Multivariate Statistics - Exercise 3

Philipp Satlawa - h0640348

11/05/2021

This document contains the answered questions of exercise 3 of the course "Multivariate Statistics".

Principal Component Analysis (PCA)

1. install and load the packages - load and summerize the data

```
# import necessary libraries
library(grid)
library(lattice)
library(modeltools)
```

```
## Loading required package: stats4
```

```
library(stats4)
library(flexclust)

# load data
data(milk)

# explore data
help(milk)
str(milk)
```

```
## 'data.frame':    25 obs. of  5 variables:
##  $ water  : num  90.1 88.5 88.4 90.3 90.4 87.7 86.9 82.1 81.9 81.6 ...
##  $ protein: num  2.6 1.4 2.2 1.7 0.6 3.5 4.8 5.9 7.4 10.1 ...
##  $ fat    : num  1 3.5 2.7 1.4 4.5 3.4 1.7 7.9 7.2 6.3 ...
##  $ lactose: num  6.9 6 6.4 6.2 4.4 4.8 5.7 4.7 2.7 4.4 ...
##  $ ash    : num  0.35 0.24 0.18 0.4 0.1 0.71 0.9 0.78 0.85 0.75 ...
```

```
summary(milk)
```

```
##      water          protein           fat            lactose
##  Min.   :44.90   Min.   : 0.600   Min.   : 1.00   Min.   :0.000
##  1st Qu.:71.30   1st Qu.: 3.000   1st Qu.: 3.40   1st Qu.:2.700
```

```
##  Median :82.00   Median : 5.900   Median : 6.30   Median :4.700
##  Mean   :78.18   Mean   : 6.212   Mean   :10.31   Mean   :4.132
##  3rd Qu.:87.70   3rd Qu.: 9.700   3rd Qu.:13.10   3rd Qu.:5.600
##  Max.   :90.40   Max.   :12.300   Max.   :42.00   Max.   :6.900
##       ash
##  Min.   :0.1000
##  1st Qu.:0.5300
##  Median :0.8000
##  Mean   :0.8632
##  3rd Qu.:1.1000
##  Max.   :2.3000
```

The dataset `milk` contains 25 records and 5 attributes. The records represent tree species and all attributes are stored as continuous numbers.

2. means and standard deviations

```r
# calulate means
apply(milk, 2, mean)
```

```
##   water protein     fat lactose     ash
## 78.1840  6.2120 10.3080  4.1320  0.8632
```

```r
# calulate standard deviations
apply(milk, 2, sd)
```

```
##      water    protein        fat    lactose        ash
## 12.8179132  3.6525471 10.5179973  1.8318297  0.5048244
```

The variables are directly comparable because they are expressed as percentages (same unit of measurement), hence we do not need to scale the data. Even though the data does not need any scaling, the data matrix has to be centered for computing the PCA.

3. PCA

```r
# PCA with milk data
pca_milk <- prcomp(x = milk, center = TRUE, scale. = FALSE, retx = TRUE)

# loadings matrix
pca_milk$rotation
```

```
##                 PC1        PC2         PC3         PC4          PC5
## water   -0.76163901  0.1560720 -0.57443881 -0.25593731 -0.007980218
## protein  0.16060436 -0.8536804 -0.26796098 -0.39275281 -0.139205643
## fat      0.62047078  0.4429998 -0.55244430 -0.33676547  0.012709314
## lactose -0.09474321  0.1822437  0.53829678 -0.81635500  0.040150186
## ash      0.01232867 -0.1319461 -0.05708616 -0.01987089  0.989335404
```

As we have 5 variables, PCA returns 5 principal components (PC). Examining the loadings matrix we can see that a particular PC describes the most variation for the variables with high absolute values. Therefore, the variables that influence most PC1 are `water` and `fat`. For PC2 this variables are especially `protein`, but also `fat`. In PC3 the most influencing variables are a mixture of `water`, `fat` and `lactose`. PC4 is shaped by `lactose` and PC5 is almost exclusively influenced by `ash`.

