ST464 - Assignment 2 - Solutions

Sean O'Riogain (18145426)

9 March 2019

```
knitr::opts_chunk$set(echo = TRUE)
getwd()
```

```
## [1] "C:/Users/oriogain/Dropbox/Maynooth/ST674 - Machine Learning/Assignments"
```

```
suppressPackageStartupMessages(library(tidyverse))
```

```
## Warning: package 'tibble' was built under R version 3.5.2
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
```

Question 1

For the data matrix below

$$\mathbf{X} = \begin{bmatrix} 4 & 2 \\ 1 & 0 \\ -1 & -1 \\ -3 & 5 \\ 1 & -1 \end{bmatrix}$$

```
x<-c(4,1,-1,-3,1,2,0,-1,5,-1)
X<-matrix(x,nrow=5)
X
```

```
##      [,1] [,2]
## [1,]    4    2
## [2,]    1    0
## [3,]   -1   -1
## [4,]   -3    5
## [5,]    1   -1
```

For a) to c) do all the calculations by hand and check your answers in R.

a. Calculate the sample variance-covariance matrix.

```
S<-cov(X)
S
```

```
##       [,1]  [,2]
## [1,]  6.80 -2.25
## [2,] -2.25  6.50
```

b. Calculate the correlation matrix.

```
R<-cor(X)
R
```

```
##           [,1]      [,2]
## [1,]  1.000000 -0.338432
## [2,] -0.338432  1.000000
```

c. Standardize the variables to have mean 0 and standard deviation 1.

```
X_std<-scale(X)
X_std
```

```
##            [,1]       [,2]
## [1,]  1.3805370  0.3922323
## [2,]  0.2300895 -0.3922323
## [3,] -0.5368755 -0.7844645
## [4,] -1.3038405  1.5689291
## [5,]  0.2300895 -0.7844645
## attr(,"scaled:center")
## [1] 0.4 1.0
## attr(,"scaled:scale")
## [1] 2.607681 2.549510
```

```
round(mean(X_std))
```

```
## [1] 0
```

```
round(sd(X_std))
```

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

d. In R find the eigenvectors of the correlation matrix of x.

```
eigen<-eigen(R)
eigen$vectors
```

```
##            [,1]       [,2]
## [1,] -0.7071068 -0.7071068
## [2,]  0.7071068 -0.7071068
```

e. Using prcomp() function, find the loadings for the principal components of x.

```
p<-prcomp(X)
p$rotation
```

```
##             PC1       PC2
## [1,] -0.7302462 0.6831841
## [2,]  0.6831841 0.7302462
```

Question 2

Body fat data. The data consists of observations taken on a sample of 88 males. In this question you will look at PCA of the variables variables were measured:

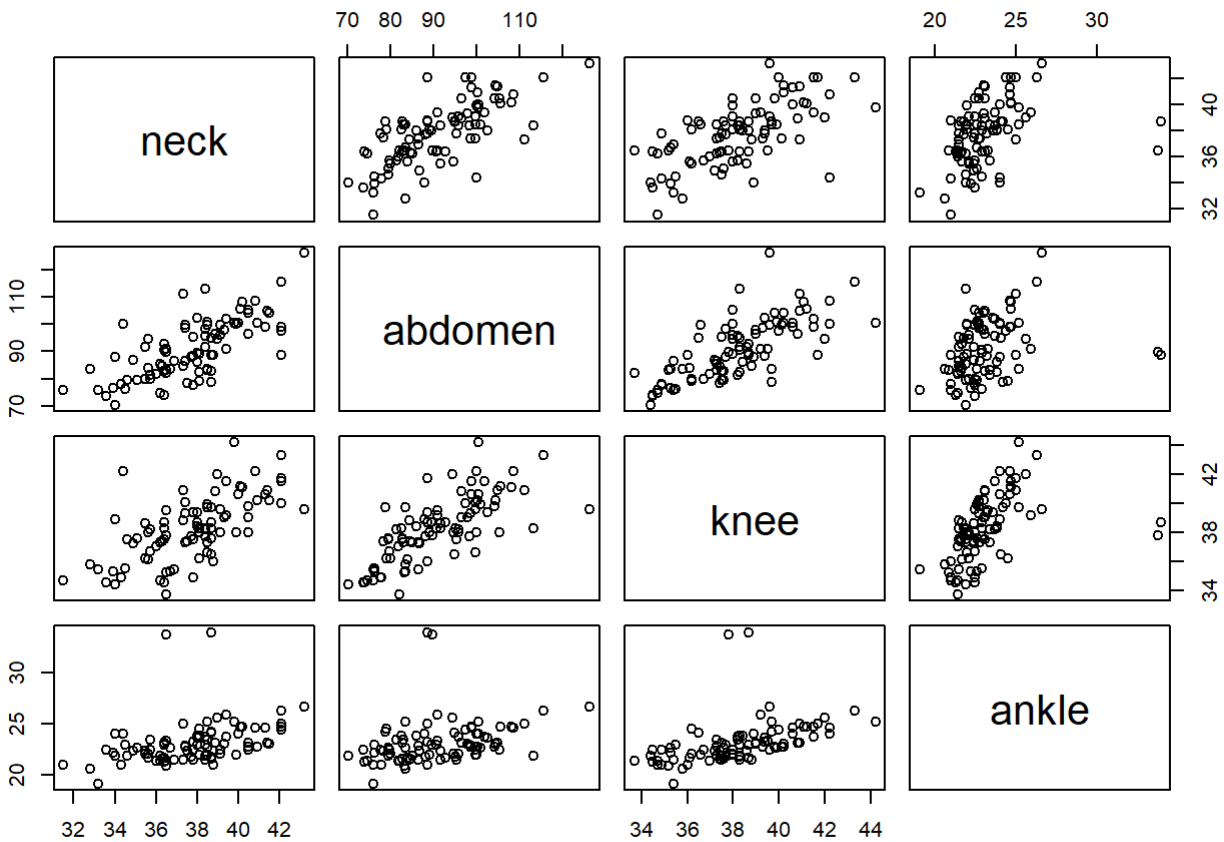
Neck circumference (cm) Abdomen circumference (cm) Knee circumference (cm) Ankle circumference (cm)

```
bfat <- read.table("data/bodyfat.txt", header=T)
bfat <- bfat[,c("neck","abdomen", "knee", "ankle")]
str(bfat)
```

```
## 'data.frame':    88 obs. of  4 variables:
##  $ neck   : num  36.2 38.5 34 37.4 34.4 39 36.4 37.8 38.1 42.1 ...
##  $ abdomen: num  85.2 83 87.9 86.4 100 94.4 90.7 88.5 82.5 88.6 ...
##  $ knee   : num  37.3 37.3 38.9 37.3 42.2 42 38.3 39.4 38.3 41.7 ...
##  $ ankle  : num  21.9 23.4 24 22.8 24 25.6 22.9 23.2 23.8 25 ...
```

a. Use pairs to construct a scatterplot matrix.

```
pairs(bfat)
```



The scatter plot matrix above indicates the existance of linear-type relationships for all of the variable pairings.

