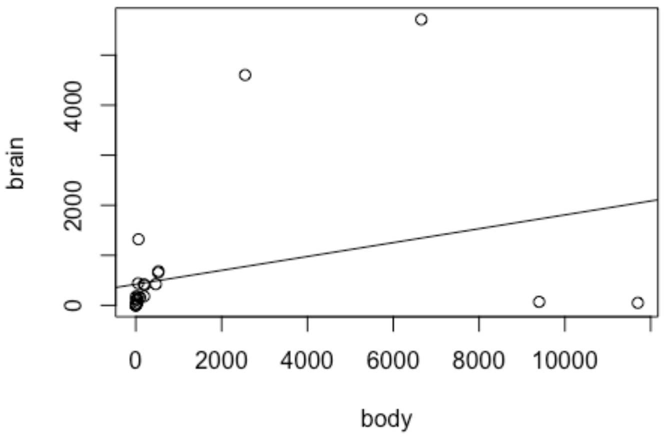
Alexander Rodríguez Castillo: ok

2) OUTLIER EXAMPLE

```
data(Animals)
plot(Animals)
```



```
grubbs.test(Animals$brain)

##
##  Grubbs test for one outlier
##
## data:  Animals$brain
## G = 3.84850, U = 0.43113, p-value = 4.985e-05
## alternative hypothesis: highest value 5712 is an outlier


grubbs.test(Animals$body)

##
##  Grubbs test for one outlier
##
## data:  Animals$body
## G = 5.019400, U = 0.032329, p-value < 2.2e-16
## alternative hypothesis: highest value 87000 is an outlier

# What is the most extreme value for brain weight?
outlier(Animals$brain)

## [1] 5712

# The most extreme value for brain weight is: 5,712 (African elephant).

# What is the most extreme value for body weight?
outlier(Animals$body)

## [1] 87000

# The most extreme value for body weight is 87,000 (Brachiosaurus).

# Which records identified?
Animals[Animals$brain==outlier(Animals$brain),]

##                  body brain
## African elephant 6654  5712

Animals[Animals$body==outlier(Animals$body),]

##                body brain
## Brachiosaurus 87000 154.5

plot(Animals)

# Add a trendline based on a linear model between brain and body weight
abline(lm(Animals$brain ~ Animals$body))
```

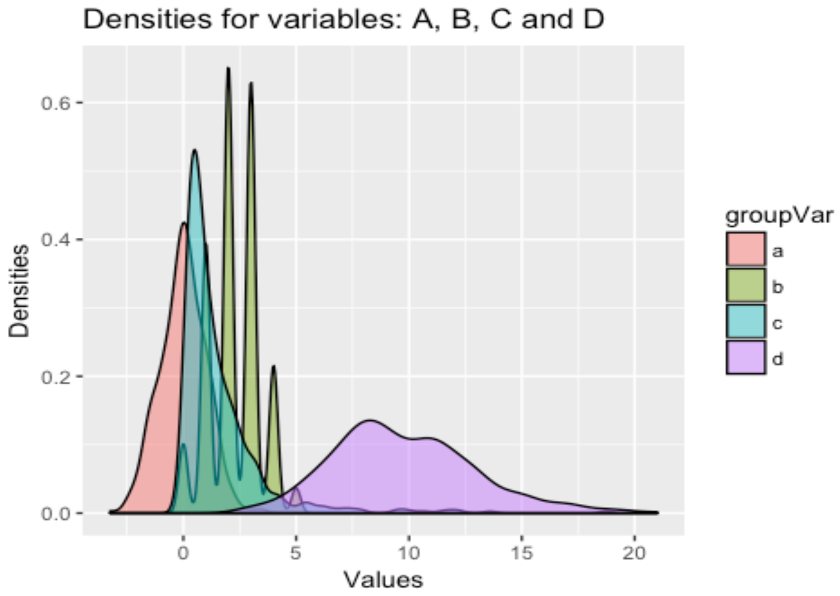


```
# What is the final animal selected in the very last step?
# Answer: HUMAN.
```

Alexander Rodríguez Castillo: good

3) GENERATING DATA AND ADVANCED DENSITY PLOTS

```
# 3 (a):
a = rnorm(500, mean = 0, sd = 1)
b = rbinom(500, size = 5, prob = 0.45)
c = rlnorm(500, meanlog = 0, sdlog = 1)
d = rpois(500, lambda = 10)

df = data.frame(a, b, c, d)
df2 <- melt(data = df, id.vars = NULL, variable.name = "groupVar")

# 3 (b):
ggplot(data=df2, aes(x=value, fill=groupVar)) + geom_density(alpha=0.5) + labs(x="Values", y="De
nsities", title="Densities for variables: A, B, C and D")
```



