Lab 6: Multivariate statistics – PCA

Sometimes, a dataset contains numerous, possibly correlated, response variables and it can be difficult to discern any patterns in the data using the normal univariate analysis tools we have seen so far, and graphical methods that only display data in 2 or 3 dimensions. To deal with data like this we can use multivariate methods where the goal is to reduce data in high dimensional space into something more interpretable. Hopefully, we can start to get a feeling for how variables relate to each other and how they separate groups from each other. We can also use multivariate techniques to reduce dimensionality for hypothesis testing. Multivariate statistics can be very abstract and relies heavily on linear algebra, which we will not go into in this class. Here we will focus on the computation and interpretation of multivariate data analysis using R.

Principal component analysis (PCA)

One of the most commonly used multivariate technique is PCA. In bioinformatics it is used to analyze and visualize the relationships between thousands of genes, transcripts, proteins, metabolites and so on. It is commonly used to detect differences between groups of individuals such as populations or different treatments without defining those groups *a priori*. In essence PCA aims to describe the variation in a large number of correlated variables by creating latent uncorrelated variables called principal components (PCs).

<u>Definition from DataCamp:</u>

"PCA is a type of linear transformation on a given data set that has values for a certain number of variables (coordinates) for a certain amount of spaces. This linear transformation fits this dataset to a new coordinate system in such a way that the most significant variance is found on the first coordinate, and each subsequent coordinate is orthogonal to the last and has a lesser variance. In this way, you transform a set of x correlated variables over y samples to a set of p uncorrelated principal components over the same samples.
Where many variables correlate with one another, they will all contribute strongly to the same principal component.

Each principal component sums up a certain percentage of the total variation in the dataset. Where your initial variables are strongly correlated with one another, you will be able to approximate most of the complexity in your dataset with just a few principal components."

If you felt the lecture did not give you enough on PCA for you to understand the very basics of it, here is a fun/helpful thread about the basics of PCA (why do it, how it's done):

https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues/140579#140579

Remember though, it is not about knowing everything in detail, or being able to do the math. Focus on understanding the principle, and today, on understanding and interpreting the output of the analyses.

6.1 PCA to reduce the number of variables

1. **American Crime**

We will start this lab by using PCA for hypothesis testing by revisiting the last question of lab 5. Here we will run a regression of crime explained by education based on principal components:
(PC.Crime ~ Police.funding + PC.Education)

First, we load the data set:

```
dat <- read.table("AmericanCrime.txt", header=TRUE)
```

During lab 5 we log-transformed violent crime before regression analysis because of an outlier. Remember that outliers can be equally problematic in PCA (they will tend to dominate the construction of PCs). Hence, it might be a good idea to carefully consider the distribution of your variables before applying PCA.

```
plot(dat)
```

I here choose to log-transform (using base-10) all variables. (You may make a different choice).

```
dat2 <- log10(dat)
```

We will run a PCA using the native prcomp function. To learn more about it and its output, type **?prcomp** into your console.

```
PC.Crime <- prcomp(cbind(dat2$violent.crime, dat2$crime.rate), scale=TRUE)
```

Note that by using **"scale=TRUE",** we are weighting the two variables equally by analysing the correlation matrix and not their covariance matrix. We have now saved the output of the PCA in the object called "PC.Crime".

For the two education variables:

```
PC.Education <- prcomp(dat2[, 4:5], scale=TRUE)
```

Note how I specified the selected variables by names in the first PCA, and by position in the dat2 data.frame in the second PCA. Both approaches work.

Now you have PCs for both crime and education. Now let's have a look at the output. For this first exercise we only need to focus on two things. We will cover more in the next exercise.

```
PC.Crime
summary(PC.Crime)

PC.Education
summary(PC.Education)
```

6.1a. In the output we see how much variance is explained by each principal component (given by their eigen-values), and how they are related to the original variables (given by their eigen-vectors/loadings***). How much variation does PC1 and PC2 explain for the crime and education PCA, respectively? How are the new PCs related to the original variables in each analysis?***

We see that for both crime and education, the first PC explains a great deal of the variation which is why we are going to focus on just using PC1 in downstream analyses. What we are after for our regression are the scores for each individual on respective PC1 (i.e., we are ordinating individuals along PC1), which are our new variables.

