Data Exploration & Visualization (Calderoni-HW2)

Ivan Calderoni

9/12/2017

Packages Used:

```
# asbio contains the Con.Dis.matrix() function needed for to calculate
# concordant and discordant pairs
library(asbio)

## Loading required package: tcltk

# needed to use the melt function
library(reshape2)
library(ggplot2)
# load "outliers" package to 'test' for outliers
library(outliers)
# to get the "Animals" dataset
library(MASS)
# needed to create the adjusted box plot
library(robustbase)
# grubbs test
library(outliers)
library(fitdistrplus)

## Loading required package: survival

##
## Attaching package: 'survival'

## The following object is masked from 'package:robustbase':
##
##     heart

# contains the freetrade dataframe needed for problem 5
library(Amelia)

## Loading required package: Rcpp

## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.4, built: 2015-12-05)
## ## Copyright (C) 2005-2017 James Honaker, Gary King and Matthew
Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```r
# to use the "aggr" function
library(VIM)

## Loading required package: colorspace

## Loading required package: grid

## Loading required package: data.table

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:reshape2':
##
##     dcast, melt

## VIM is ready to use.
##  Since version 4.0.0 the GUI is in its own package VIMGUI.
##
##           Please use the package to use the new (and old) GUI.

## Suggestions and bug-reports can be submitted at:
https://github.com/alexkowa/VIM/issues

##
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':
##
##     sleep

# the following package has a dataset needed for problem 5 (b)
library(HSAUR2)

## Loading required package: tools

##
## Attaching package: 'HSAUR2'

## The following object is masked from 'package:robustbase':
##
##     epilepsy

library(devtools)
install_github("ggbiplot", "vqv")

## Warning: Username parameter is deprecated. Please use vqv/ggbiplot

## Skipping install of 'ggbiplot' from a github remote, the SHA1
(7325e880) has not changed since last install.
##   Use `force = TRUE` to force installation

library(ggbiplot)
```

```
## Loading required package: plyr

## Loading required package: scales
```

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)

# View z:
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

# This will add all the 1's in the matrix which is the total number of
# concordant pairs.
Concordant <- sum(z == 1, na.rm = TRUE)

# Print the total of Concordant pairs:
Concordant

## [1] 6

# This will add all the -1's in the matrix which is the total number
# of discordant pairs.
Discordant <- sum(z == -1, na.rm = TRUE)

# Print the total of Concordant pairs:
Discordant

## [1] 15

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

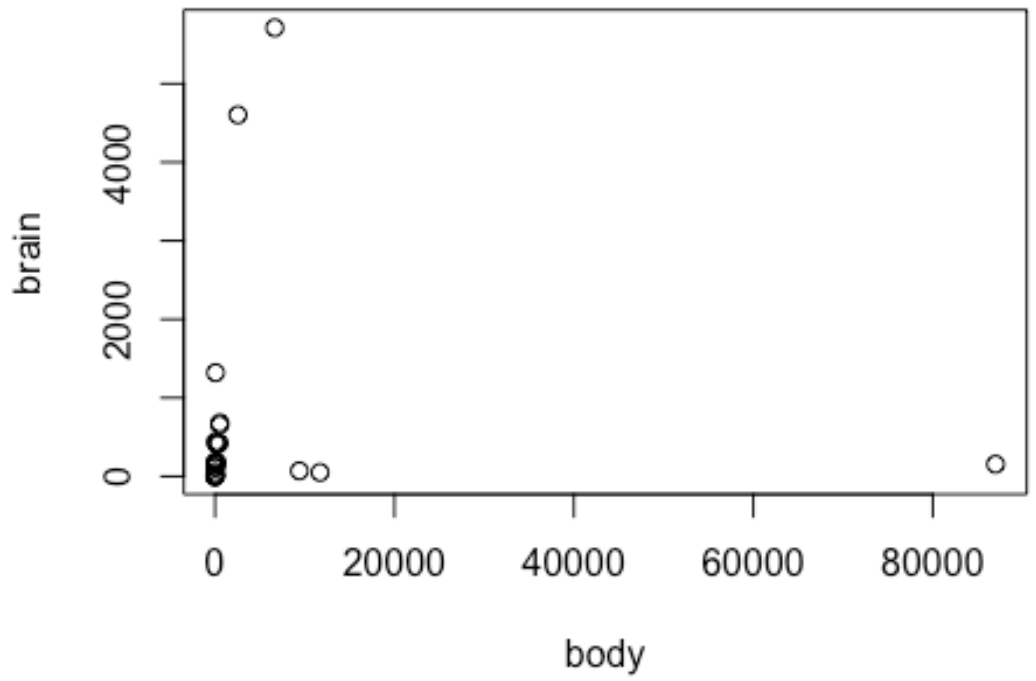
2) OUTLIER EXAMPLE

```
# Outlier example and some R functionality including basic bivariate
# outlier labeling.

# Load the "Animals" dataset from the MASS package:
data(Animals)
```

```r
# Look at the first few records:
head(Animals)
```

```
##                       body brain
## Mountain beaver       1.35   8.1
## Cow                 465.00 423.0
## Grey wolf            36.33 119.5
## Goat                 27.66 115.0
## Guinea pig            1.04   5.5
## Dipliodocus       11700.00  50.0
```

```r
# Plots animal's brain weight vs. body weight:
plot(Animals)
```




```r
# Use the interactive functionality to identify:
# identify(Animals)

# A few data points you think might be outliers
# - just scroll the mouse over the plot and left-click on the points
# - press ESC when finished

# Do it again, but this time the names of the animals will display:
# plot(Animals)
```

```r
# Can label, and save indices in vector:
# v <- identify(Animals,labels=row.names(Animals))

# These are the records you selected with the mouse:
# Animals[v,]

# We could easily delete all of these values:
# Animals<-Animals[-v,]

# and repeat the process...

# Plots animal's brain weight vs. body weight:
# plot(Animals)

# Interactivley point and click:
# v <- identify(Animals,labels=row.names(Animals))

# Display animals selected when finished:
# Animals[v,]

# Let's try something else...

# Re-load full data set and  examine the outliers:
data(Animals)

# Univariate test for 'brain' outliers:
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

# Univariate test for 'body' outliers:
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
```

```r
# 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
```

