Dominik Bahlburg
4.4.2017

PCA (Principal Component Analysis)

What is it good for?

Finding patterns in your data and identify the principal components which are responsible for these patterns

Can reduce multidimensional datasets into fewer dimensions to understand them better

Principal component analysis (PCA) has been called one of the most valuable results from applied linear algebra. PCA is used abundantly in all forms of analysis - from neuroscience to computer graphics - because it is a **simple, non-parametric** method of **extracting relevant information from confusing data sets**. With minimal additional effort PCA provides a roadmap for how to **reduce a complex data** set to a lower dimension to **reveal the sometimes hidden, simplified dynamics** that often underlie it.

What is it good for?

Look, we have some wine bottles standing here on the table. We can describe each wine by its colour, by how strong it is, by how old it is, and so on (see this very nice visualization of wine properties taken from here). We can compose a whole list of different characteristics of each wine in our cellar. But many of them will measure related properties and so will be redundant. If so, we should be able to summarize each wine with less characteristics! This is what PCA does.

“amoeba” at Stackoverflow

What’s the math behind?

Basic statistics:

- Mean
- Covariance

Basic matrix algebra:

- Eigenvectors
- Eigenvalues
- matrix multiplication

The steps

We start with a dummy dataset

1. (Standardize/center the data)
2. Calculate the covariance-matrix
3. Find eigenvectors and eigenvalues of covariance-matrix
4. Select n largest eigenvalues and corresponding eigenvectors -> principal components
5. plot/rotate data around principal components
6. How to interpret your PCA's
7. How to do it in R

wine	a	b	c
Chardonnay	1	3	5
Burgundy	2.3	5.2	8.2
Chardonnay	0.7	2.1	3
Riesling	1.5	4.3	6.6
Burgundy	1.8	4.9	7.5

1. Standardize/center the data

Why?:

Let’s look at it from the other side… If you don’t standardize/center the data, each variable will contribute to covariance and thus eigenvectors and eigenvalues with the ”magnitude of its own range”

that’s a problem when:

- variables do not have the same unit
- e.g. in taxonomic data when you have very dominant species with high values (can be problematic in dependence on what you’re looking for)

<u>Standardizing the data: each variable contributes equally to covariance</u>

Summary: **You** have to decide whether you want to standardize your data prior to PCA or not. It’s your decision and it should be dependent on your data structure and on what you’re looking for.

In case you standardize your data: **covariance-matrix = correlation-matrix**

1. Center the data

wine	a	b	c
Chardonnay	1	3	5
Burgundy	2.3	5.2	8.2
Chardonnay	0.7	2.1	3
Riesling	1.5	4.3	6.6
Burgundy	1.8	4.9	7.5

$mean_{var1} = 1.46$

$mean_{var2} = 3.9$

$mean_{var3} = 6.06$

wine	a	b	c
Chardonnay	-0.46	-0.9	-1.06
Burgundy	0.84	1.3	2.14
Chardonnay	-0.76	-1.8	-3.06
Riesling	0.04	0.4	0.54
Burgundy	0.34	1	1.44

Reason: now dataset has a mean of zero!

2. Create covariance matrix

What is covariance?

Covariance is a measure of how changes in one variable are associated with changes in a second variable. Specifically, covariance measures the degree to which two variables are linearly associated. (http://stats.stackexchange.com/questions/29713/what-is-covariance-in-plain-language)

$$Cov_{xy} = \frac{\Sigma(x-\bar{x})(y-\bar{y})}{(n-1)} \quad \leftrightarrow \quad s^2 = \frac{\sum(x-\bar{x})^2}{n-1}$$

Cov = 0 -> variables are not linearly associated

Cov >> 0 -> variables are linearly associated (higher values of x correspond with higher values of y)

Cov << 0 -> variables are linearly associated (high values of x correspond with low values of y or vice versa)

2. Create covariance matrix

a	b	c
-0.46	-0.9	-1.06
0.84	1.3	2.14
-0.76	-1.8	-3.06
0.04	0.4	0.54
0.34	1	1.44

$$\begin{bmatrix} Var_a & Cov_{a,b} & Cov_{c,a} \\ Cov_{a,b} & Var_b & Cov_{c,b} \\ Cov_{a,c} & Cov_{b,c} & Var_c \end{bmatrix}$$

$$\begin{bmatrix} 0.4030 & 0.8075 & 1.2805 \\ 0.8075 & 1.725 & 2.7250 \\ 1.2805 & 2.7250 & 4.3580 \end{bmatrix}$$