4. scores matrix

```
# scores matrix
pca_milk$x
```

```
##                   PC1         PC2         PC3          PC4          PC5
## HORSE      -15.699712  1.39197129  0.78431583 -0.745985682 -0.107171004
## ORANGUTAN  -13.038725  3.12466664  0.16567273  0.029808998 -0.040544661
## MONKEY     -13.369091  2.15252958  0.66944715 -0.314736887 -0.204578654
## DONKEY     -15.681458  2.23453003  0.30995318 -0.007946847  0.042963397
## HIPPO      -13.843989  4.27403009 -2.11711932  0.829914816 -0.134380501
## CAMEL      -12.034705  0.88207313 -0.55503648  0.413741173  0.089044149
## BISON      -12.354335 -0.96671901  0.96894134 -0.058081301  0.116964048
## BUFFALO     -4.581620 -0.07472475 -0.52511055 -0.530816742 -0.077929801
## GUINEA PIG  -4.432366 -2.07028321 -1.50604285  0.798296377 -0.305085635
## CAT         -4.490963 -4.49773268 -0.63919284 -1.268082504 -0.720663411
## FOX         -5.346419 -1.61967975  0.77852120 -0.170495743 -0.040371922
## LLAMA      -11.255276  0.39863441  0.56309423 -0.023754469  0.161556626
## MULE       -15.089414  1.97199739 -0.19988497  0.386359851  0.049827843
## PIG         -6.560672 -2.45475661 -0.25849121  0.571663412 -0.009711660
## ZEBRA      -10.151385  1.78744627 -0.06310054  0.114573084  0.198592076
## SHEEP       -5.482741 -0.51588050  0.43396688  0.115167826  0.074179545
## DOG          1.540935 -3.53888784  0.07257563  0.458893058 -0.137343266
## ELEPHANT     9.663123  4.59042799  1.77412249 -0.708163934  0.344232864
## RABBIT       8.182417 -5.63109875 -0.50283397  0.224097979  0.574798165
## RAT          6.316605 -2.64501034  0.71973369  0.177876460  0.156213136
## DEER        16.007811 -1.68177440 -0.10968560 -0.423827042  0.103966802
## REINDEER    17.275552 -1.86198226  0.05651254 -0.380545631  0.074593918
## WHALE       17.987185 -2.00835779 -1.04946468 -0.111977448  0.291015498
## SEAL        44.823398  5.35002888 -2.40823821 -0.534540641 -0.008082254
## DOLPHIN     41.615844  1.40855222  2.63734433  1.158561837 -0.492085297
```

The PCA basically maps the data points to a different coordinate system (where the variance is maximized). The scores in this matrix show where the data points lie on the new axes. If we take for example the data point `HORSE` with the value *-15.7* in the column `PC1`, it means that this data point is mapped onto the value *-15.7* on the new principal component axis `PC1`.