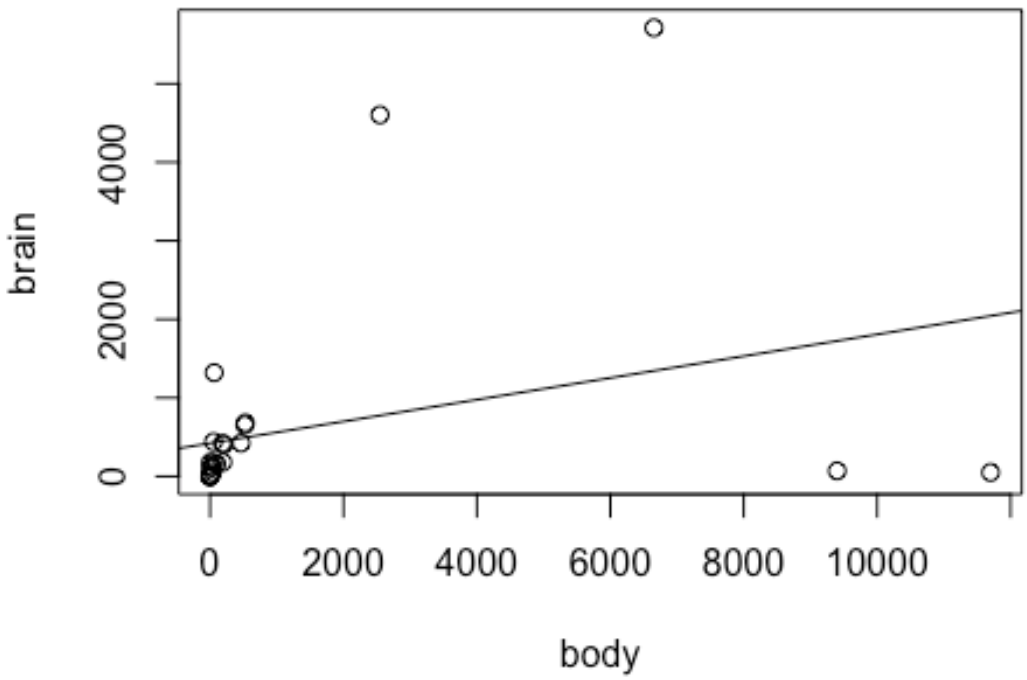
```r
# The most extreme value for body weight is 87,000 (Brachiosaurus).

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

```
##                   body brain
## African elephant 6654  5712
```

```r
# Identified: African elephant.

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

```
##                 body brain
## Brachiosaurus 87000 154.5
```

```r
# Identified: Brachiosaurus.

# Let's remove the body weight outlier.

# Return all records in Animals EXCEPT for the record that has the body
# weight equal to the value of outlying body weigth.
Animals<-Animals[Animals$body!=outlier(Animals$body),]

plot(Animals)

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

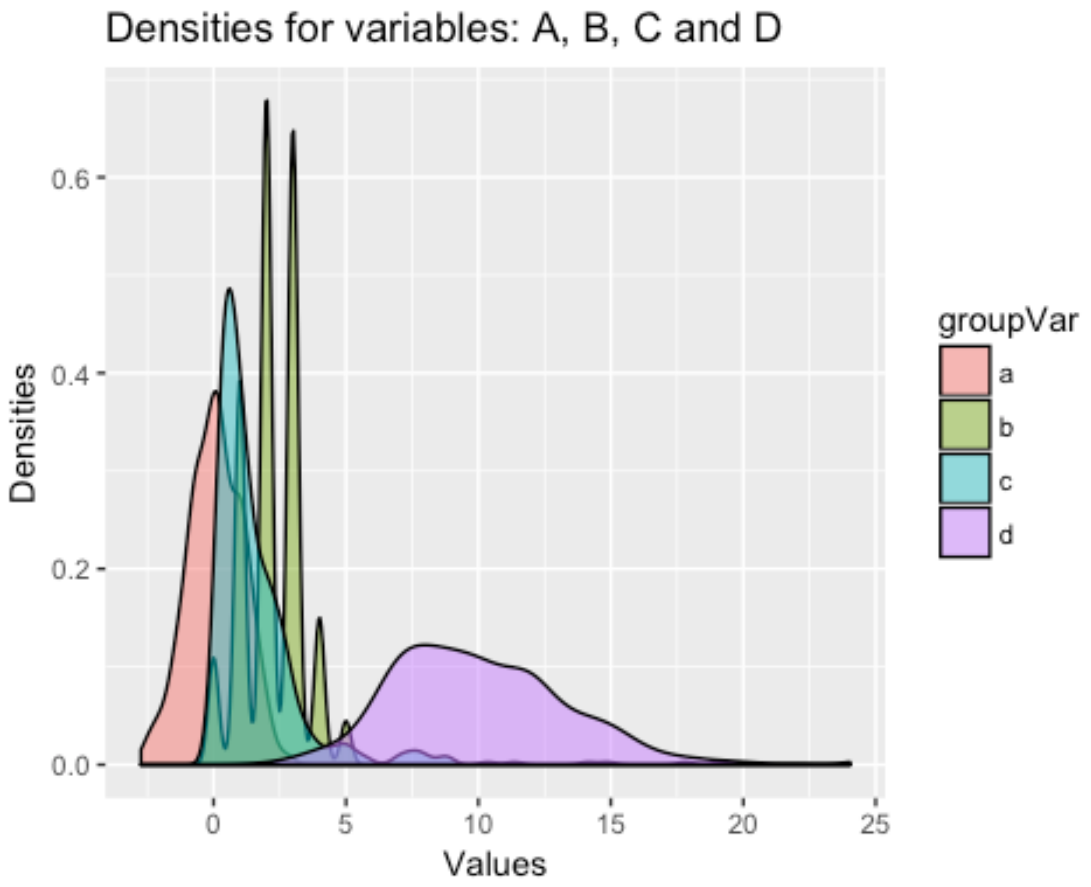



```
# The scatter plot with a simple regression line allows us to
# visualize bivariate outliers -- that is, data points that are far
# from the trendline.