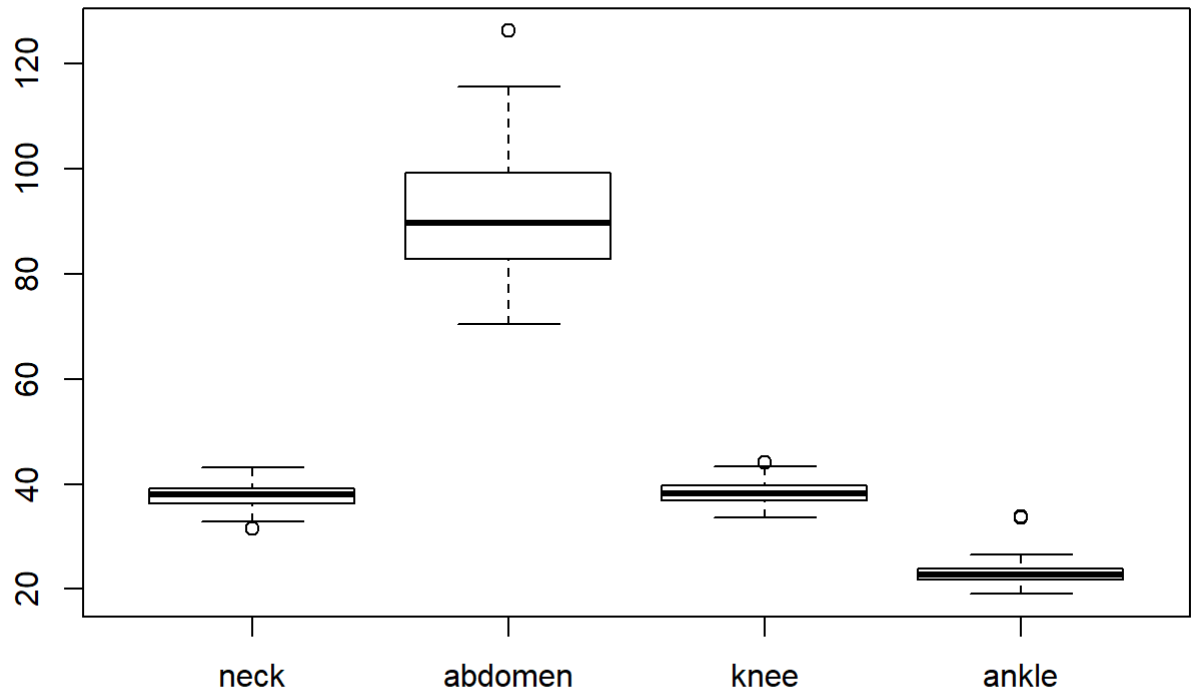
Are there any outliers?

Yes, there are two particularly obvious outliers in the ankle variable.

If so, which cases are they?

Let's do some boxplotting and analyse the output to answer this question accurately.

```
boxplot(bfat)                       # Display a boxplot for the dataset

# Construct an outliers data frame containing variable name, value and observation number.

outliers<-function(data){
  box<-boxplot(data)                # Boxplot the data and store its results

  var<-rep("",length(box$out))      # Name of variable containing the outlier
  val<-box$out                      # Outlier value
  obs<-rep(0,length(box$out))       # Observation (row/case) number

  for(i in c(1:length(box$out))){  # Parse the outlier details provided by boxplot
    var[i]<-box$names[box$group[i]]
    obs[i]<-which(data[,box$group[i]] == box$out[i])
  }

  out<-data.frame(var, val, obs)    # Create the data frame
  out<-arrange(out, var)            # Sort the data frame by variable name
  return(out)                       # Return the contents of the data frame
}

outliers(bfat)                      # Display the outlier details.
```




```
##       var   val obs
## 1 abdomen 126.2  40
## 2   ankle  33.9  31
## 3   ankle  33.7  84
## 4    knee  44.2  34
## 5    neck  31.5  43
```

Note how box plotting has identified 3 more outliers in the other variables too.

b. Carry out a principal components analysis of the data.

```
p<-prcomp(bfat, scale=T)
psum<-summary(p)
psum
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.6283 0.8731 0.58678 0.49183
## Proportion of Variance 0.6629 0.1906 0.08608 0.06047
## Cumulative Proportion  0.6629 0.8535 0.93953 1.00000
```

Note that, even though the measurement units are the same for all variables (cms), I have elected to standardise (scale) them during prcomp because of the relative magnitude difference between the abdominal measurements and the other measurement types.

What percentage of the variability in the dataset is accounted for by the first component?

```
round(psum$importance[3,1]*100,1)
```

```
## [1] 66.3
```

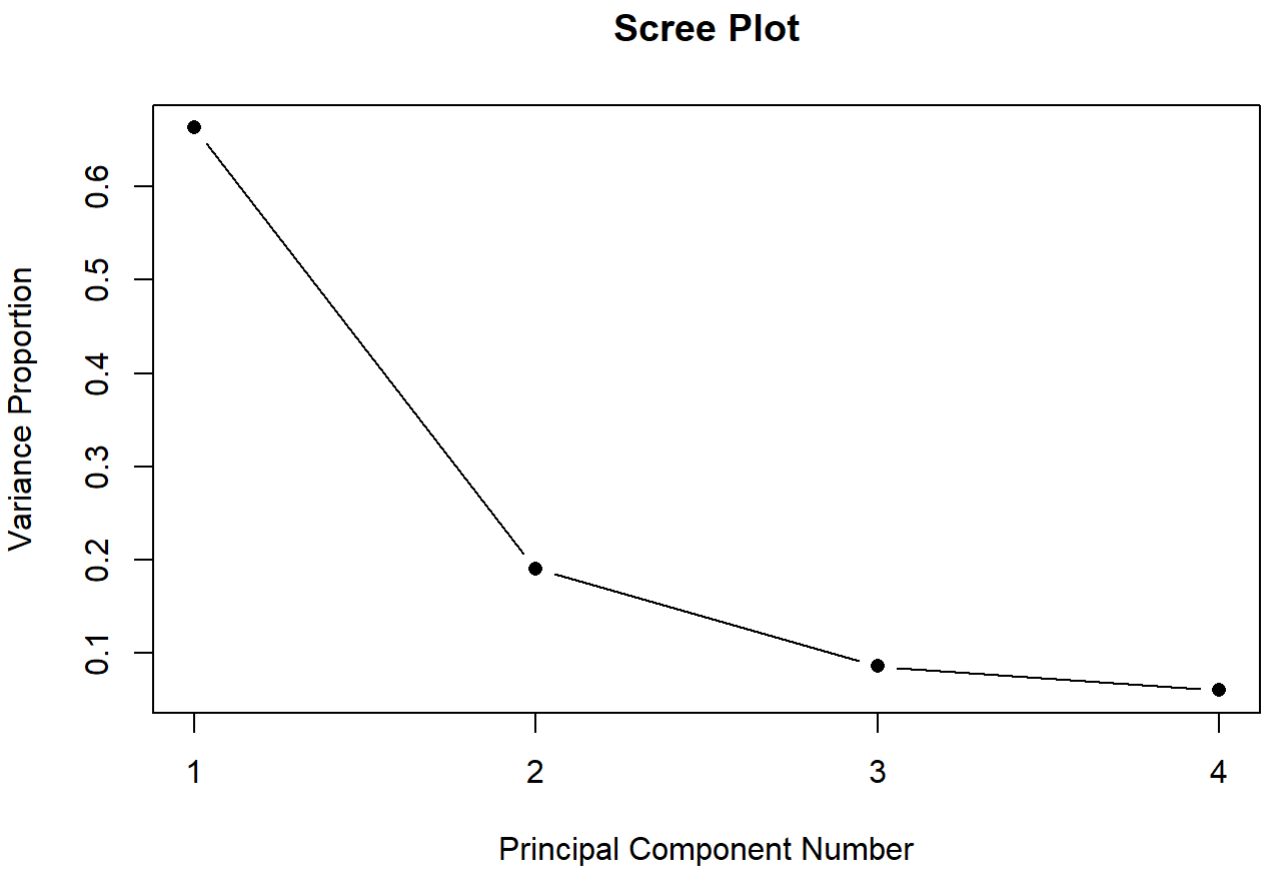
What percentage of the variability in the dataset is accounted for by the first two components?

```
round(psum$importance[3,2]*100,1)
```

```
## [1] 85.3
```

Examine the scree diagram and comment. (You will find the code for the screeplot in h1code.R).