5. plot summary of PCA

```
# summary method
summary(pca_milk)
```

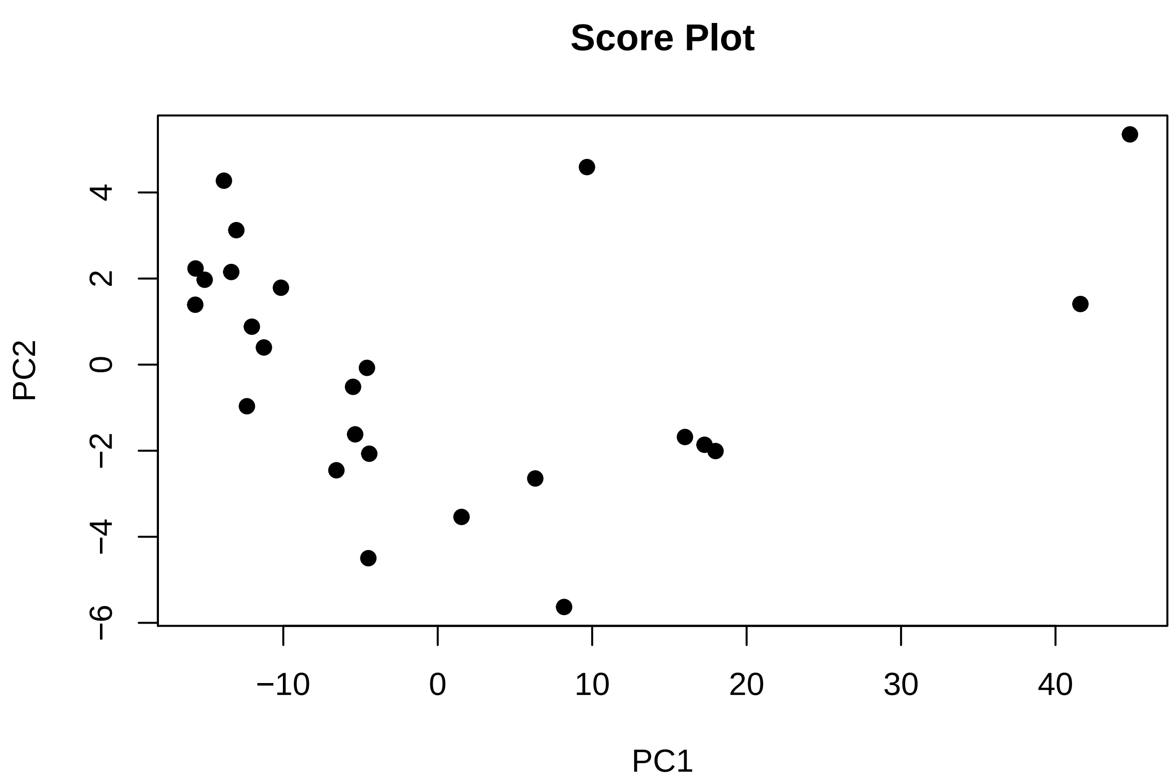
```
## Importance of components:
```

```
##                           PC1     PC2     PC3     PC4     PC5
## Standard deviation     16.7978 2.85207 1.09708 0.55294 0.26026
## Proportion of Variance  0.9667 0.02787 0.00412 0.00105 0.00023
## Cumulative Proportion   0.9667 0.99460 0.99872 0.99977 1.00000
```

The first PC (`PC1`) explains 96.7% of the variance of the data. If we look at the second PC (`PC2`), we can see that `PC2` just explains 2.8% of the variance and the consecutive PCs (`PC3` to `PC5`) explain just a minimal fraction of the variance. Due to the high proportion of variance of `PC1`, it is sufficient to choose just this PC for further computation.