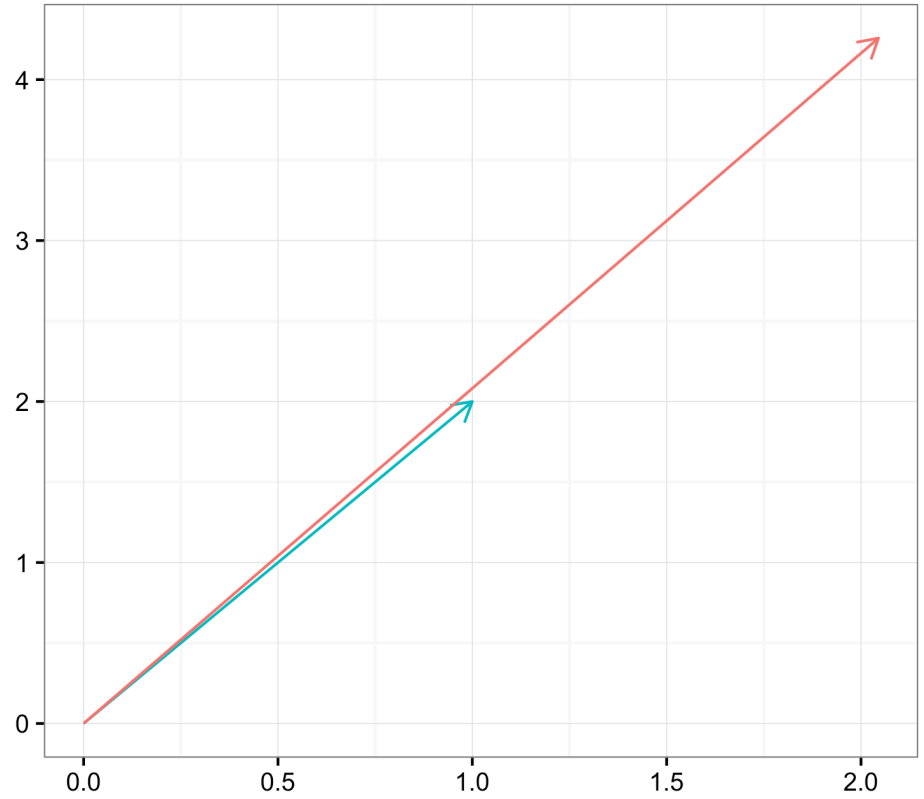
All variables increase together – surprise surprise…

3. Find Eigenvalues and Eigenvectors of Covariance-Matrix

What on earth are those things?

We can look at a matrix like at a function

$$\begin{bmatrix} 0.4030 & 0.8075 \\ 0.8075 & 1.725 \end{bmatrix} \times \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 2.045 \\ 4.2575 \end{pmatrix}$$

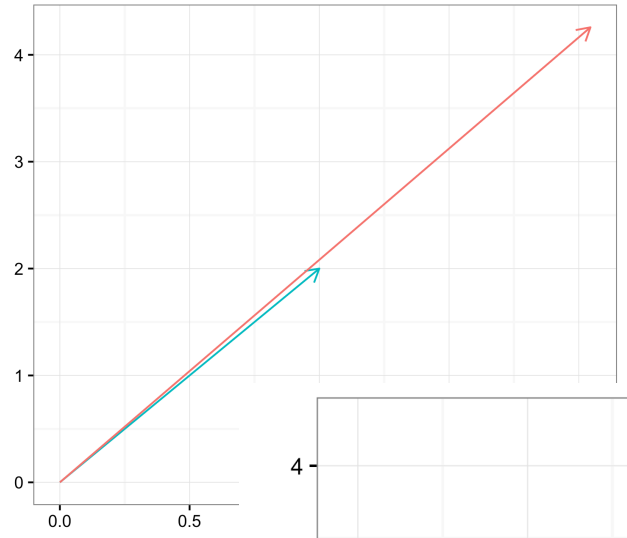


Directions of this transformation: **eigenvectors!**

Magnitude of transformation in direction of the respective eigenvectors: **eigenvalues!**