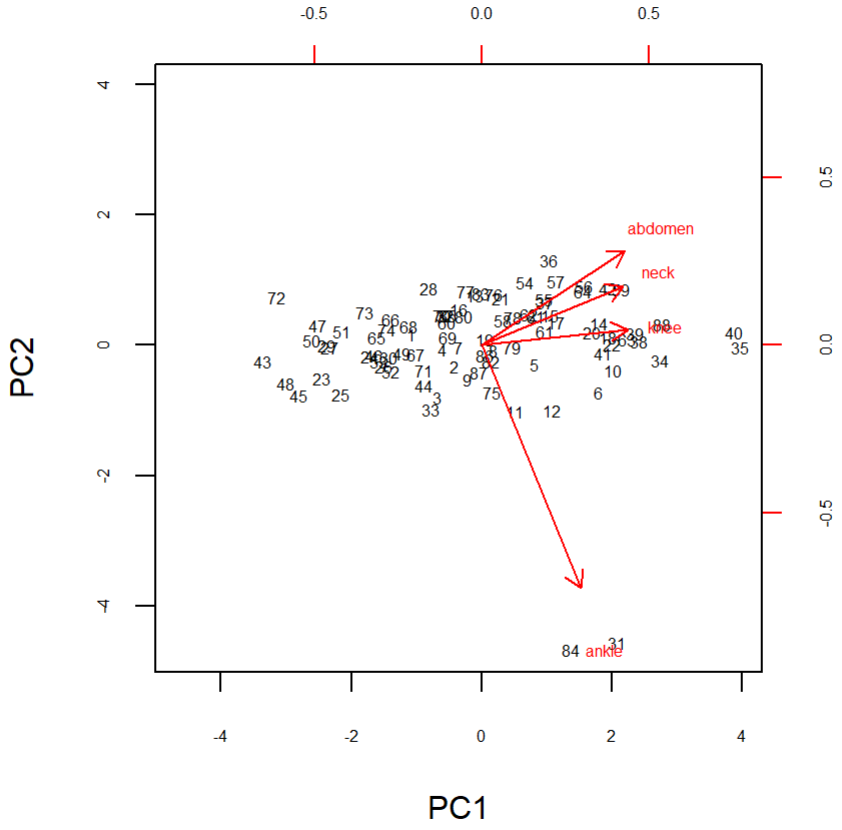
```
scree<-function(p){
  if(class(p)=="princomp"){
    var<-summary(p)$sdev^2
    var_prop<-var/sum(var)
  } else {
    var_prop<-summary(p)$importance[2,]
  }

  len<-length(var_prop)
  plot(var_prop, type="b", pch=16,
       xlim=c(1,len),
       xlab="Principal Component Number",
       ylab="Variance Proportion", main="Scree Plot",
       xaxt="n")
  axis(1,at=c(1:len))
}

scree(p)
```



The scree plot above also clearly shows that over 66% of the variance in the data is explained by PC1, while just over 85% is explained by PC1 and PC2 combined.

Make a biplot to assist your interpretations.

```
biplot(p, scale=0, cex=c(.5,.5), cex.axis=.5)
```



The biplot above indicates, through the proximity of the red vectors, that there is a high degree of colinearity between 3 of the 4 measurement types - i.e. between the abdomen, knee and neck.

c. What does the first component measure?

```
p$rotation[, 1]
```

```
##      neck   abdomen      knee     ankle
## 0.5283837 0.5351101 0.5460990 0.3691120
```

PC1 appears to measure the overall 'size' of the sampled males, giving (almost) equal weigthing to the knee, abdomen and neck measurements - and slighlty less to the ankle measurement - thereby, slightly downplaying the influence of the smallest measurement type (ankle), the one with the lower colinearity.

the second component?

```
p$rotation[, 2]
```

```
##        neck     abdomen        knee       ankle
##  0.21938447  0.35203936  0.05660109 -0.90814925
```

PC2 appears to be focusing on the first three measurment types only, while seeking to downplay/eliminate/ignore the ankle measurement using a negative loading. Of the first three measurement types, the most emphasis appears to be placed on the larger measurment types with particular focus on the abdominal measurement and, to a lesser extent, on the neck measurement. Therefore, PC2 looks like it could be useful as an (surrogate) indicator of weight or BMI (body mass index).

Are there any outliers?

Of the 5 outlier observations previously identified using boxplotting (i.e. 31, 34, 40, 43 and 84), 3 of them (i.e. 31, 40 and 84) are also immediately apparent in the biplot above. Interestingly, the biplot also appears to identify that case 35 as an outlier, which boxplotting did not.

What can you say about the outliers from the plot?

As can be seen from the previous boxplot above, those 3 outliers are the most extreme of the 5 identified there, with the 2 most extreme being ankle measurements and the third being an abdominal measurement. The other 2 of the 5 (for knee and neck, respectively) are shown to be marginal ones by that boxplot. Therefore, the most prominent outliers are still being identified by the biplot of the PC1 & PC2 data. As boxplot implements the formal definition of an outlier (1.5 x IQR), I will discount case 35 on the biplot as a real outlier.

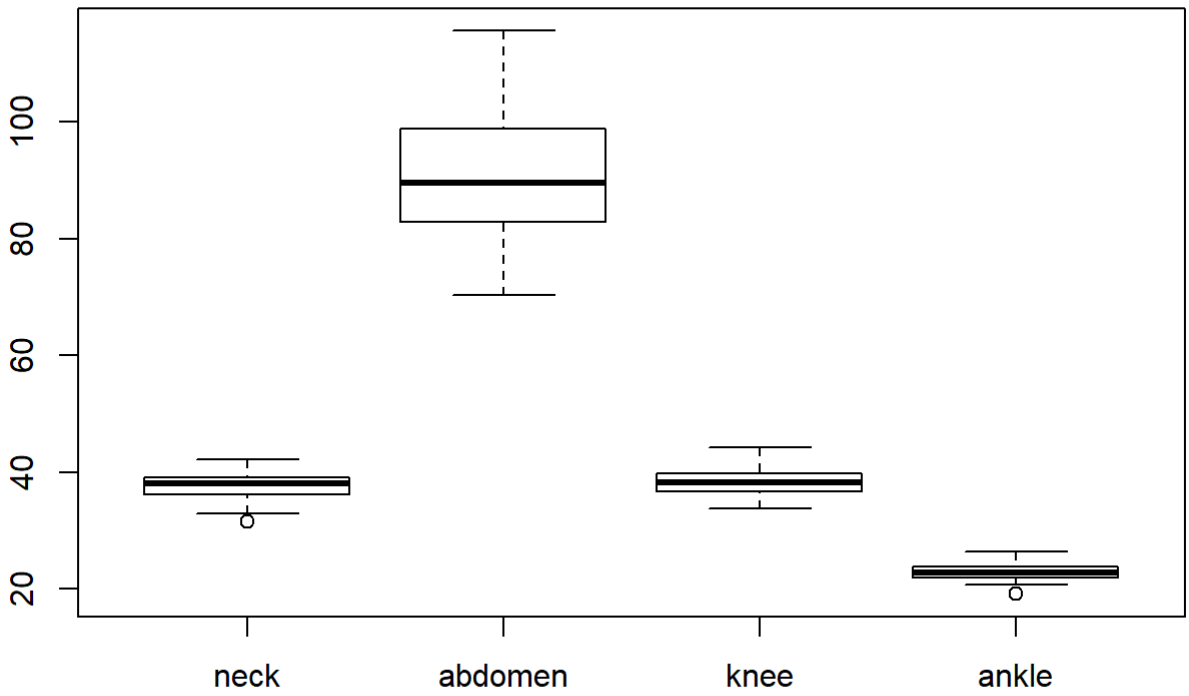
d. Omiting any outliers identified, repeat parts (b) and (c).