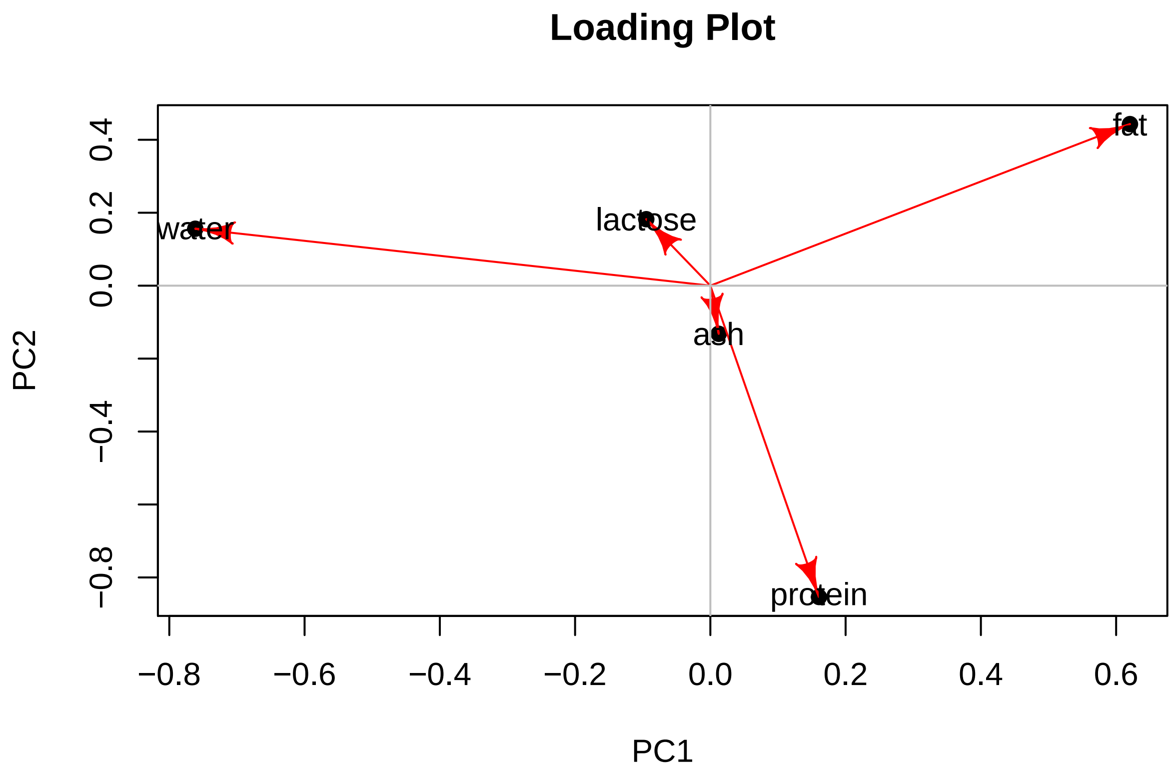
6. score plot and a loadings plot

```r
library(shape)
# score plot
plot(pca_milk$x[, 1:2], pch = 19,
xlab = "PC1", ylab = "PC2", main = "Score Plot")
```



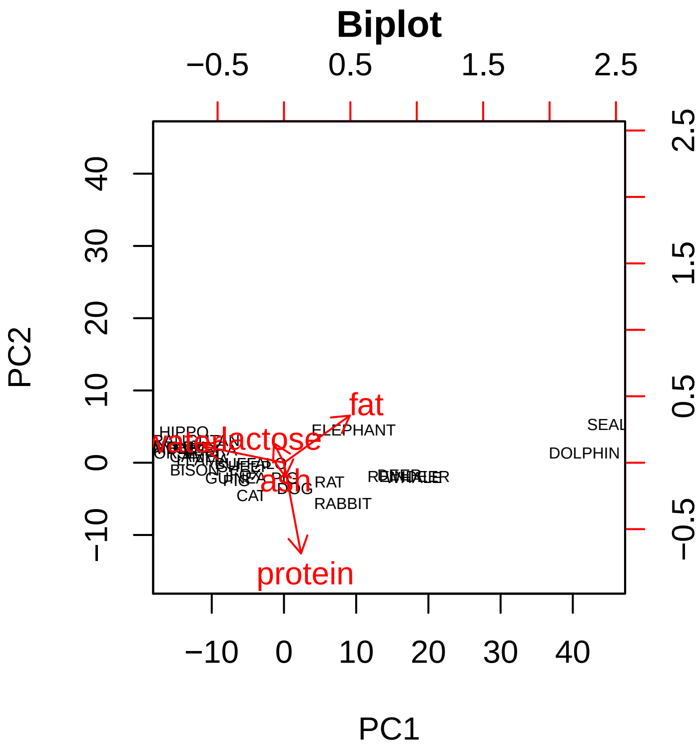

```r
# loading plot
plot(pca_milk$rotation[, 1:2], pch = 19,
xlab = "PC1", ylab = "PC2", main = "Loading Plot")
# Arrows (shape package required)
Arrows(x0 = 0, y0 = 0, x1 = pca_milk$rotation[, 1],
y1 = pca_milk$rotation[, 2], arr.adj = 1, col = "red")
```

```r
# show zero lines
abline(h = 0, col = "grey")
abline(v = 0, col = "grey")
# variable names
text(pca_milk$rotation[, 1:2], labels = rownames(pca_milk$rotation))
```



7. biplot

```r
# biplot
biplot(pca_milk, main = "Biplot", cex = c(0.5,1), scale = 0)
```