We can extract the scores for the first PC, and save them as variables inside the dat2 dataset, by typing:

```
dat2$PC.Crime.scores <- PC.Crime$x[,1]

dat2$PC.Education.scores <- PC.Education$x[,1]
```

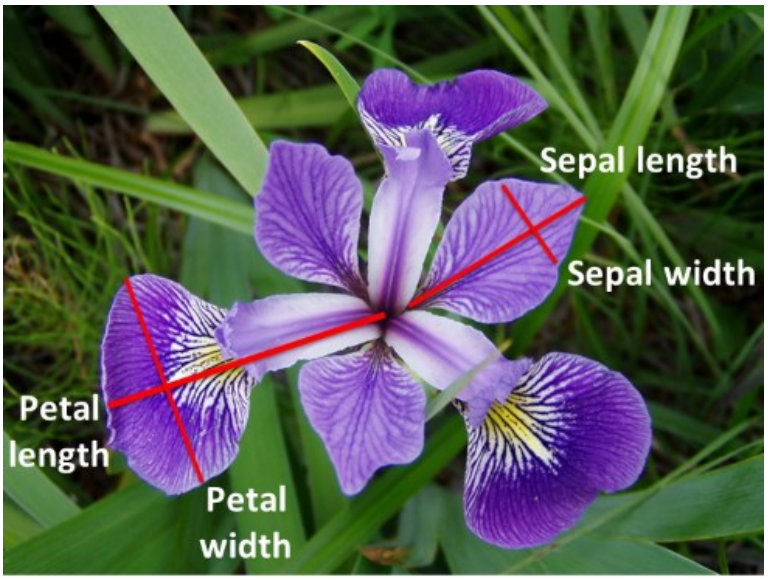
Take a look at the scores by plotting - you can try a histogram, scatterplot, or something else if you prefer that. But make sure to get an idea of how the new variables look. Now we are ready to run our regression again:

```
mod <- lm(PC.Crime.scores ~ police.funding + PC.Education.scores, dat2)
summary(mod)
```

6.1b. How do the result compare to the last lab? Are they different? How does police funding and education relate to crime rate based on model coefficients?

6.2 Ordination with PCA on the Iris data

We will continue learning more about PCA using the classic *Iris* data, which is already present in R. In this dataset, measurements of 4 continuous floral traits have been measured for 3 Iris species to describe the morphology of flowers.



Load the dataset:
```
data("iris")
```
To find out more about it
```
?iris
```
Inspect the data
```
str(iris)
iris
```

Since most multivariate techniques assume that the variables follow a multivariate normal distribution It is advisable to inspect the variables before doing PCA and if necessary, transform them using an appropriate method. Now let's have a look at the distribution of the variables for each species at a time.

First, we subset each species under a new variable name:
```
iris.setosa <- iris[iris$Species=="setosa",1:4]
```

Above we are picking all rows where the variable "Species" is equal to "setosa", and for those rows we choose columns 1 to 4 and save them into a new dataset.

Then we specify that we want to plot a histogram for each of the four measured traits in one single plot.

```
par(mfrow=c(2,2))
```

This **for-loop** will plot a histogram of each of the four traits.

```
for (i in 1:4) { hist(iris.setosa[,i],xlab=NULL,
main=names(iris.setosa)[i]) }
```

Now the second species

```
iris.virginica <- iris[iris$Species=="virginica",1:4]
for (i in 1:4) { hist(iris.virginica[,i],xlab=NULL,
main=names(iris.virginica)[i]) }
```

Third

```
iris.versicolor <- iris[iris$Species=="versicolor",1:4]
for (i in 1:4) { hist(iris.versicolor[,i],xlab=NULL,
main=names(iris.versicolor)[i]) }
```

reset the plot settings

```
par(mfrow=c(1,1))
```

*6.2a**. To think about: How do the distributions look? Do any of the variables need to be transformed in your opinion?** Normal distributions of variables are not the most essential thing in the PCA unless you have very extreme outliers (like for the American Crime data). More important is that the original variables show (approximately) linear relationships with each other and can be described by linear combinations of the new latent variables (PCs).* ***Hence, it might also be helpful to look at scatterplots between the traits for each of the three species.***

```
plot(iris.setosa[,1:4])
```

Despite that some individual variables show deviations from the normal distribution (e.g. petal width in Iris setosa), the relationship between variables look linear and residuals from linear relationships also seem not to bad from just eye-balling the data (heteroscedasticity is low). The distribution of petal

width in Iris setosa does not improve by log-transformation. If we were too worried about it, we would have to perform more in-depth research on what transformation would be most fitting. However, for this lab we will keep the variable as is as the distribution is not too crazy! Let's accept this data as is and run the PCA, again using the native function **prcomp**.

If your variables are measured on different scales, one of them (the one with most variability) can take over in the analysis and bias it, so it is advisable to correct for that by passing the logical value TRUE to the argument **scale** again (we will analyze the correlation matrix). We perform the PCA on the four continuous variables from the data frame:

```
ir.pca <- prcomp(iris[, 1:4] , scale = TRUE)
```

Look at the output:

```
ir.pca
```

The loadings in the eigen-vectors of each PC represent standardized regression slopes describing how the original variables relate to the PC. PC2 is strongly negatively related to sepal width, with a coefficient of -0.92, also suggesting that PC2 is in essence capturing variation almost exclusively in Sepal width as none of the loadings on PC2 for the three other traits are very strong.