# Pick the four furthest points from the line..
# v <- identify(Animals$body,Animals$brain,labels=row.names(Animals))
# Animals[v,]

# And delete if you want...
# Animals <- Animals[-v,]

# Plot what ever is left:
# plot(Animals)

# And the new model looks like this:
# abline(lm(Animals$brain ~ Animals$body))

# Now please identify the most extreme bivariate data point in the
# resulting plot:
# v <- identify(Animals$body,Animals$brain,labels=row.names(Animals))
# Animals[v,]
```

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

3) GENERATING DATA AND ADVANCED DENSITY PLOTS

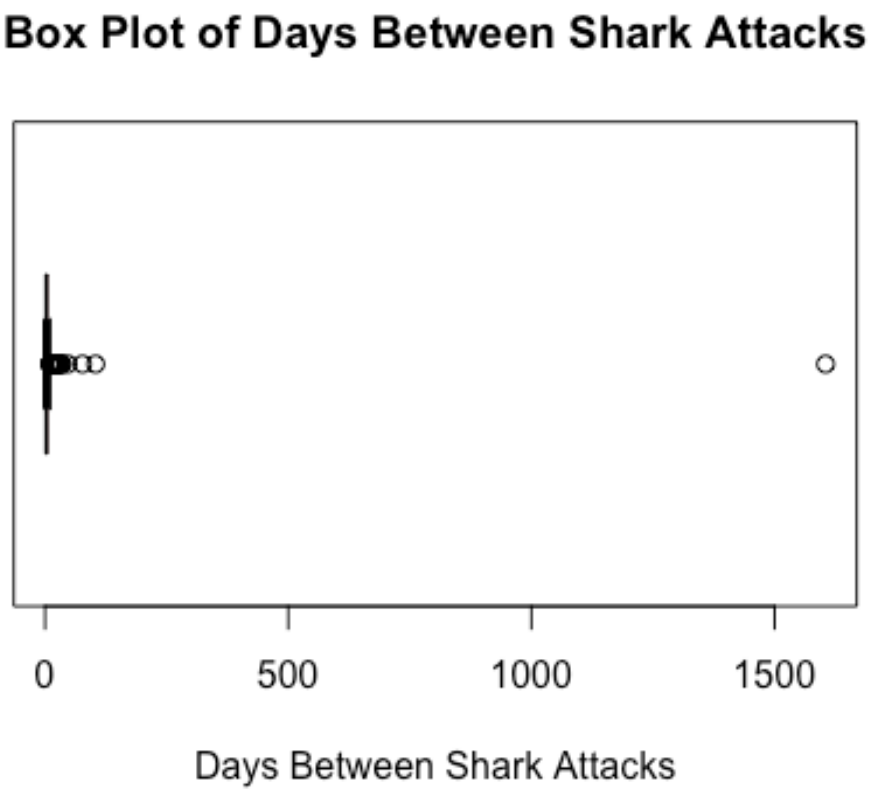
```
# 3 (a): Create a data frame df with 500 rows and 4 variables: a, b, c,
# and d. Each variable should contain data generated randomly from a
# different type of distribution.

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): Using df2, plot the densities of each distribution overlaid on
# each other on one plot. Each density should have some level of
# transparency and be colored differently.

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



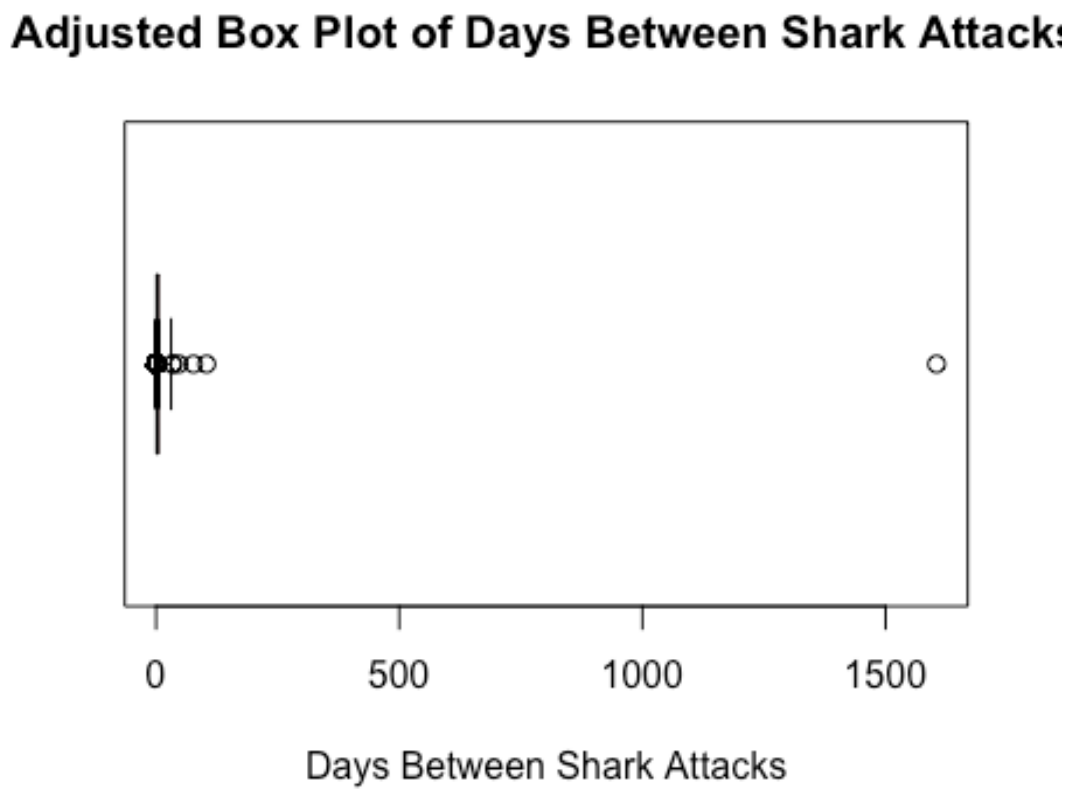
4) SHARK ATTACKS

```
# 4 (a): The data contains historical information ranging from early
dates till August, 2015. 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 countries. Also,
# some of this data is not fully documented, there are some fields that
# are empty that are important.

# 4 (b): Create a new data frame, GSAFdata, which contains incidents
# occurring on or after the year 2000:

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

# Here, I am creating a new data frame, GSAFdata, which contains rows
# from 4,070 through the end of the data frame "sharks."
GSAFdata = sharks[4070:5750, ]

# 4 (c): The Date field is currently stored as character field and
```