Alexander Rodríguez Castillo: ok

4) SHARK ATTACKS

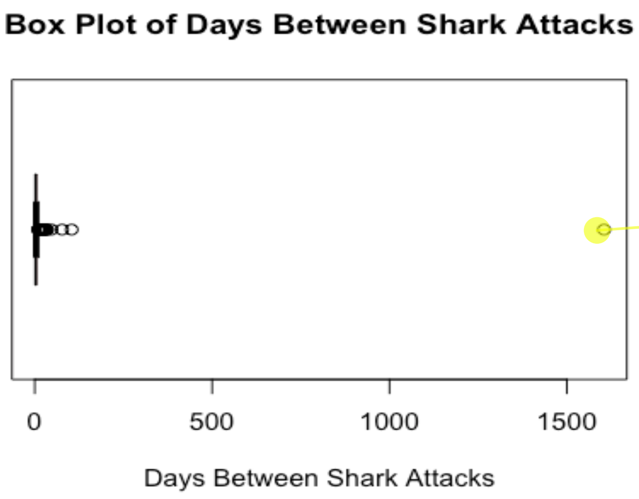
```
# 4 (a): What issues, if any, might impact your evaluation of the timeliness question of data
quality?

# Answer: The data could be incomplete or inconsistent due to the possibility of a number of
attacks that were not recorded because of standards of communication across  the different
```

Alexander Rodríguez Castillo: good

```
countries. Also, some of this data is not fully documented, there are some fields that are empty
that are important.

# 4 (b):
sharks = read.csv("ISE 5103 GSAF.csv", header = TRUE)

GSAFdata = sharks[4070:5750, ]

# 4 (c):
newDate <- as.Date(GSAFdata$Date, "%d-%b-%y")
GSAFdata <- data.frame(newDate, GSAFdata)

# 4 (d):
missing <- GSAFdata[is.na(GSAFdata$newDate), ]

# Answer: 125/1,681 = 0.0743605 which is about 7.44%. This data is not necessarily missing, but
it is formatted differently.
```

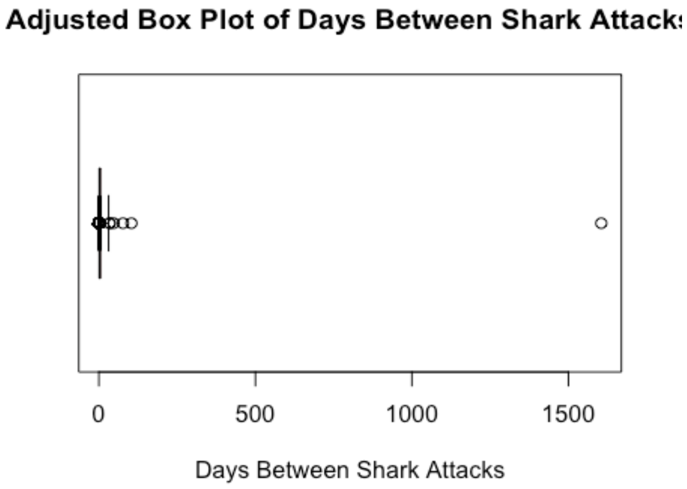
Alexander Rodríguez Castillo: ok

```
# 4 (e):
GSAFdata <- GSAFdata[!is.na(GSAFdata$newDate), ]

# 4 (f):

# i.
GSAFdata <- GSAFdata[order(GSAFdata$newDate),]
daysBetween <- diff(GSAFdata$newDate)
daysBetween <- c(0, daysBetween)
GSAFdata <- data.frame(daysBetween, GSAFdata)


# ii.
boxplot(GSAFdata$daysBetween, notch=T,col="red", horizontal=T, xlab="Days Between Shark
Attacks", main="Box Plot of Days Between Shark Attacks")
```