3. Find Eigenvalues and Eigenvectors of Covariance-Matrix

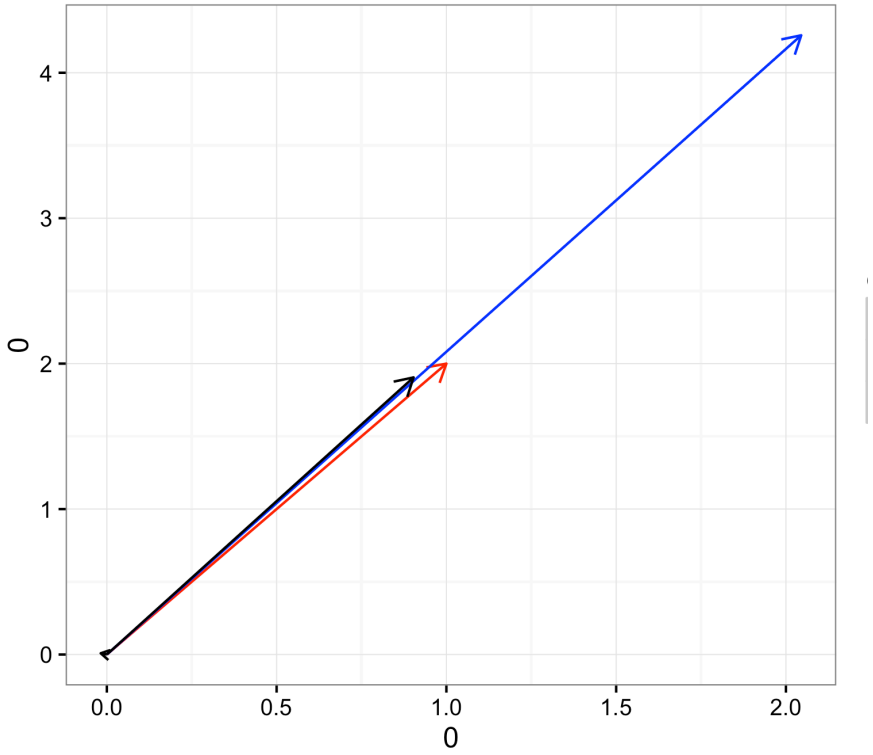
$$\begin{bmatrix} 0.4030 & 0.8075 \\ 0.8075 & 1.725 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2.045 \\ 4.2575 \end{bmatrix}$$



Eigenvalues: $\lambda_1 = 2.108$
$\lambda_2 = 0.020$

Eigenvectors: $e1 = \begin{bmatrix} 0.428 \\ 0.904 \end{bmatrix}$

$e2 = \begin{bmatrix} -0.904 \\ 0.428 \end{bmatrix}$



3. Find Eigenvalues and Eigenvectors of Covariance-Matrix

Eigenvectors and their corresponding eigenvalues give information about:

The “direction” of a matrix (eigenvectors)

The magnitude each direction has (eigenvalues)

-> getting these information from our covariance matrix reveals a lot about the data structure and the how strong the different **components** shape and characterize the data

3. Find Eigenvalues and Eigenvectors of Covariance-Matrix

$$\begin{bmatrix} 0.4030 & 0.8075 & 1.2805 \\ 0.8075 & 1.725 & 2.7250 \\ 1.2805 & 2.7250 & 4.3580 \end{bmatrix}$$