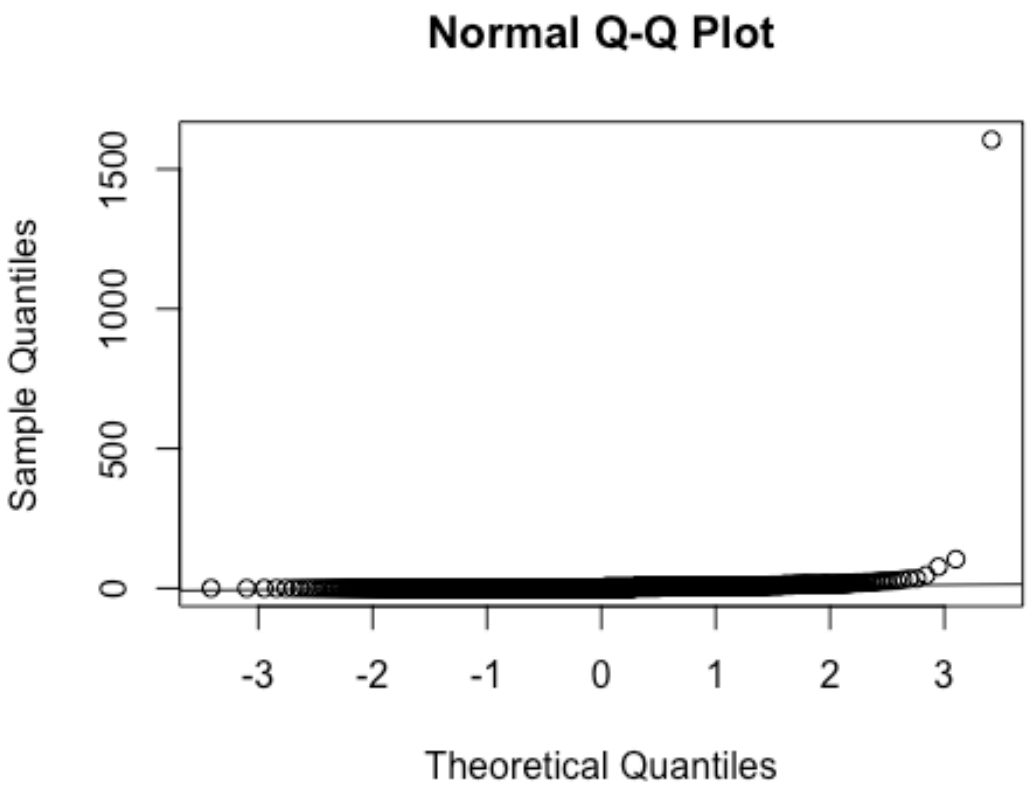
```r
# listed as a "factor". Use the as.Date command to create a new
# variable in the data frame which converts the factor to an R date
# type.

# We use 'b' instead of 'm' because the months are abbreviated:
newDate <- as.Date(GSAFdata$Date, "%d-%b-%y")

# Now append newDate to GSAFdata:
GSAFdata <- data.frame(newDate, GSAFdata)

# 4 (d): What percentage of the new date field is missing? Why is the
# data missing?
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.

# 4 (e): Delete all of the records in GSAFdata that have missing values
# for the new date field.
GSAFdata <- GSAFdata[!is.na(GSAFdata$newDate), ]

# 4 (f):

# i. Use the diff command to help you create a vector daysBetween with
# days between attacks. Notice that the vector daysBetween will have
# one less value than the number of rows in GSAFdata. Add a missing
# value as the first element of daysBetween and add the revised vector
# as a new variable in GSAFdata.
GSAFdata <- GSAFdata[order(GSAFdata$newDate),]

daysBetween <- diff(GSAFdata$newDate)

# Adding a 0 as the first element of daysBetween:
daysBetween <- c(0, daysBetween)

GSAFdata <- data.frame(daysBetween, GSAFdata)

# ii. Run and comment on the results from boxplot and adjbox for
# GSAFdata$daysBetween.

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



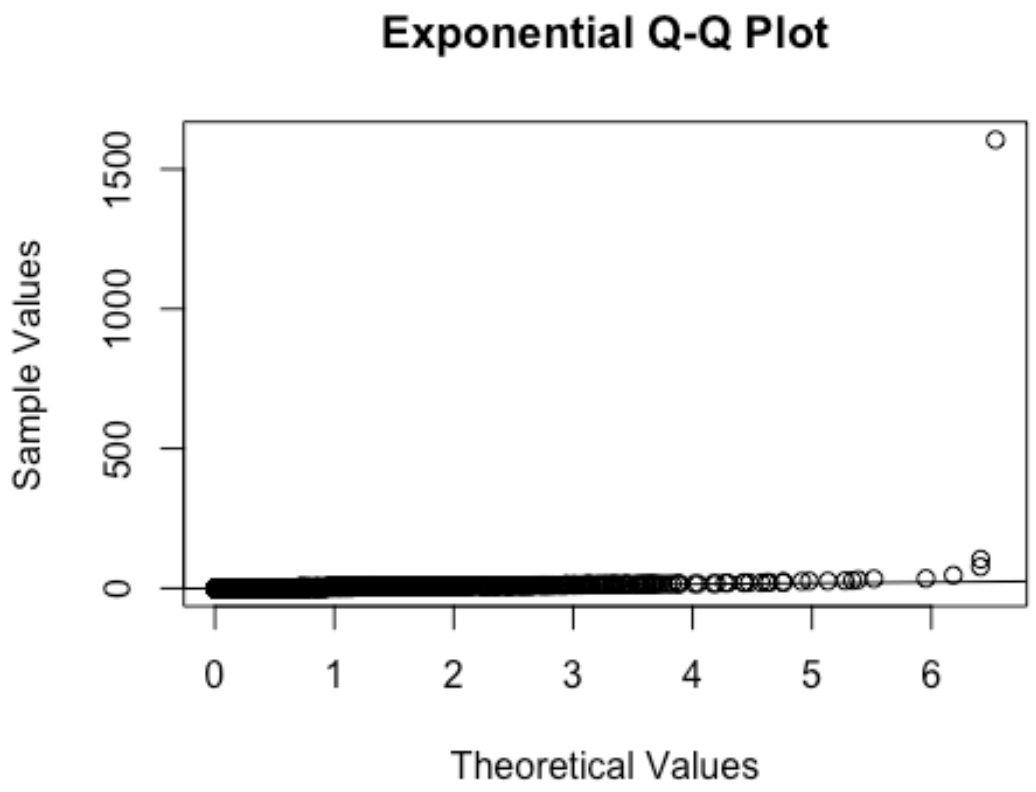