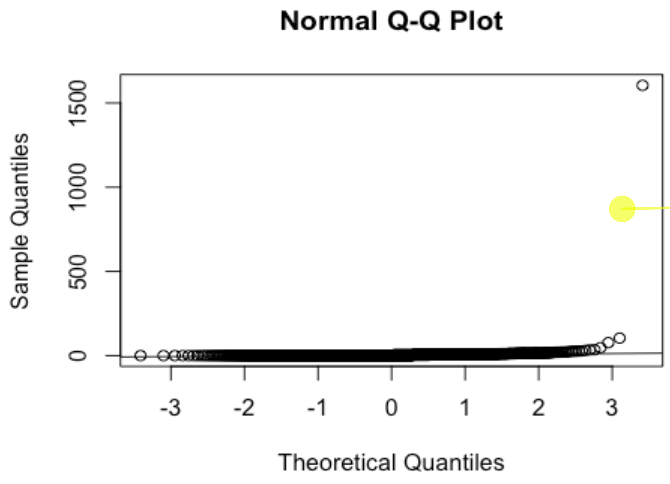
Alexander Rodríguez Castillo: this outlier comes from a miscalculation, you may have had some problems with sorting the dates

```
adjbox(GSAFdata$daysBetween, notch=T, col="red", horizontal=T, xlab="Days Between Shark
Attacks", main = "Adjusted Box Plot of Days Between Shark Attacks")
```



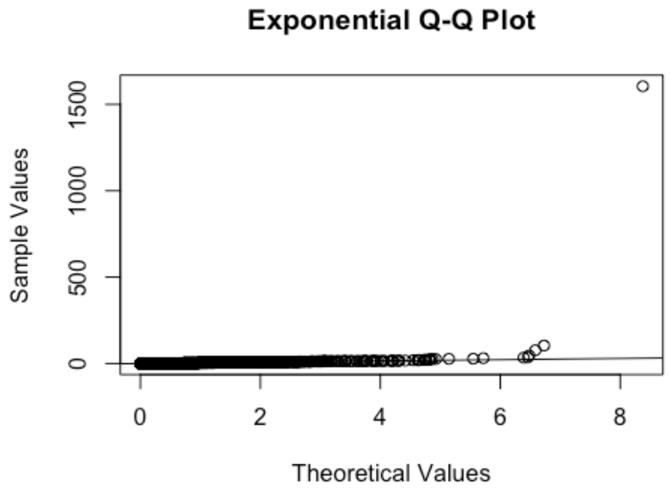
```
# iii.
grubbs.test(GSAFdata$daysBetween, type=10)

## 
##  Grubbs test for one outlier
## 
## data:  GSAFdata$daysBetween
## G = 39.081000, U = 0.017177, p-value < 2.2e-16
## alternative hypothesis: highest value 1605 is an outlier

# Answer: The Grubbs test might not be very helpful for this data set since it is large and as
the boxplots show, there are multiple outliers.

# 4 (g):
qqnorm(GSAFdata$daysBetween)
qqline(GSAFdata$daysBetween, distribution = qnorm)
```

Alexander Rodríguez Castillo: -1 Both tests are not appropriate because they assume normality.

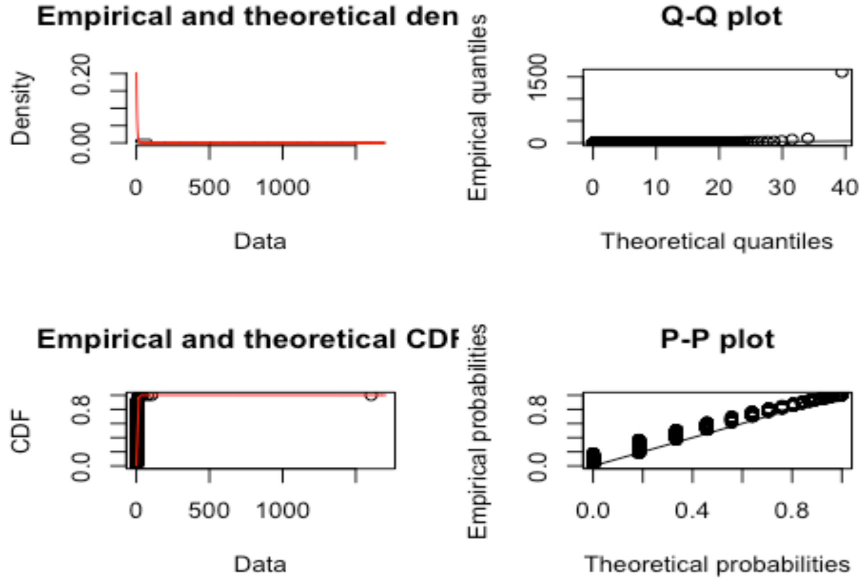


Alexander Rodríguez Castillo: this wasn't required

```
x <- rexp(1556)

qqplot(x, GSAFdata$daysBetween, main="Exponential Q-Q Plot", xlab="Theoretical Values", ylab="Sa
mple Values")

qqline(GSAFdata$daysBetween, distribution = qexp)
```



```
# 4 (h):
fitExponential <- fitdist(GSAFdata$daysBetween, "exp")
plot(fitExponential)
```