Eliminating the 3 observations (i.e. 31, 40 and 84) containing the most prominent outliers…

```
bfat2<-bfat[c(-31,-40,-84),]    # Eliminate the observations with outliers
```

Boxplotting the reduced dataset and analysing its details…

```
boxplot(bfat2)                  # Display a boxplot of the new dataset
outliers(bfat2)                 # Display the remaining (marginal) outlier details.
```




```
##     var  val obs
## 1 ankle 19.1  70
## 2  neck 31.5  41
```

Note that only the 2 marginal outliers remain above.

Carry out a principal components analysis of the data.

```
p2<-prcomp(bfat2, scale=T)
psum2<-summary(p2)
```

What percentage of the variability in the dataset is accounted for by the first component?

```
round(psum2$importance[3,1]*100,1)
```

```
## [1] 73.1
```

Removing the 3 outlier cases has caused this percentage to be increased from 66.3 to 73.1.

What percentage of the variability in the dataset is accounted for by the first two components?

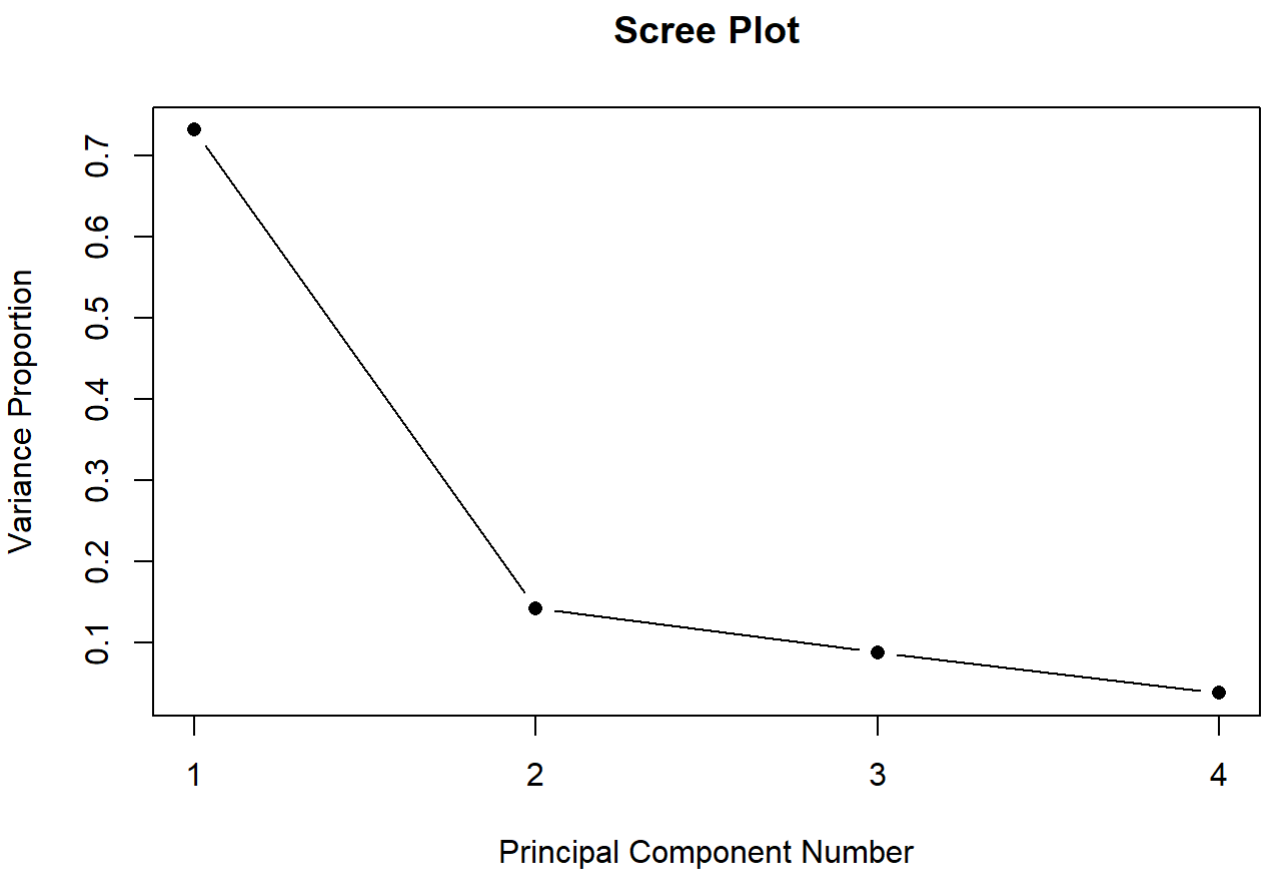
```
round(psum2$importance[3,2]*100,1)
```

```
## [1] 87.3
```

This percentage has increased from 85.3 to 87.3.

Examine the scree diagram and comment.

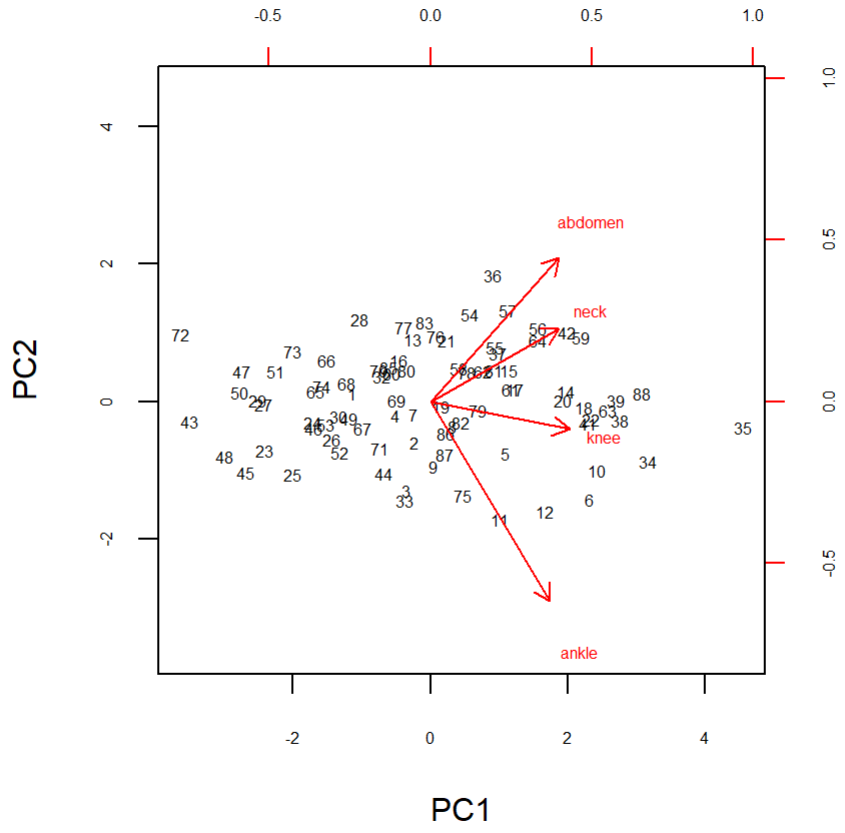
```
scree(p2)
```



Comparing the above scree plot with the previous one shows that PC1 now explains a higher proportion of the overall variance compared to PC2. In fact, by comparing the percentage variances figures underlying those 2 scree plots, we can see that PC1 gained about 7% while the PC1-PC2 combined percentage increased by 2%. This means that PC1 acquired 5% of its additional variance explanation from PC2.