```
adjbox(GSAFdata$daysBetween, notch=T, col="red", horizontal=T,
xlab="Days Between Shark Attacks", main = "Adjusted Box Plot of Days
Between Shark Attacks")
```



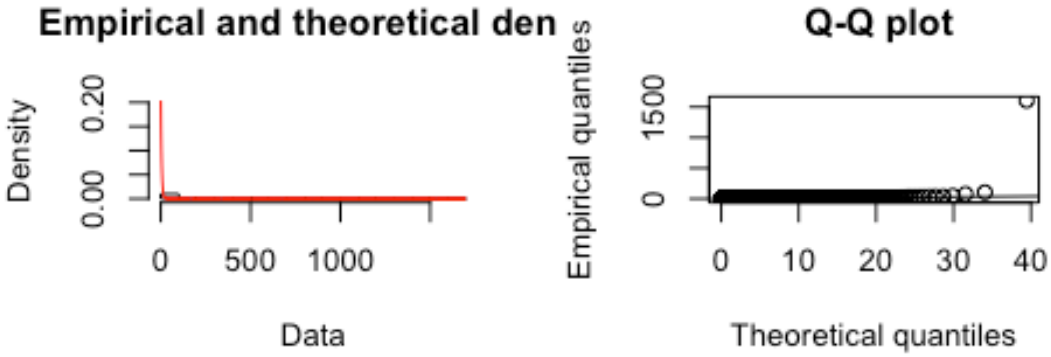

```
# iii. Is the Grubb's test, the Generalized ESD test, both, or neither
# appropriate for this data?
grubbs.test(GSAFdata$daysBetween, type=10)

##
##  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

# 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)
```

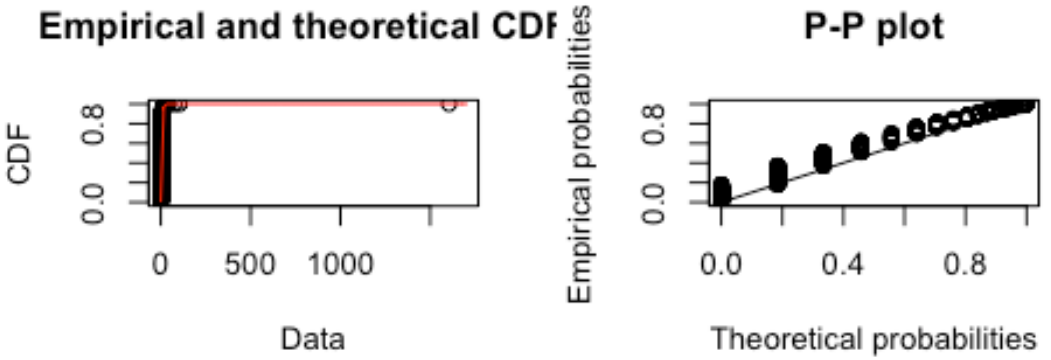



```r
x <- rexp(1556)