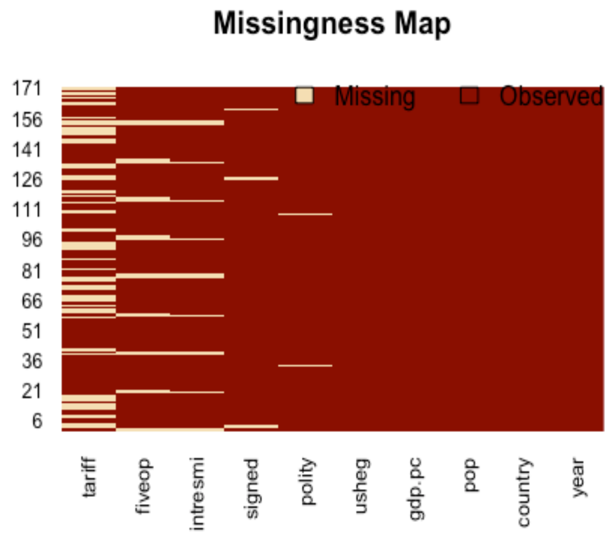


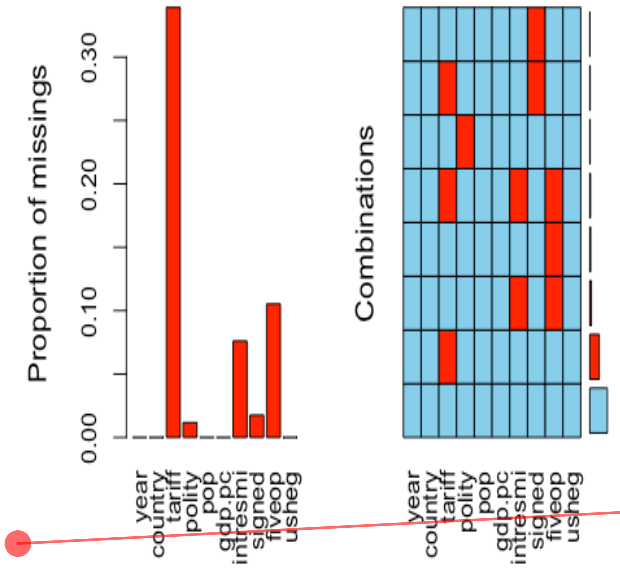
```
gofstat(fitExponential)

## Goodness-of-fit statistics
##                              1-mle-exp
## Kolmogorov-Smirnov statistic  0.1876041
## Cramer-von Mises statistic   14.6720142
## Anderson-Darling statistic          Inf
##
## Goodness-of-fit criteria
##                                1-mle-exp
## Akaike's Information Criterion  8061.577
## Bayesian Information Criterion  8066.927
```

Alexander Rodríguez Castillo: -6 didn't respond part i - How do you respond to the claim that shark attacks occur as a Poission process?

5) MISSING DATA

```
# 5 (a): Explore the "missingness" in the freetrade using your choice of methods:
data("freetrade")
missmap(freetrade, by = list(freetrade$country))
```





Alexander Rodríguez Castillo: -4 you only showed plots without any thoughts about them

```
# 5 (b):

tariffVar <- table(freetrade$country, is.na(freetrade$tariff))

chisq.test(tariffVar)
##  Pearson's Chi-squared test
##
## data:  tariffVar
## X-squared = 23.064, df = 8, p-value = 0.003283

# Chi-square test while excluding Nepal..
tariffVarNoNepal <- table(freetrade$country[freetrade$country!="Nepal"], is.na(freetrade$tariff[
freetrade$country!="Nepal"]))

chisq.test(tariffVarNoNepal)
##  Pearson's Chi-squared test
##
## data:  tariffVarNoNepal
## X-squared = 15.836, df = 7, p-value = 0.02666

# Chi-square test while excluding the Philippines..
tariffVarNoPhil <- table(freetrade$country[freetrade$country != "Philippines"], is.na(freetrade$
tariff[freetrade$country != "Philippines"]))

chisq.test(tariffVarNoPhil)
##  Pearson's Chi-squared test
##
## data:  tariffVarNoPhil
## X-squared = 11.486, df = 7, p-value = 0.1188

# Chi-square test while excluding Nepal and the Philippines...
NoNepalNoPhil <- freetrade$country != "Nepal" & freetrade$country != "Philippines"

tariffVarNoNepalNoPhil <- table(freetrade$country[NoNepalNoPhil], is.na(freetrade$tariff[NoNepal
NoPhil]))

chisq.test(tariffVarNoNepalNoPhil)
##  Pearson's Chi-squared test
##
## data:  tariffVarNoNepalNoPhil
## X-squared = 5.982, df = 6, p-value = 0.4252
```