Make a biplot to assist your interpretations.

```
biplot(p2, scale=0, cex=c(.5,.5), cex.axis=.5)
```



The biplot above indicates, through the proximity of the red vectors, that there is still a reasonable level of collinearity between the abdominal and neck measurements, with decreasing levels between those for neck and knee, and between those for knee and ankle.

c. What does the first component measure?

```
p2$rotation[, 1]
```

```
##      neck   abdomen      knee     ankle
## 0.4965605 0.4974463 0.5388220 0.4643766
```

PC1 still appears to measure the overall 'size' of the sampled males, now giving (roughly) equal loadings to neck and abdomen measurements, with those for knee and ankle only slightly above and below them, respectively.

the second component?

```
p2$rotation[, 2]
```

```
##       neck    abdomen       knee      ankle
##  0.2819182  0.5574592 -0.1079343 -0.7733767
```

PC2 now appears to be even more focused on the measurements that would be indication of weight/BMI by now discounting knee measurement as well as ankle measurement using negative loadings.

Are there any outliers?

There are now no obvious outliers, although the biplot still appears to identify that case 35 (and 74, perhaps) as a possible outlier, which boxplotting did not.

What can you say about the outliers from the plot?

As those possible outliers were not identified by boxplotting, they can be discounted.

Question 3

A 1902 study obtained measurements on seven physical characteristics for each of 3000 criminals. The seven variables measured were (1) head length (2) head breadth (3) face breadth (4) left finger length (5) left forearm length (6) left foot length (7) height.

$$\mathbf{R} = \begin{bmatrix} 1.000 & & & & & & \\ 0.402 & 1.000 & & & & & \\ 0.396 & 0.618 & 1.000 & & & & \\ 0.301 & 0.150 & 0.321 & 1.000 & & & \\ 0.305 & 0.135 & 0.289 & 0.846 & 1.000 & & \\ 0.339 & 0.206 & 0.363 & 0.759 & 0.797 & 1.000 & \\ 0.340 & 0.183 & 0.345 & 0.661 & 0.800 & 0.736 & 1.000 \end{bmatrix}$$

read in the correlation data as a vector

```
crimcorr <- matrix(c(
1.000, 0.402, 0.396, 0.301, 0.305, 0.339, 0.340,
0.402, 1.000, 0.618, 0.150, 0.135, 0.206, 0.183,
0.396, 0.618, 1.000, 0.321, 0.289, 0.363, 0.345,
0.301, 0.150, 0.321, 1.000, 0.846, 0.759, 0.661,
0.305, 0.135, 0.289, 0.846, 1.000, 0.797, 0.800,
0.339, 0.206, 0.363, 0.759, 0.797, 1.000, 0.736,
0.340, 0.183, 0.345, 0.661, 0.800, 0.736, 1.000), nrow = 7, byrow = TRUE)
colnames(crimcorr)<- c("Head-L","Head-B","Face-B","L-Fing","L-Fore","L-Foot",
                       "Height")
```

Using the correlation matrix given above,find the principal components of the data

Firstly, we will perform PCA the hard using the eigen vectors (loadings) & values (variations)

```
e<-eigen(crimcorr)              # Get eigen values and vectors (Primary Components)

e_sdev<-sqrt(e$values)          # Calculate the standard deviations for the PCs
names(e_sdev)<-paste("PC", as.character(c(1:nrow(e$vectors))), sep="")
"Standard Deviations:"
```

```
## [1] "Standard Deviations:"
```

```
e_sdev
```

```
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7
## 1.9492241 1.2256950 0.8061063 0.6000474 0.5823766 0.4850290 0.3375164
```

```
# Name the Primary Components and their Loadings
colnames(e$vectors)<-paste("PC", as.character(c(1:nrow(e$vectors))), sep="")
rownames(e$vectors)<-colnames(crimcorr)
"Loadings:"
```

```
## [1] "Loadings:"
```

```
e$vectors
```

```
##                PC1        PC2          PC3         PC4         PC5
## Head-L -0.2763037 -0.3647677  0.882274766 -0.08573946 -0.06740350
## Head-B -0.2118636 -0.6392041 -0.257527788  0.68707351  0.08129399
## Face-B -0.2951449 -0.5123928 -0.381447691 -0.69856220 -0.10071831
## L-Fing -0.4375581  0.2349399 -0.069924234  0.10160027 -0.61923662
## L-Fore -0.4557045  0.2766674 -0.036669136  0.11311530 -0.03907675
## L-Foot -0.4502341  0.1784374 -0.059124621  0.05299938 -0.03440885
## Height -0.4356893  0.1795404 -0.006212105 -0.08162701  0.76976550
##                 PC6         PC7
## Head-L  0.005384671 -0.01638732
## Head-B  0.034955657  0.01762744
## Face-B  0.033740772 -0.07462604
## L-Fing  0.318242311  0.50339046
## L-Fore  0.290305975 -0.78475748
## L-Foot -0.870489463  0.01445146
## Height  0.233030117  0.35269900
```

```
e_varp<-e$values/sum(e$values)  # Calculate the variation proportions
names(e_varp)<-paste("PC", as.character(c(1:nrow(e$vectors))), sep="")
"Variation Percentages:"
```

```
## [1] "Variation Percentages:"
```

```
e_varp
```

```
##        PC1        PC2        PC3        PC4        PC5        PC6
## 0.54278208 0.21461832 0.09282963 0.05143670 0.04845178 0.03360759
##        PC7
## 0.01627391
```

```
e_varc<-cumsum(e_varp)          # Calculate the cumulative variation proportions
names(e_varc)<-paste("PC", as.character(c(1:nrow(e$vectors))), sep="")
"Cumulative Variation Percentages:"
```

```
## [1] "Cumulative Variation Percentages:"
```

```
e_varc
```

```
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7
## 0.5427821 0.7574004 0.8502300 0.9016667 0.9501185 0.9837261 1.0000000
```

Secondly, we will perform PCA the easy way using the princomp function

```
p<-princomp(covmat=crimcorr)
p$sdev
```

```
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7
## 1.9492241 1.2256950 0.8061063 0.6000474 0.5823766 0.4850290 0.3375164
```

```
p$loadings
```

```
##
## Loadings:
##        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## Head-L  0.276  0.365  0.882
## Head-B  0.212  0.639 -0.258 -0.687
## Face-B  0.295  0.512 -0.381  0.699  0.101
## L-Fing  0.438 -0.235        -0.102  0.619  0.318 -0.503
## L-Fore  0.456 -0.277        -0.113         0.290  0.785
## L-Foot  0.450 -0.178                      -0.870
## Height  0.436 -0.180               -0.770  0.233 -0.353
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000
## Proportion Var  0.143  0.143  0.143  0.143  0.143  0.143  0.143
## Cumulative Var  0.143  0.286  0.429  0.571  0.714  0.857  1.000
```

```
psum<-summary(p)
psum
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3    Comp.4     Comp.5
## Standard deviation     1.9492241 1.2256950 0.80610632 0.6000474 0.58237656
## Proportion of Variance 0.5427821 0.2146183 0.09282963 0.0514367 0.04845178
## Cumulative Proportion  0.5427821 0.7574004 0.85023003 0.9016667 0.95011851
##                            Comp.6     Comp.7
## Standard deviation     0.48502898 0.33751644
## Proportion of Variance 0.03360759 0.01627391
## Cumulative Proportion  0.98372609 1.00000000
```

and interpret the results.

From the results above, we can see that PC1 explains about 54% of the variability in the correlation matrix, while PC1 & PC2 explain almost 76% between them, and PC1 , PC2 & PC3 together explain 85% of the variability.

```
p$loadings
```

```
##
## Loadings:
##        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## Head-L  0.276  0.365  0.882
## Head-B  0.212  0.639 -0.258 -0.687
## Face-B  0.295  0.512 -0.381  0.699  0.101
## L-Fing  0.438 -0.235        -0.102  0.619  0.318 -0.503
## L-Fore  0.456 -0.277        -0.113         0.290  0.785
## L-Foot  0.450 -0.178                      -0.870
## Height  0.436 -0.180               -0.770  0.233 -0.353
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000
## Proportion Var  0.143  0.143  0.143  0.143  0.143  0.143  0.143
## Cumulative Var  0.143  0.286  0.429  0.571  0.714  0.857  1.000
```

The loadings for PC1 above gives focuses on the final 4 (non-head) measurements, while PC2 places emphasis on the head measurements and downplays the other measurements (using negative loadings). PC3 completely elimates the non-head measurements (using zero loadings), while placing particular emphasis on the head length measurement (using a relatively large positive loading).