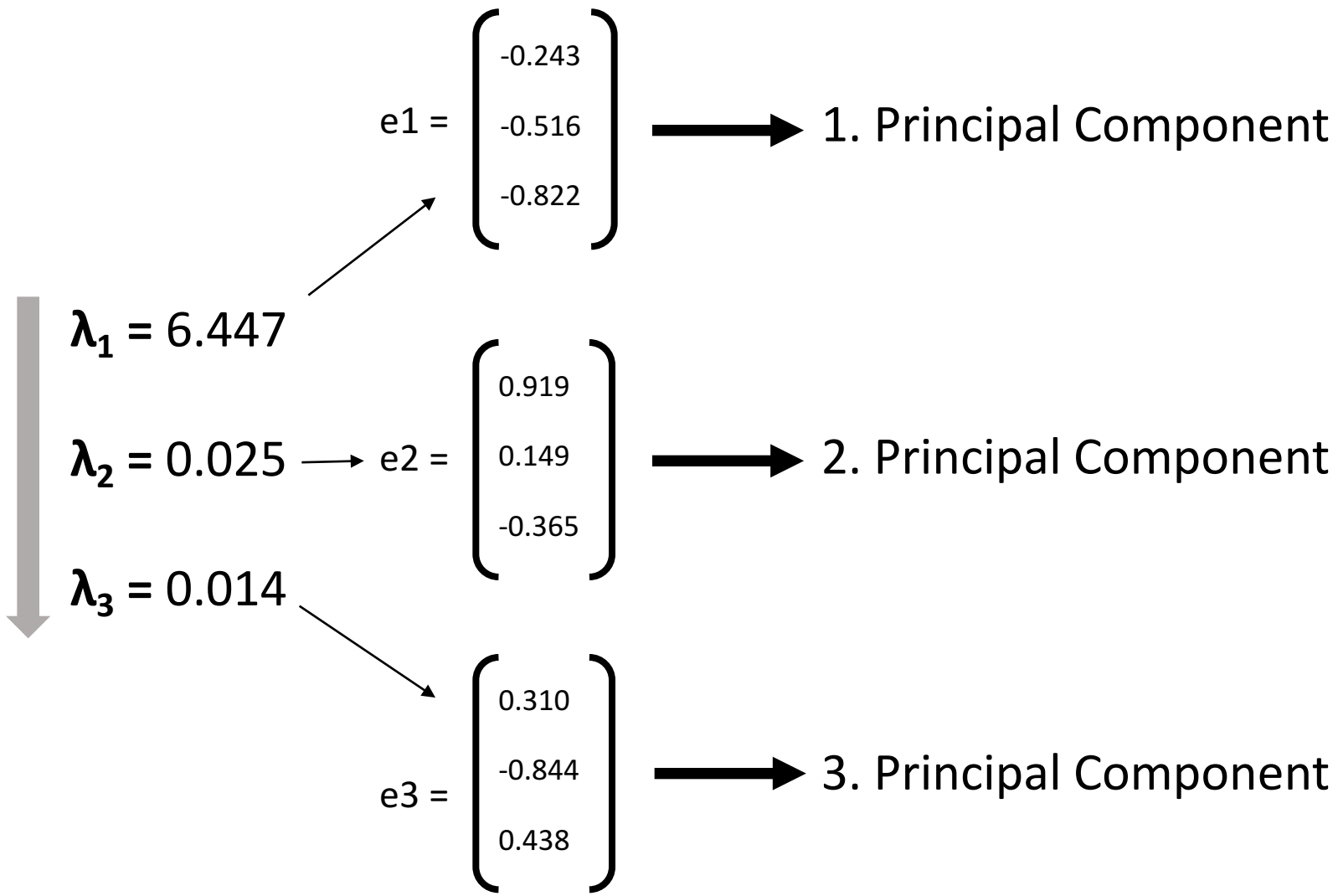
$\lambda_1 = 6.447$
$\lambda_2 = 0.025$
$\lambda_3 = 0.014$