Alexander Rodríguez Castillo: ok

Alexander Rodríguez Castillo: ok

```
# Answer: There is not sufficient evidence to support null hypothesis, which is that the tariff
is independent of country. However, when the Phillipines and Nepal are removed the p-value =
0.42.
```

Alexander Rodríguez Castillo: -1 any idea of why this happens?

6) PRINCIPAL COMPONENT ANALYSIS

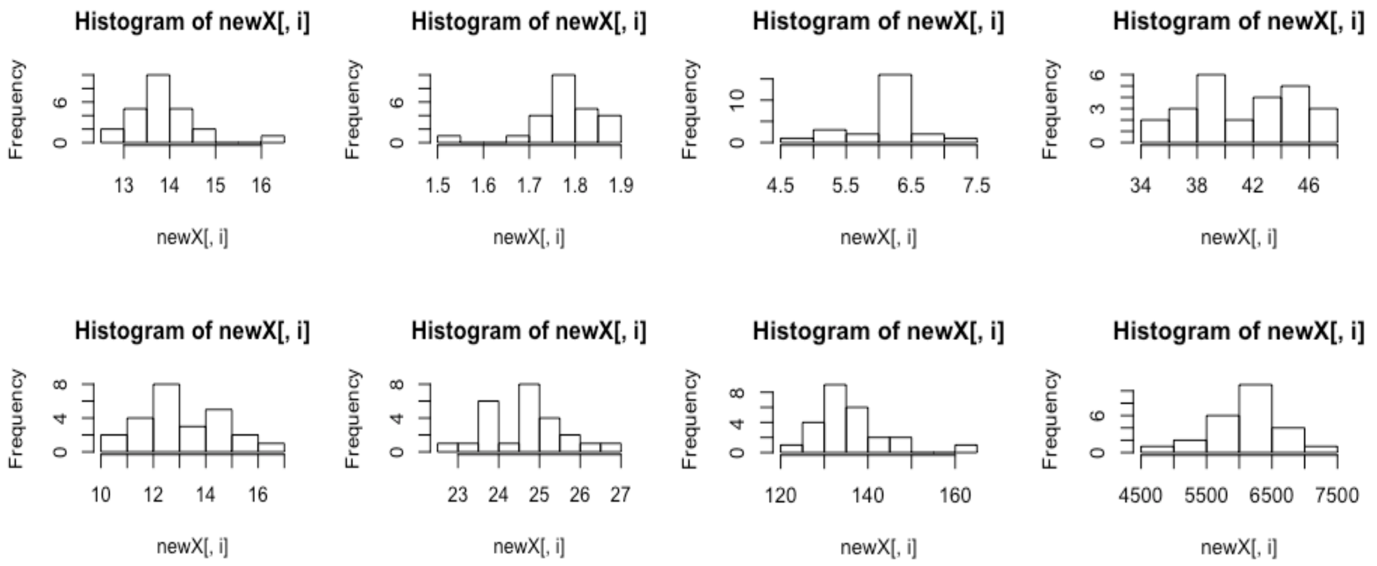
```
# 6 (a) (i):
data("mtcars")
corMat <- cor(mtcars)

# 6 (a) (ii): Compute the eigenvalues and eigenvectors of corMat.
eigenValuesAndVectors <- eigen(corMat)

# 6 (a) (iii):
prcompValues <- prcomp(mtcars, scale = TRUE)

# 6 (a) (iv):

# Answer: Both (ii) and (iii) are the same in magnitude because principal components
are the same as the eigenvector with the highest eigen value.
```

Alexander Rodríguez Castillo: ok

```
# 6 (a) (v):
PCA <- as.data.frame(prcompValues$rotation)

PCA$PC1%*%PCA$PC2

##                  [,1]
## [1,] -2.775558e-17
```

Alexander Rodríguez Castillo: ok

```
# Principal components 1 and principal components 2 are orthogonal.

# 6 (b) (i):
data("heptathlon")

par(mfrow=c(2,2))

m <- apply(heptathlon[,1:8], 2, hist)
```