Another way to look at loadings are as weights, where variables contributing more to the PC (i.e. they are more strongly correlated to it) get a higher loading for that PC, but keep in mind that the PC itself might not explain a lot of variance in the data – this is shown by its eigen-value.

Now investigate how much of the variance is explained by each principal component:

```
summary(ir.pca)
```

6.2b. How much of the variance in the data set is explained by the first two principal components together? How much variance does the third explain by itself? Which PCs would you retain for further analysis or graphical exploration?

Now plot the results. Here we project the data over the first 2 Principal

Components (i.e. we plot the PC-scores for each observation), colouring the points by which species they are. We will use the *ggfortify* package, which you need to download and install.

```
install.packages("ggfortify")
library(ggfortify)

autoplot(ir.pca, data = iris, colour = 'Species',
loadings = TRUE, loadings.colour = 'blue',
loadings.label = TRUE, loadings.label.size = 3)
```

The direction of the arrows represents the loading on the two PCs for each original variable (compare with the summary output from the PCA). This also tells you something about the differences between your groupings (species) and individual observations, such that points positioned far out in the same direction as an arrow have higher values for the original variable, and points placed far away on the opposite side of the arrow have lower values.

6.2c. Are there any differences between the three species? Explain the overall graphical result – can you match the output in the summary of the PCA with this figure? What is the loading for sepal length on PC3?

6.3 Using PCA to understand the effects of food in Tadpoles

Here you are going to run a PCA on 7 morphological traits collected from 8 populations of moor frog tadpoles measured under two food treatments (R = Restricted and U = Unlimited). Perform a PCA analysis on this new dataset using what you have learned so far. To save time you can skip graphically inspecting the variables for normality and just assume that they are following a multivariate normal distribution and that the variables are linearly related.

```
dat <- read.table("tadpole_food.txt", header=TRUE)
head(dat)
str(dat)
```

Run PCA:

```
tadpole.pca <- prcomp(dat[,3:9] , scale = TRUE)
```

6.3a How many principal components do you need to explain at least 96% of the variance in tadpole morphology?

I suggest to color your points based on the food factor for better visualization when you plot.

```
autoplot(tadpole.pca, data = dat, colour = 'Food',
loadings = TRUE, loadings.colour = 'blue',
loadings.label = TRUE, loadings.label.size = 3)
```

6.3b Can you see a difference between the two treatments? What does the graph illustrate? Explain the biology behind this.

6.4 MANOVA to test effects of the food treatment. (Advanced Bonus - Optional)

If you have time left, you can try to run a MANOVA on the tadpole data (just to see that this is easy in R). This allows you to statistically test if the food treatment had an effect on tadpole morphology, jointly considering responses in the 7 measured traits. Like in the previous example, we will assume that the assumptions of the MANOVA are fulfilled: multivariate normality; linear relationships; and homogeneous covariance matrices across the compared treatments. Note that the first two assumptions are the same as for the previous multivariate methods, and the last assumption is the multivariate version of the assumption of *homogeneity of variances* in the ANOVA.

The population variable is not a factor yet. Let's do something about that first:

```
dat$Pop <- factor(dat$Pop)
```

Then we run a MANOVA to see if there are differences in morphology based on food treatment based on all seven traits, using the base function **manova()** in R:

```
mod.MAN <- manova(cbind(Muscle_Width, Body_Width, Body_Length,
Tail_length, Body_depth, Muscle_depth, Tail_depth) ~ Food , data=dat )

summary(mod.MAN)

mod.MAN
```

6.4a Was there an effect? Look to see that you understand the gist of this MANOVA table. You get whether the first dimension (or discriminant function) is significant based on an F-ratio and Pillai's trace test statistic (the variation along the discriminant function explained by Food divided by the total variation along the discriminant function – so, the higher the value of the test statistic, the more significant the p-value).

6.4b You also get the amount of variation in the seven original variables captured by the discriminant function, and the residual variation that was not captured. Can you reconcile these values with the effects shown in the previous PCA-plot?

Let’s include the information on population and consider that populations can respond differently to the food treatment:

```
mod.MAN2 <- manova(cbind(Muscle_Width, Body_Width, Body_Length, Tail_length, Body_depth, Muscle_depth, Tail_depth) ~ Food*Pop , data=dat )

summary(mod.MAN2)
```

6.4c What was your conclusion about the effect of Food treatment and populations? You can also compare the population effect by projecting samples coloured by their population identity onto PC1 and PC2 for comparison to your previous graph. Does this graph make sense with reference to the MANOVA table and the estimated effects of [Pop] and [Pop:Food]?

```
autoplot(tadpole.pca, data = dat, colour = 'Pop',
loadings = TRUE, loadings.colour = 'blue',
loadings.label = TRUE, loadings.label.size = 3)
```