What percentage of the variability in the dataset is accounted for by the first component?

```
round(e_varc[1]*100, 1)
```

```
##  PC1
## 54.3
```

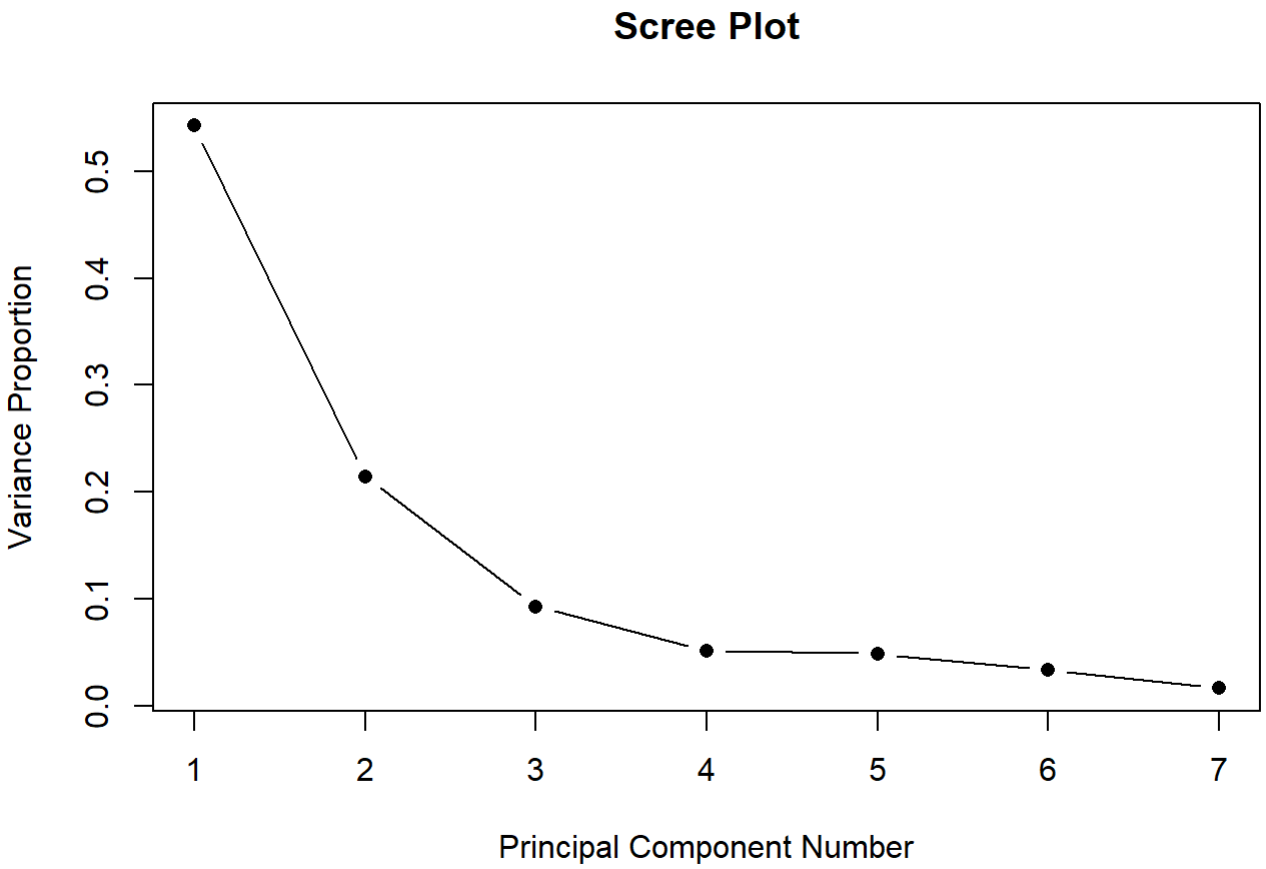
What percentage of the variability in the dataset is accounted for by the first two components?

```
round(e_varc[2]*100, 1)
```

```
##  PC2
## 75.7
```

Examine the scree diagram and comment.

```
scree(p)
```



The scree plot confirms the answers to the previous parts of this question - i.e. that PC1 explains just over 54% of the variance in the dataset's correlation matrix and PC1 and PC2 combined explain almost 76% (54+22) of the variance in it. PC1, PC2 & PC3 combined explain about 85% (54+22+9) of the variance, while adding PC4 brings the total explanation up to 90% (54+2+9+5).

Question 4

For each of the following situations, answer, if possible:

i. Is it a classification or regression problem?

ii. Are we most intererested in inference or prediction?

iii. Provide n and p. For each predictor described state whether it is categorical or quantitative.

iv. Indicate whether we would expect the performance of a flexible learning method to be better or worse than an inflexible method.

a. We have a set of data on 500 worldwide tech firms. For each firm, information on profit, CEO salary, number of employees, average employee salary, and home country is recorded. We are interested in the relationship between CEO salary and other measurements.

   i. This is a regression problem.

   ii. We are more interested in inference.

   iii. n=500; p=4 (which does not include the response variable of the CEO salary). The first 4 features are quantitative while the 5th is categorical.

   iv. An inflexible method is more likely to be more suitable here.

b. A company wishes to launch a new product. They want to know in advance whether it will be a success or failure. They collect data on 20 similar products, and record whether they succeeded or not, price charged, marketing budget, and 10 other variables.

   i. This is a regression problem.

ii. **We are more interested in prediction (of success or failure).**

iii. **n=20; p=12 (which does not include the response variable of the success-failure indicator). The first 2 features are quantative and the other 10 could be of either or both types.**

iv. **A flexible method is more likely to give better results here (provided that care is taken to avoid overfitting).**

c. A dataset was collected to relate the birthweight of babies to the days of gestation and gender.

i. **This is a regression problem.**

ii. **It is not clear as to whether this is an inference or prediction problem. If the ojective of the exercise is to be able to predict the birthweight response based on the other two features (predictors), it is a prediction problem. On the other hand, if the objective is to understand how changes in the values of those predictors influence the value of the response, the problem is one of inference.**

iii. **n is not specified. p=2 (which does not include the birthweight response variable). The first feature is quantative and the second one is categorical.**

iv. **If this is an inference problem, an inflexible (parametric) method would be better. Otherwise, a flexible method could be used instead (or as well). However, with only 2 predictors (of which one of them is categorical and binary) the level of flexibility of the method is probably not of major concern here.**

d. Observations were collected on 56 attributes from 32 lung cancer patients belonging to one of 3 classes.

i. **It is not clear as to whether this is a classification problem or a regression problem. If the classification of the patients is known and recorded it is a predictiregression problem. Otherwise, it is a classification problem.**

ii. **We are more interested in prediction (most likely).**

iii. **n=32, p=56 (which does not include the class response if it is recorded). The types of the 56 features is not given; they could be either all categorical (probably unlikely), all quantative or some combination of both types.**

iv. **A flexible method is more likely to give better results here (provided that care is taken to avoid overfitting).**

Question 5

In this exercise you will conduct an experiment to compare the fits on a linear and exible model fit. You will use the Auto data from the package ISLR and explore the relationship between the response mpg with weight and horsepower.

a.