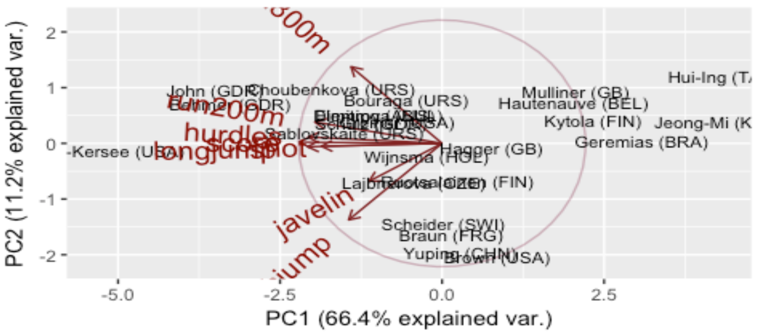


```
# Answer: The distributions seem to be reasonably normally distributed at a quick glance from
these histograms.
```

```
# 6 (b) (ii): Examine the event results using the Grubb's test.
grubbsTest <- apply(heptathlon[,1:8], 2, grubbs.test)
grubbsTest

# Answer: Launa seems to be the outlier in 5 of 8 the competitions.

heptathlon <- heptathlon[!rownames(heptathlon) %in% "Launa (PNG)", ]

# 6 (b) (iii):
heptathlon[,"hurdles"] <- max(heptathlon$hurdles) - heptathlon[,"hurdles"]
heptathlon[,"run200m"] <- max(heptathlon$hurdles) - heptathlon[,"run200m"]
heptathlon[,"run800m"] <- max(heptathlon$hurdles) - heptathlon[,"run800m"]

# 6 (b) (iv):
prcompHeptathlon <- prcomp(heptathlon, scale = TRUE)
Hpca <- as.data.frame(prcompHeptathlon$rotation)

# 6 (b) (v):
ggbiplot(prcompHeptathlon, circle = T, obs.scale = 1, varname.size = 5, labels = rownames(heptat
hlon))
```

Alexander Rodríguez Castillo: ok

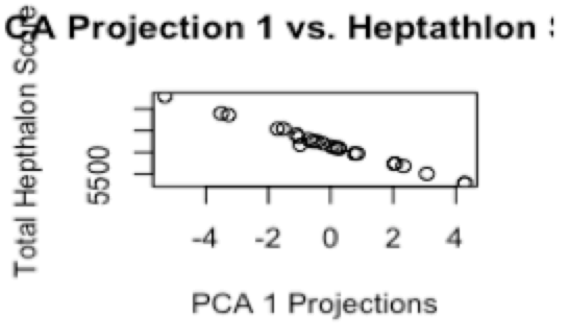
Alexander Rodríguez Castillo: You shouldn't have used scores as you the questions tells you to only input the 7 event results



```
# Answer: Hurdles, score, shot & longjump are the biggest contibuting factors for PC1.

# 6 (b) (vi):
plot(prcompHeptathlon$x[,1], heptathlon$score, main = "PCA Projection 1 vs. Heptathlon
Score", xlab = "PCA 1 Projections", ylab = "Total Hepthalon Score")

# Answer: The plot shoes that there is a strong relationship between PCA comp 1 and the
total Heptathlon score.
```

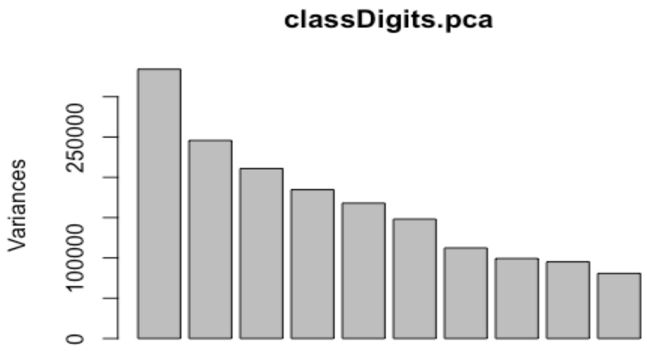
Alexander Rodríguez Castillo: ok

Alexander Rodríguez Castillo: ok

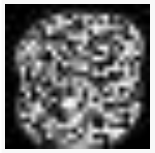


6) c) Face Recognition - well, sort of...

```
classDigits <- read.csv("ClassDigits.csv", header = TRUE)
classDigitsNoLabelCol <- classDigits[,-1]

# 6 (c) (i): Compute the eigenvectors of the digit data.
classDigits.pca <- prcomp(classDigitsNoLabelCol, scale=F)
classDigits.eigen <- classDigits.pca$rotation

plot(classDigits.pca)
```