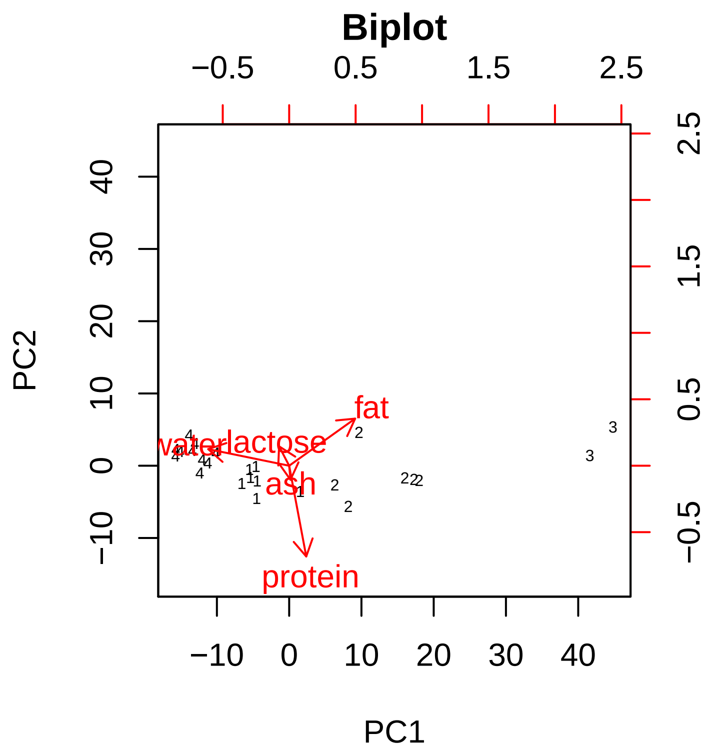


The biplot is the combination of the score plot and the loadings plot. It shows on the one hand the variable loadings (red vectors) in the space of `PC1` and `PC2` and on the other hand the data records and maps them into the space of `PC1` and `PC2`. For example we can see that the vector `fat` points in the upper right corner, therefore the most right data points (`SEAL` and `DOLPHIN`) are also the ones with the highest fat content. Similarly we can interpret that the data point `RABBIT` should have the highest protein content because it is the lowest data point and the `protein` vector is pointing down.

8. cluster data

```r
# calculate k-means
km <- kmeans(as.matrix(milk), centers=4)

# biplot
biplot(pca_milk, main = "Biplot", cex = c(0.5,1), scale = 0, xlabs=km$cluster)
```



PCA for Dimensionality Reduction

9. load photo

```
# load package
require(jpeg)
```

```
## Loading required package: jpeg
```

```
# load photo
photo <- readJPEG("Foxe_Basin_Canada.jpg")

nrow(photo)
```

```
## [1] 864
```

```
ncol(photo)
```

```
## [1] 1024
```

```r
# dimensions / resolution
dim(photo)
```

```
## [1]  864 1024    3
```

```r
# class of object photo
class(photo)
```

```
## [1] "array"
```

```r
# create matrices for every channel
r <- photo[,,1]
g <- photo[,,2]
b <- photo[,,3]

# preform PCA on data
r.pca <- prcomp(r, center = F)
g.pca <- prcomp(g, center = F)
b.pca <- prcomp(b, center = F)
rgb.pca <- list(r.pca, g.pca, b.pca)
```