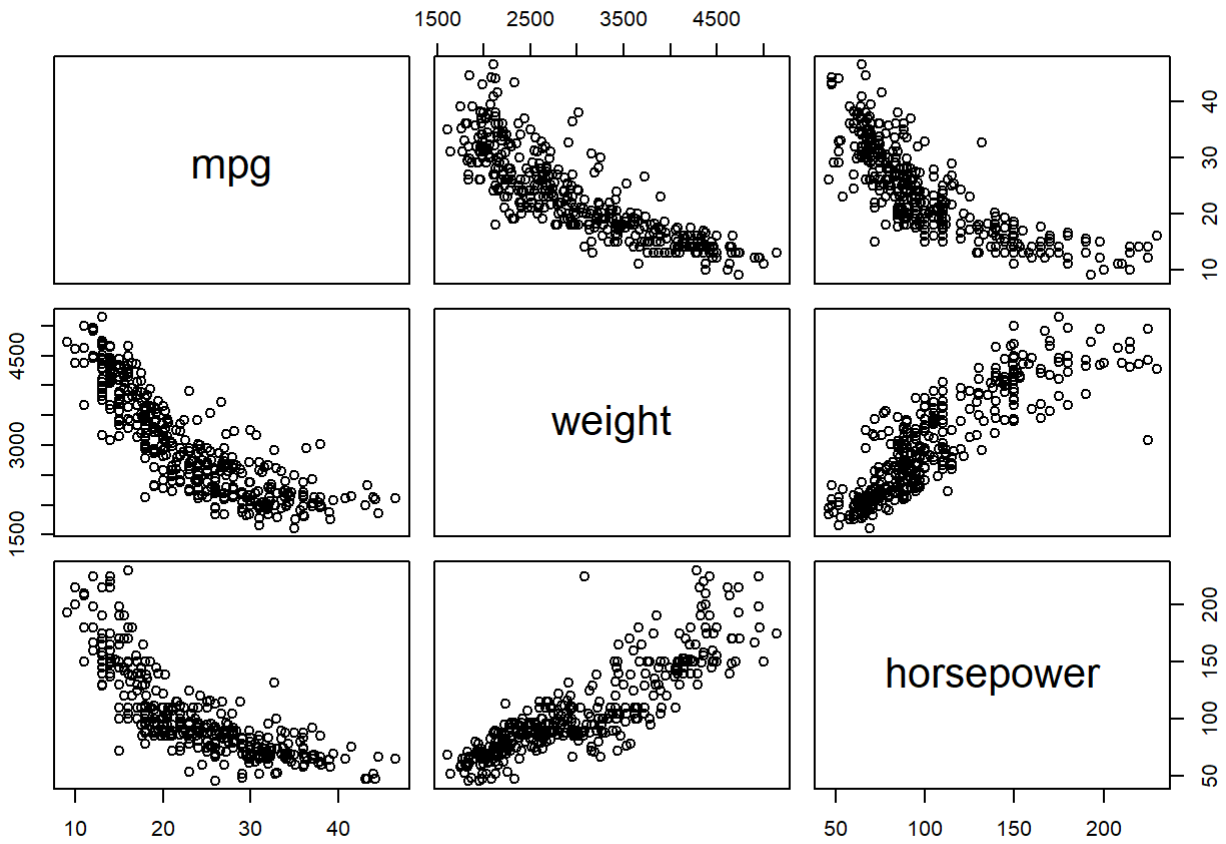
```
# install.packages("ISLR") #home computer, first time only
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.5.2
```

```
Auto <-Auto[complete.cases(Auto[,c(1,4,5)]),] # to remove NAs
```

Plot the response (miles per gallon) vs weight and horsepower.

```
pairs(Auto[c(1,5,4)])
```



What do they tell you about the relationship between mpg and the predictors?

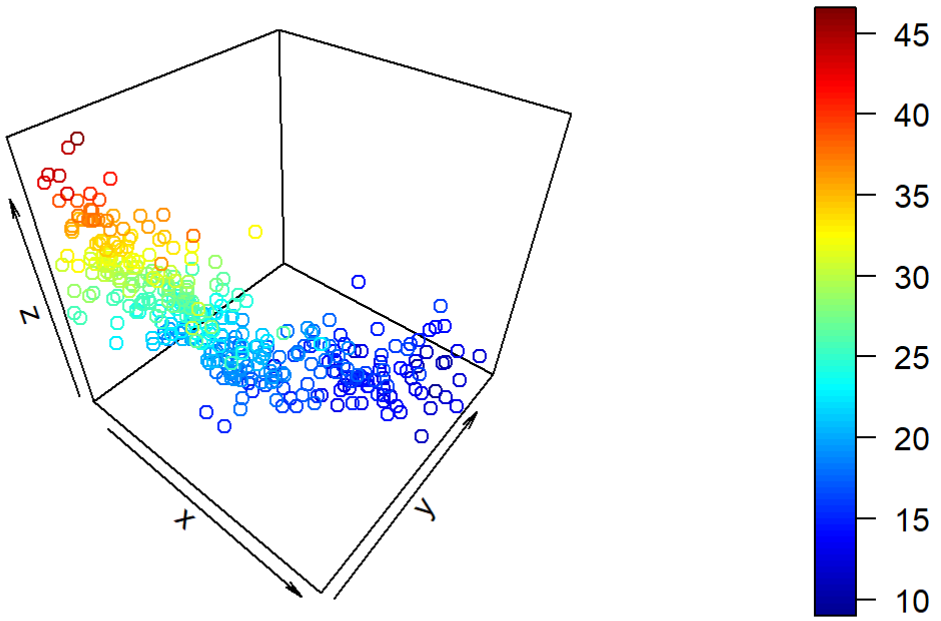
There is (roughly) negative linear (although some curvature is evident at the higher values of mpg) between the mpg response variable and both of those predictors - i.e. as the predictor values increase, the value of the response decreases.

b. Make a 3d plot of weight, horsepower and mpg (see commands below).

```
# install.packages("plot3D") #home computer, first time only nstall package
library(plot3D)
```

```
## Warning: package 'plot3D' was built under R version 3.5.2
```

```
scatter3D(Auto$weight,Auto$horsepower,Auto$mpg)
```




```
library(plot3Drgl)
```

```
## Warning: package 'plot3Drgl' was built under R version 3.5.2
```

```
## Loading required package: rgl
```

```
## Warning: package 'rgl' was built under R version 3.5.2
```

```
scatter3Drgl(Auto$weight,Auto$horsepower,Auto$mpg)
```

What do they tell you about the relationship between mpg and the predictors?

As to be expected, there is a negative relationship between mpg (response) and both weight and horsdepower (predictors) - i.e. mpg decreases when either or both of those 2 predictors increase. As observed previously, those relationships are not perfectly linear as the plots show curvature in the data.

c. Next, divide the data into a training set and a test set as follows:

```
set.seed(123)
train <- sample(nrow(Auto), round(.8*nrow(Auto)))
AutoTrain <- Auto[train,]
AutoTest <- Auto[-train,]
```

Fit a linear regression model to mpg versus weight and horsepower on AutoTrain. Call the fit f1.

```
f1<-lm(mpg ~ weight + horsepower, data=AutoTrain)
summary(f1)
```

```
##
## Call:
## lm(formula = mpg ~ weight + horsepower, data = AutoTrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.4101  -2.7431  -0.4644   2.5079  16.0258
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46.1626326  0.8950964   51.57  < 2e-16 ***
## weight      -0.0060579  0.0005553  -10.91  < 2e-16 ***
## horsepower  -0.0431712  0.0120932   -3.57 0.000414 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.339 on 311 degrees of freedom
## Multiple R-squared:  0.7087, Adjusted R-squared:  0.7068
## F-statistic: 378.3 on 2 and 311 DF,  p-value: < 2.2e-16
```