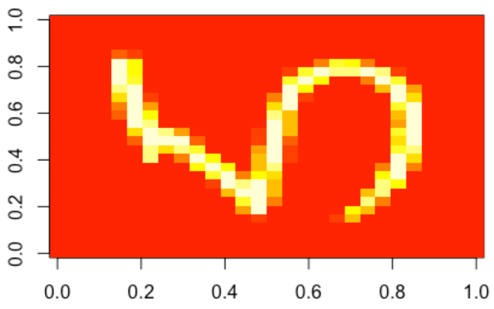
```
# 6 (c) (ii): Create a JPG image of the mean digit. Name this file meanDigit.jpg.
classDigits.mean <- colMeans(classDigitsNoLabelCol[sapply(classDigitsNoLabelCol, is.numeric)])
digitMatrix <- matrix(classDigits.mean, 28, 28, byrow=T)
writeJPEG(digitMatrix, target="meanDigit.jpg")
```

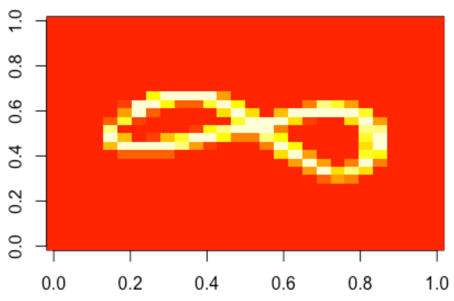


Alexander Rodríguez Castillo: ok

```
# 6 (c) (iii): iii.
image15 <- unlist(classDigitsNoLabelCol[15, ])
image15matrix <- matrix(image15, 28, 28, byrow=T)
actualImage15 <- image(image15matrix)

image100 <- unlist(classDigitsNoLabelCol[100, ])
image100matrix <- matrix(image100, 28, 28, byrow=T)
actualImage100 <- image(image100matrix)
```

Alexander Rodríguez Castillo: you had to scale from 0 to 1

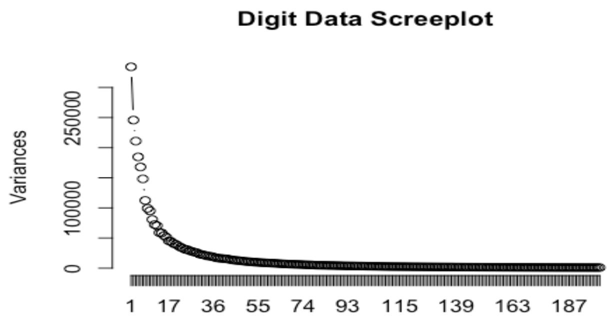




```
# image15-5
A = classDigits.pca$x[15,1:5] %*% t(classDigits.pca$rotation[,1:5]) + classDigits.mean
matrixA <- matrix(A, 28, 28, byrow=T)
writeJPEG(A, target="image15-5.jpg")
image15-5.jpg
```