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

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




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

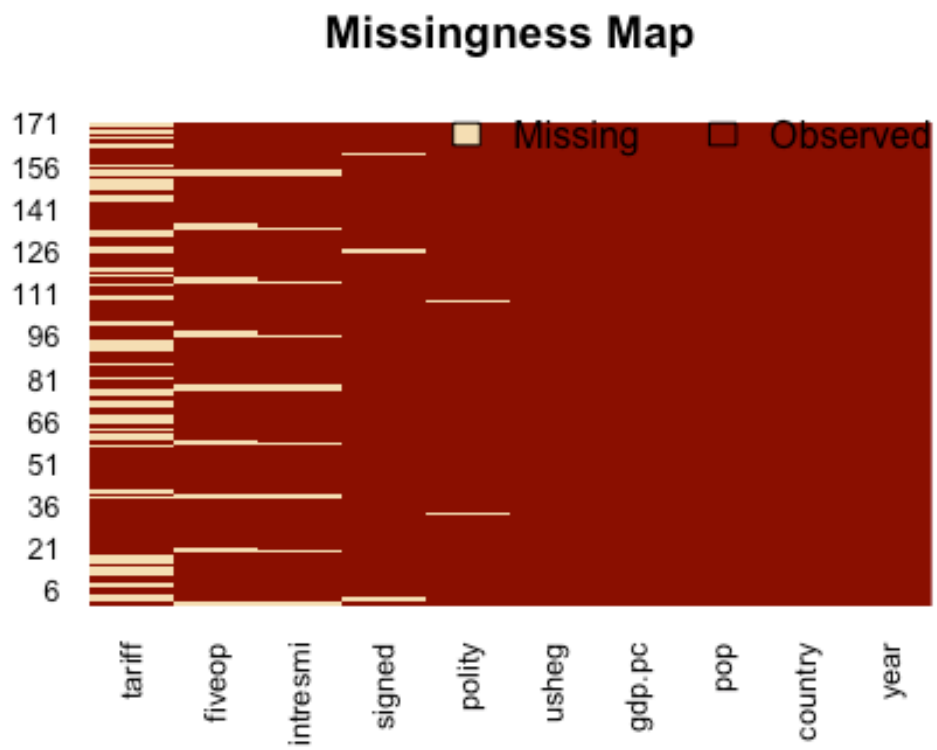




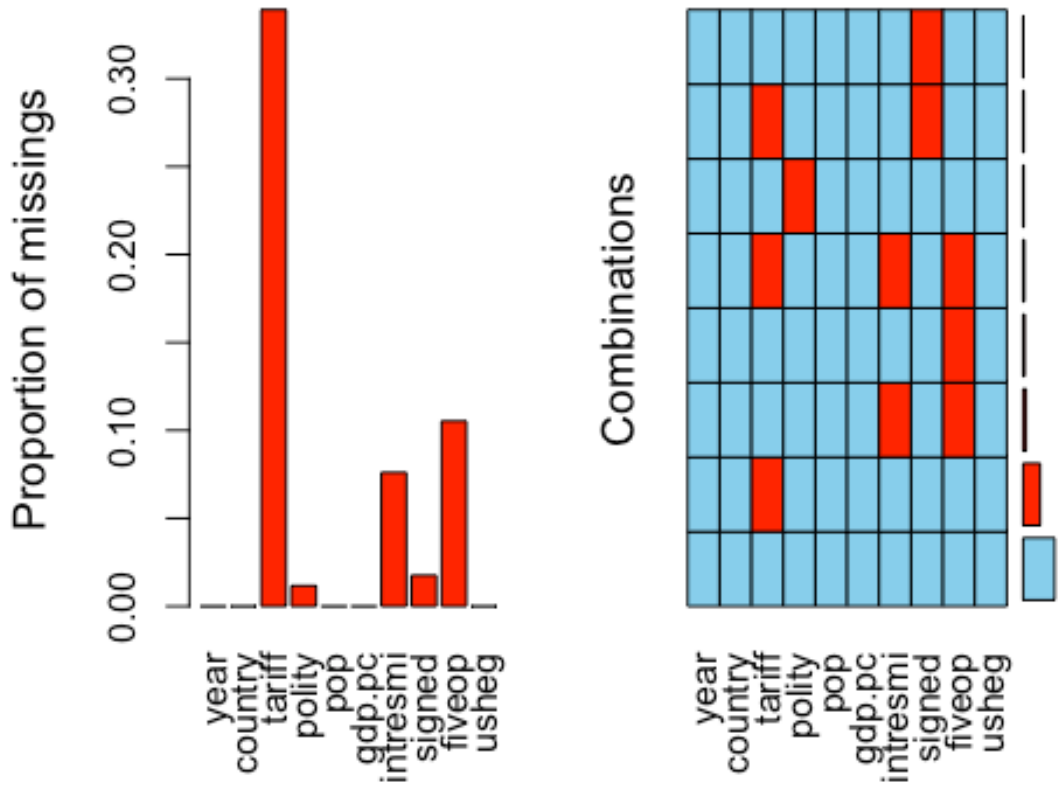

```
gofstat(fitExponential)

## 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
```

5) MISSING DATA

```
# 5 (a): Explore the "missingness" in the freetrade using your choice
# of methods:
data("freetrade")

# missmp (missingness map) plots a map showing where the missingness
# occurs in the dataset passed to Amelia.
missmap(freetrade, by = list(freetrade$country))
```




```r
# aggr (aggregations for missing/imputed values) plots the a mount of
# missing/imputed values in each variable and the amount of
# missing/imputed values in certain combinations of variables.
aggr(freetrade)
```



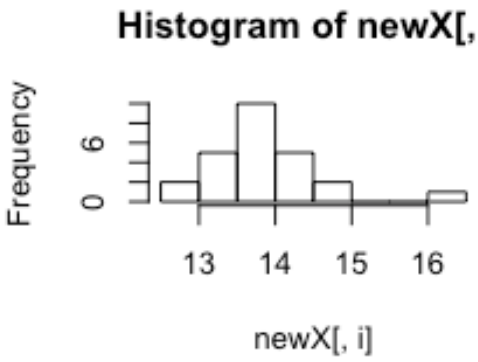

```
# 5 (b): Implement your own statistical test to determine if the
# missingness in the tariff variable is independent with the country
# variable. Does your answer change if you remove Nepal or if you
# remove the Philippines?

# Test to be used to determine if the missingness in the variable
# tariff is independent with the country variable: CHI-SQUARE TEST.
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)
```

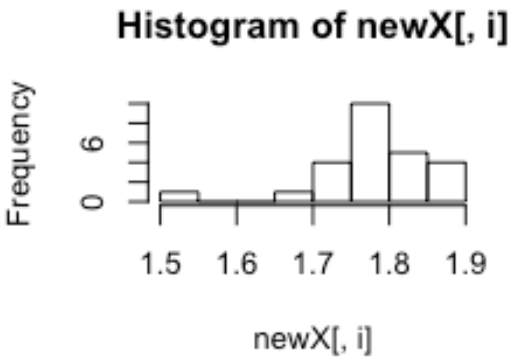
```
##
##  Pearson's Chi-squared test
##
## data:  tariffVarNoNepal
## X-squared = 15.836, df = 7, p-value = 0.02666

# 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[NoNepalNoPhil]))

chisq.test(tariffVarNoNepalNoPhil)

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

# 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.
```