interpret results

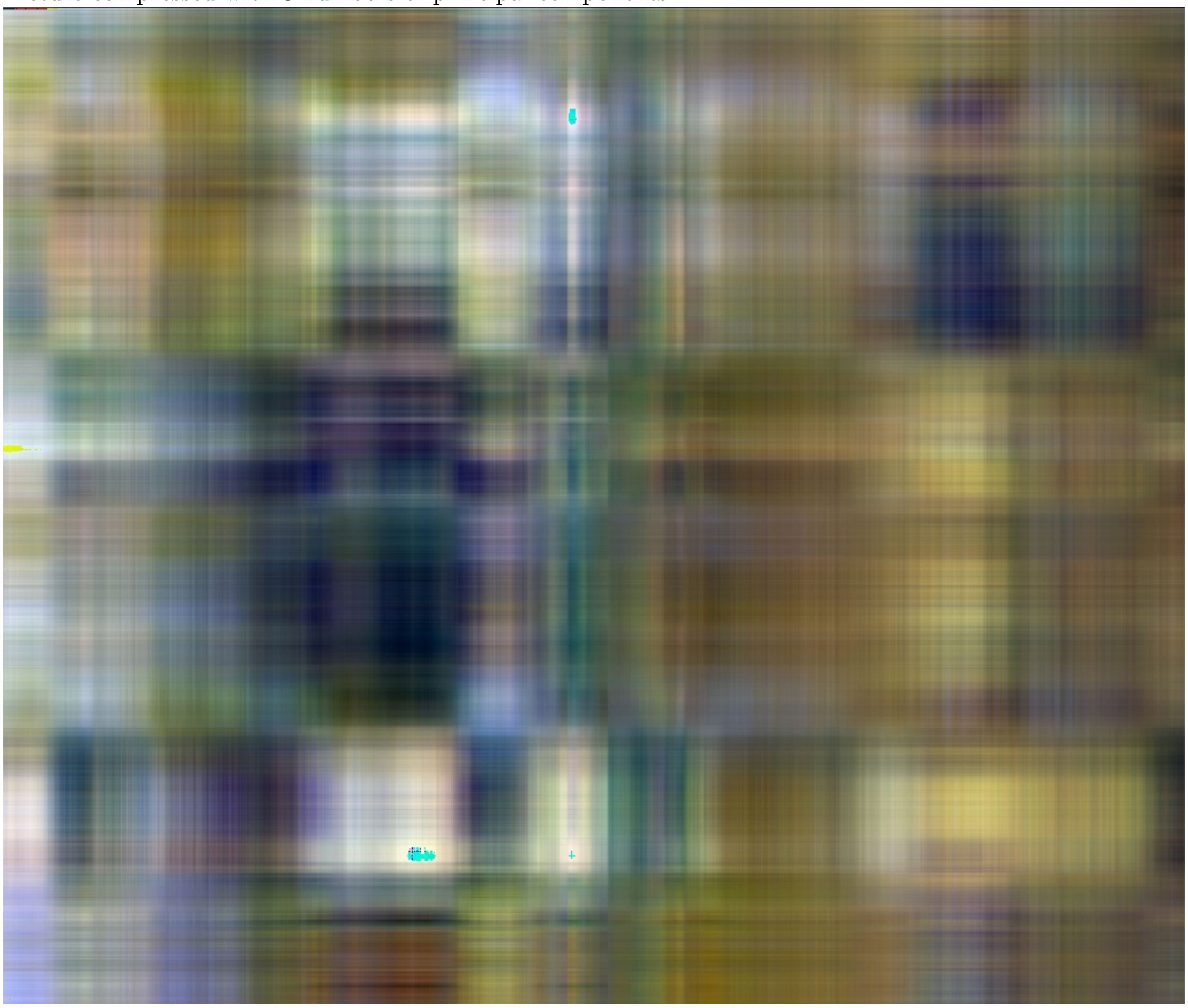
10. apply PCA compression on photo

```r
for (i in seq.int(3, round(nrow(photo)/5), length.out = 20)) {
pca.img <- sapply(rgb.pca, function(j) {
compressed.img <- j$x[,1:i] %*% t(j$rotation[,1:i])
}, simplify = 'array')
writeJPEG(pca.img, paste('PIC_', round(i,0), '_components.jpg', sep = ''),
quality=1)
}
```

As expected the lower the PCs used the lower the quality of the image. With 3 PCs the image is not recognizable, however with just 12 PCs shapes are already recognizable although the image quality is very low. With 57 PCs the quality improved drastically however there is still a quality deficit. When using more than 100 PCs the quality difference is much harder recognizable and can be considerd satisfactory.

11. include photo

Picture compressed with 3 numbers of principal components



Picture compressed with 101 numbers of principal components