Alexander Rodríguez Castillo: good

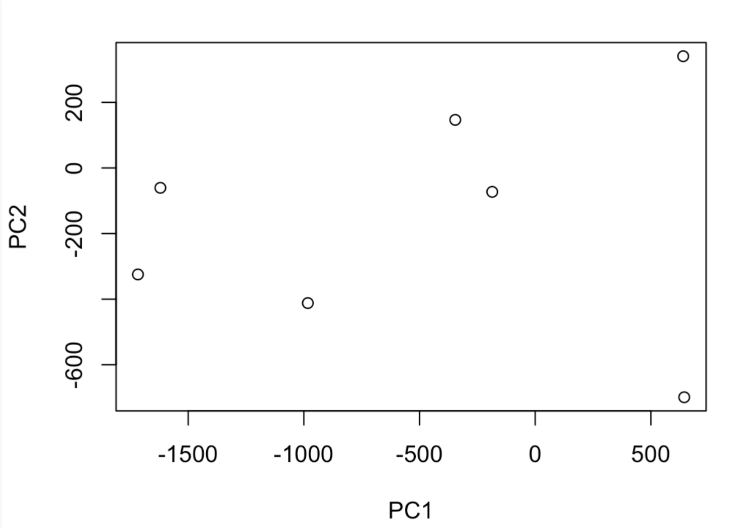
```
# image15-20
B = classDigits.pca$x[15,1:20] %*% t(classDigits.pca$rotation[,1:20])+classDigits.mean
matrixB <- matrix(B, 28, 28, byrow=T)
writeJPEG(B, target="image15-20.jpg")
image15-20.jpg


# image15-100
C = classDigits.pca$x[15,1:100] %*% t(classDigits.pca$rotation[,1:100])+classDigits.mean
matrixC <- matrix(C, 28, 28, byrow=T)
writeJPEG(C, target="image15-100.jpg")
image15-100.jpg

# image100-5
X = classDigits.pca$x[100,1:5] %*% t(classDigits.pca$rotation[,1:5]) + classDigits.mean
matrixA <- matrix(X, 28, 28, byrow=T)
writeJPEG(A, target="image100-5.jpg")
image100-5.jpg

# image100-20
Y = classDigits.pca$x[100,1:20] %*% t(classDigits.pca$rotation[,1:20]) + classDigits.mean
matrixB <- matrix(Y, 28, 28, byrow=T)
writeJPEG(B, target="image100-20.jpg")
image100-20. jpg

# image100-100
Z = classDigits.pca$x[100,1:100] %*% t(classDigits.pca$rotation[,1:100]) + classDigits.mean
matrixC <- matrix(Z, 28, 28, byrow=T)
writeJPEG(C, target="image100-100.jpg")
image100-100. jpg

# 6 (c) (iv):
screeplot(classDigits.pca2, npcs = 200, type = "lines", main="Digit Data Screeplot")
```



```
meanClassDigits <- t(meanClassDigits)

# Answer: Looking at the graph, one can tell most of the variance is covered by PC 33. So I will
pick my k to be 33.
head(mahaDistance)

[1] 2.124739 0.000000 0.000000 0.000000 0.000000 0.000000

plot(weightsClass7Test)
```

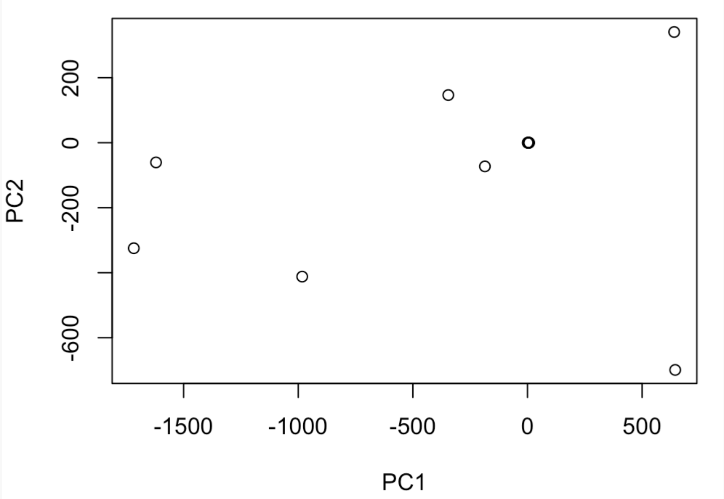
Alexander Rodríguez Castillo: ok

Alexander Rodríguez Castillo: -1 your covariance for the Mahalanobis distance should be from the train digit data; other than that, it looks ok your code

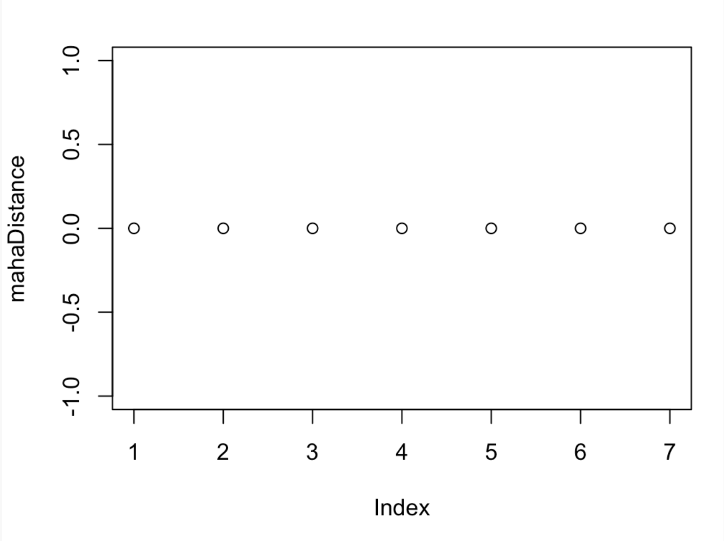


Alexander Rodríguez Castillo: don't understand what this is for... if you are going to put some random graph, explain it

```
points(mahaDistance)
```



```
plot(mahaDistance)
```



```
# 6 (c) (iv):
```

```
[1] 4
[1] 11
[1] 2
```