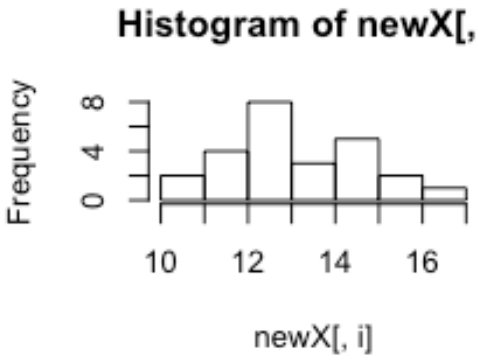
6) PRINCIPAL COMPONENT ANALYSIS

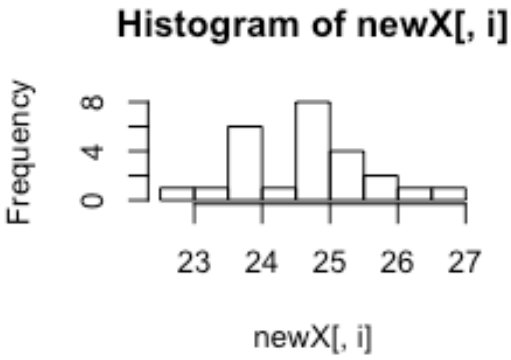
```
# 6 (a): Mathematics of principal components

# 6 (a) (i): Using the data mtcars, create the correlation matrix of
# all the attributes and store the results in a new object corMat.
data("mtcars")

corMat <- cor(mtcars)

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

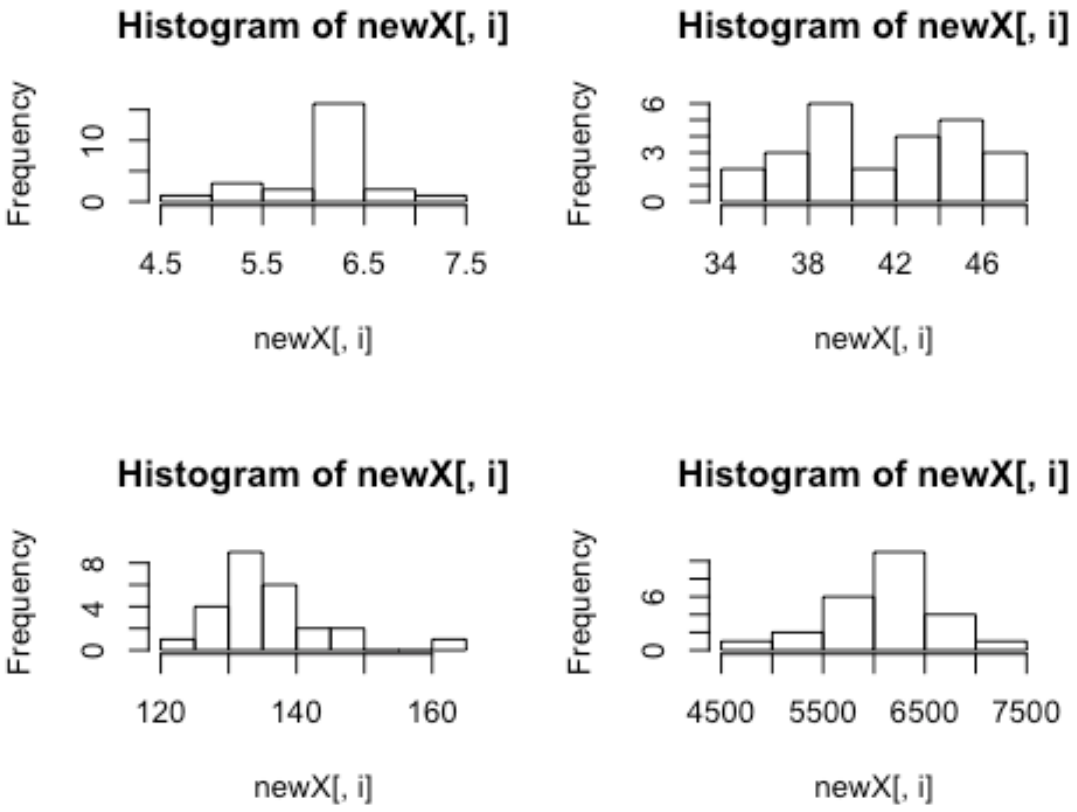
```r
# 6 (a) (iii): Use "prcomp" to compute the principal components of the
# "mtcars" attributes.
prcompValues <- prcomp(mtcars, scale = TRUE)

# 6 (a) (iv): Compare the results from (ii) and (iii).

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

# 6 (a) (v): Using R demonstrate that principal components 1 and 2 from
# (iii) are orthogonal.
PCA <- as.data.frame(prcompValues$rotation)

PCA$PC1%*%PCA$PC2

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

# Principal components 1 and principal components 2 are orthogonal.

# 6 (b): The HSAUR2 package contains the data heptathlon which are the
# results of the women's olympic heptathlon competition in Seoul, Korea
# from 1988.
data("heptathlon")

# 6 (b) (i): Look at histograms of each numerical variable using
# apply(heptathlon[,1:8],2,hist) (note: these are not labeled well, but
# that is okay for now since you just want to take a quick look at the
# distributions). From this quick inspection, are the distributions
# reasonably normal?
par(mfrow=c(2,2))

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


Histogram of newX[, i]
Frequency
0
6
13
14
15
16
newX[, i]


Histogram of newX[, i]
Frequency
0
6
1.5
1.6
1.7
1.8
1.9
newX[, i]


Histogram of newX[, i]
Frequency
0
4
8
10
12
14
16
newX[, i]


Histogram of newX[, i]
Frequency
0
4
8
23
24
25
26
27
newX[, i]