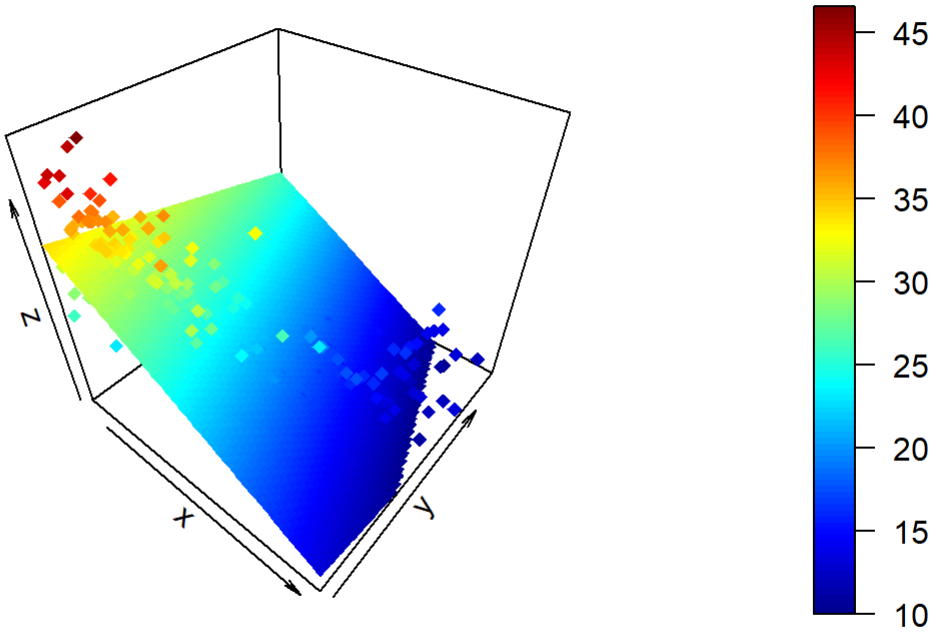
Examine summary(f1) and comment on the significance of the predictors.

In the Estimate column, the Intercept value tells us that the plot hits the Z-plane when the mpg value is 46.16 (its maximum value) and, as noted previously in relation to the 3D plots, the slope of its relationship between both horsepower and weight are both negative.

In the Pr(>|t|) column, all scores are tiny which tells us that all 3 coefficients are highly significant. In particular, for the 2 predictors (horsepower and mpg), this means that we can reject the null Hypothesis (H_0) - i.e. that their slope is zero - and, therefore, that their is a strong indication of linearity in their relationships with the response variable (mpg).

d. Plot the fitted surface and the data. (See lecture notes for code).

```
wt <- seq(min(AutoTrain$weight), max(AutoTrain$weight), length.out = 100)
hp <- seq(min(AutoTrain$horsepower), max(AutoTrain$horsepower), length.out = 100)
pred <- predict(f1, expand.grid(weight=wt, horsepower=hp))
pred <- matrix(pred,100,100)
scatter3D(AutoTrain$weight, AutoTrain$horsepower, AutoTrain$mpg, pch = 18,
          surf = list(x = wt, y = hp, z = pred))
```



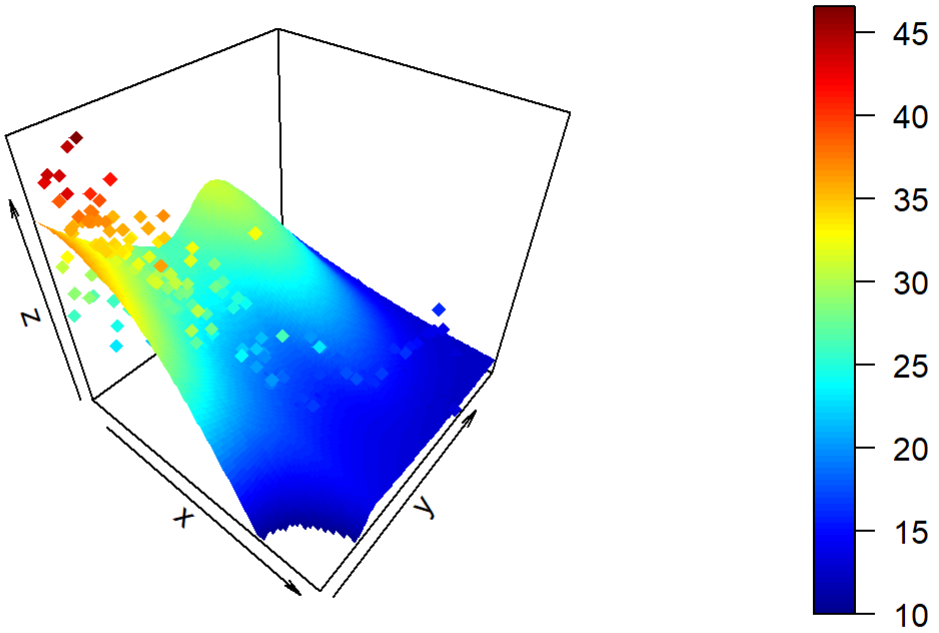

```
scatter3Drgl(AutoTrain$weight, AutoTrain$horsepower, AutoTrain$mpg, pch = 18)
rgl.surface(wt, hp, pred,  coords=c(1,3,2), alpha.col=0.2)
```

Does the linear surface look like a good fit?

It is not easy to judge the goodness of fit from the static scatter3D plot above but it doesn't look particularly good. However, the rotatable scatter3Drgl plot clearly shows that the fit is poor.

e. Use loess to fit a surface to the same data. Call the fit f2. Plot the fitted surface and the data.

```
f2 <- loess(mpg ~ weight + horsepower, data=AutoTrain)
wt <- seq(min(AutoTrain$weight), max(AutoTrain$weight), length.out = 100)
hp <- seq(min(AutoTrain$horsepower), max(AutoTrain$horsepower), length.out = 100)
pred <- predict(f2, expand.grid(weight=wt, horsepower=hp))
pred <- matrix(pred,100,100)
scatter3D(AutoTrain$weight, AutoTrain$horsepower, AutoTrain$mpg, pch = 18,
          surf = list(x = wt, y = hp, z = pred))
```




```
scatter3Drgl(AutoTrain$weight, AutoTrain$horsepower, AutoTrain$mpg, pch = 18)
rgl.surface(wt, hp, pred,  coords=c(1,3,2), alpha.col=0.2)
```

Does the loess surface look like a good fit?

Again, it is not easy to judge the goodness of fit from the static scatter3D plot above but it does appear to be better than that of the linear fit used previously. However, the rotatable scatter3Drgl plot clearly shows that fit is much better than the previous than the linear one.

f. Calculate the MSE for both fits on the training data. (See lecture notes for code.)

```
mean((f1$residuals)^2)
```

```
## [1] 18.64557
```

```
mse_f1<-mean((f1$fitted.values - AutoTrain$mpg)^2)
mse_f1
```

```
## [1] 18.64557
```

```
mean((f2$residuals)^2)
```

```
## [1] 16.12921
```

```
mse_f2<-mean((f2$fitted - AutoTrain$mpg)^2)
mse_f2
```

```
## [1] 16.12921
```

What do these tell you?

The loess fit is confirmed as a better one because of the lower value of its MSE compared to the linear equivalent.

g. Calculate the MSE for both fits on the test data.

```
AutoTestHat1<-predict(f1, AutoTest)

mse_f1<-mean((AutoTestHat1 - AutoTest$mpg)^2)
mse_f1
```

```
## [1] 14.81678
```

```
AutoTestHat2<-predict(f2, AutoTest)

mse_f2<-mean((AutoTestHat2 - AutoTest$mpg)^2)
mse_f2
```

```
## [1] 14.70665
```

What do these numbers tell you?

This time, the linear fit achieved a lower MSE value than the loess equivalent. This may be an indication that the loess fit is overfitted. This calls into question the previously declared preference for the loess fit and it means that further evaluation of those 2 fits is required using additonal test datasets, assuming that they can be acquired.

Interestingly, the MSE values achieved for the test dataset are lower than those observed for the training set when higher values would have been expected - which is a further indication of possible overfitting in the case of the loess fit.