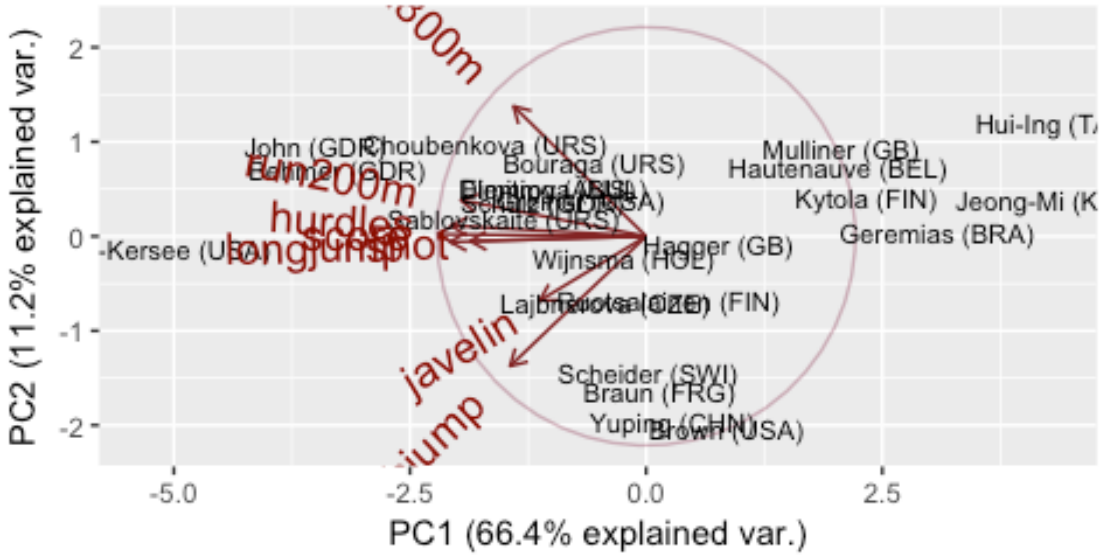



```
# The distributions seem to be reasonably normally distributed at a
# quick glance from these histograms.

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

grubbsTest

## $hurdles
##
##  Grubbs test for one outlier
##
## data:  newX[, i]
## G.Launa (PNG) = 3.5024, U = 0.4676, p-value = 0.000436
## alternative hypothesis: highest value 16.42 is an outlier
##
##
## $highjump
##
##  Grubbs test for one outlier
##
## data:  newX[, i]
## G.Launa (PNG) = 3.61810, U = 0.43184, p-value = 0.0001698
```

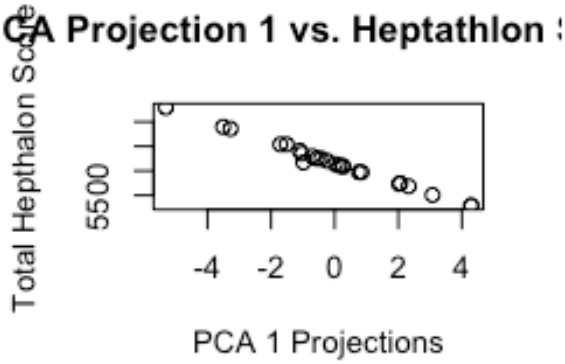
```
## alternative hypothesis: lowest value 1.5 is an outlier
##
##
## $shot
##
##  Grubbs test for one outlier
##
## data:  newX[, i]
## G.Hui-Ing (TAI) = 2.08970, U = 0.81047, p-value = 0.3702
## alternative hypothesis: lowest value 10 is an outlier
##
##
## $run200m
##
##  Grubbs test for one outlier
##
## data:  newX[, i]
## G.Joyner-Kersee (USA) = 2.15480, U = 0.79847, p-value = 0.3048
## alternative hypothesis: lowest value 22.56 is an outlier
##
##
## $longjump
##
##  Grubbs test for one outlier
##
## data:  newX[, i]
## G.Launa (PNG) = 2.68320, U = 0.68752, p-value = 0.04594
## alternative hypothesis: lowest value 4.88 is an outlier
##
##
## $javelin
##
##  Grubbs test for one outlier
##
## data:  newX[, i]
## G.Scheider (SWI) = 1.69720, U = 0.87498, p-value = 1
## alternative hypothesis: highest value 47.5 is an outlier
##
##
## $run800m
##
##  Grubbs test for one outlier
##
## data:  newX[, i]
## G.Launa (PNG) = 3.30190, U = 0.52681, p-value = 0.001808
## alternative hypothesis: highest value 163.43 is an outlier
##
##
## $score
##
```

```
##  Grubbs test for one outlier
##
## data:  newX[, i]
## G.Launa (PNG) = 2.68190, U = 0.68781, p-value = 0.04618
## alternative hypothesis: lowest value 4566 is an outlier

# 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): Perform a principal component analysis on the 7 event
# results and save the results of the prcomp function to a new variable
# Hpca.
prcompHeptathlon <- prcomp(heptathlon, scale = TRUE)

Hpca <- as.data.frame(prcompHeptathlon$rotation)

summary(Hpca)

##       PC1                PC2                 PC3
##  Min.   :-0.4319   Min.   :-0.654101   Min.   :-0.87844
##  1st Qu.:-0.4049   1st Qu.:-0.102538   1st Qu.:-0.13854
##  Median :-0.3746   Median :-0.007082   Median : 0.06703
##  Mean   :-0.3462   Mean   :-0.016133   Mean   :-0.04823
##  3rd Qu.:-0.2805   3rd Qu.: 0.086096   3rd Qu.: 0.18843
##  Max.   :-0.2205   Max.   : 0.655357   Max.   : 0.27579
##       PC4                PC5                 PC6
##  Min.   :-0.54739   Min.   :-0.63467   Min.   :-0.3673827
##  1st Qu.:-0.10042   1st Qu.:-0.15879   1st Qu.:-0.1533007
##  Median :-0.02828   Median : 0.06172   Median :-0.0688886
##  Mean   : 0.03278   Mean   : 0.02129   Mean   :-0.0007889
##  3rd Qu.: 0.15256   3rd Qu.: 0.14391   3rd Qu.:-0.0012861
##  Max.   : 0.57564   Max.   : 0.64841   Max.   : 0.8587017
##       PC7                PC8
##  Min.   :-0.73521   Min.   :-0.89036
##  1st Qu.:-0.10560   1st Qu.: 0.10506
##  Median : 0.02157   Median : 0.13808
##  Mean   : 0.01131   Mean   : 0.02992
##  3rd Qu.: 0.26668   3rd Qu.: 0.18677
##  Max.   : 0.47901   Max.   : 0.25178
```

```r
# 6 (b) (v): Use "ggibiplot" to visualize the first two principal
# components. Provide a concise interpretation of the results.
ggbiplot(prcompHeptathlon, circle = T, obs.scale = 1, varname.size = 5,
labels = rownames(heptathlon))
```




```r
# Answer: Hurdles, score, shot and 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")

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


CA Projection 1 vs. Heptathlon
Total Hepthalon Score
5500
-4
-2
0
2
4
PCA 1 Projections