$$e1 = \begin{pmatrix} -0.243 \\ -0.516 \\ -0.822 \end{pmatrix}$$

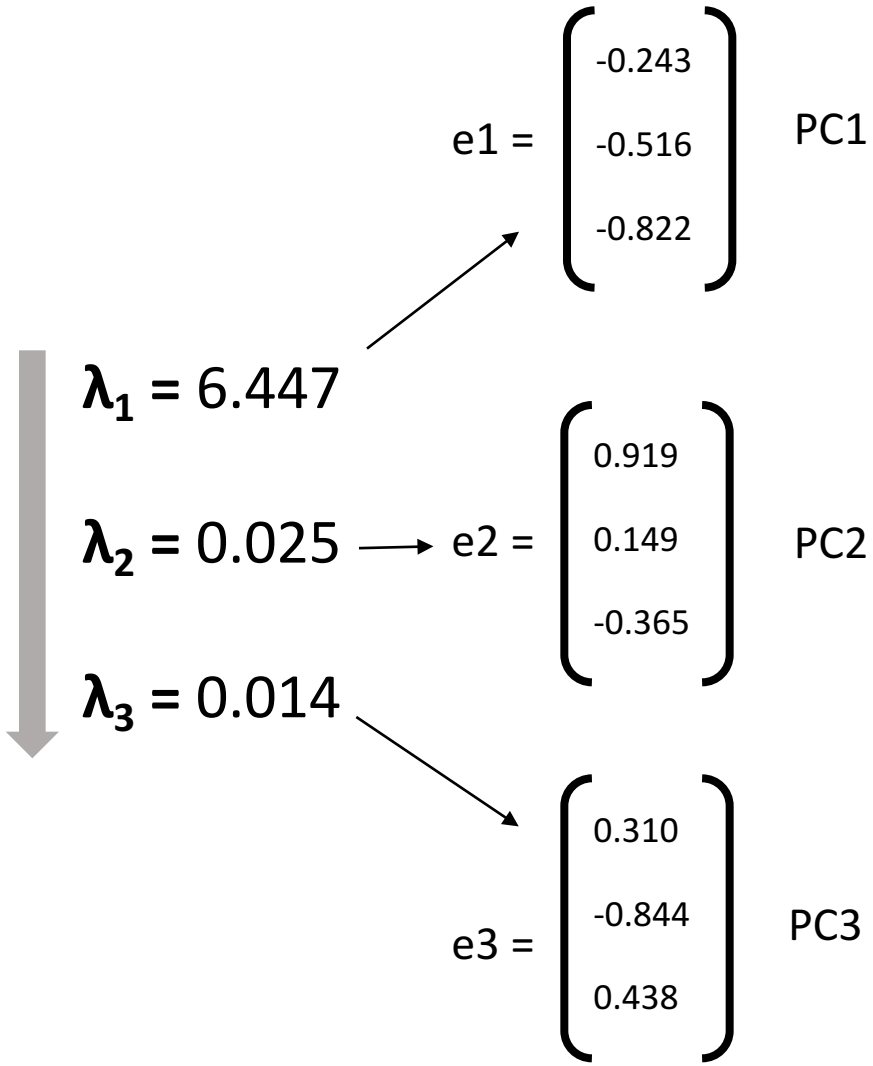
$$e2 = \begin{pmatrix} 0.919 \\ 0.149 \\ -0.365 \end{pmatrix}$$

$$e3 = \begin{pmatrix} 0.310 \\ -0.844 \\ 0.438 \end{pmatrix}$$

4. Rank the Eigenvalues



4. Rank the Eigenvalues



Loadings of the PCs as linear combinations of the different variables

	PC1	PC2	PC3
a	0.2429645	0.9190116	0.3104607
b	0.5156843	0.1487096	-0.8437744
c	0.8216070 -	0.3651069	0.4377887

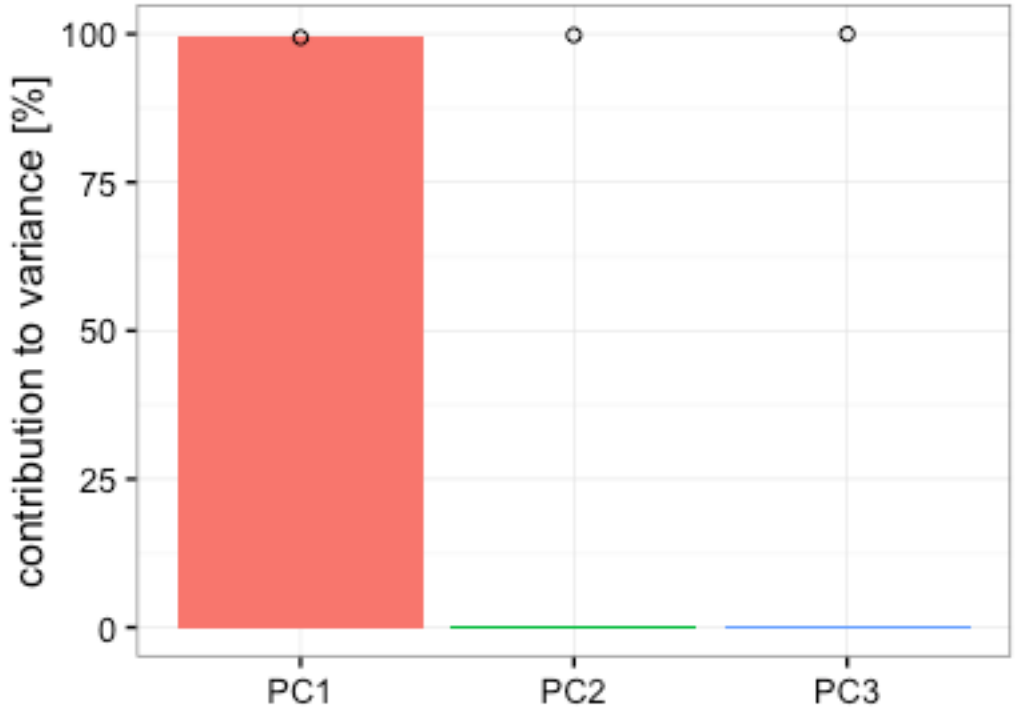
Importance of components:

	PC1	PC2	PC3
Standard deviation	2.539	0.15794	0.11846
Proportion of Variance	0.994	0.00385	0.00216
Cumulative Proportion	0.994	0.99784	1.00000

4. Rank the Eigenvalues

Importance of PC's:

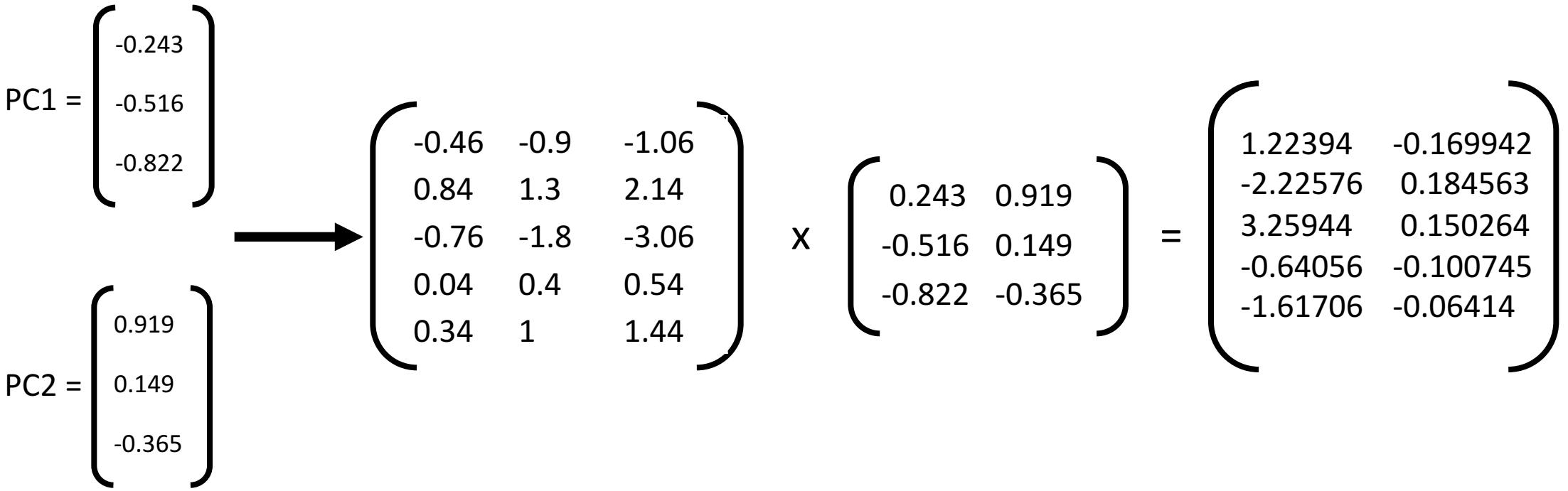
	PC1	PC2	PC3
Standard deviation	2.539	0.15794	0.11846
Proportion of Variance	0.994	0.00385	0.00216
Cumulative Proportion	0.994	0.99784	1.00000



>99% of the data variance is explained by the first Principal Component

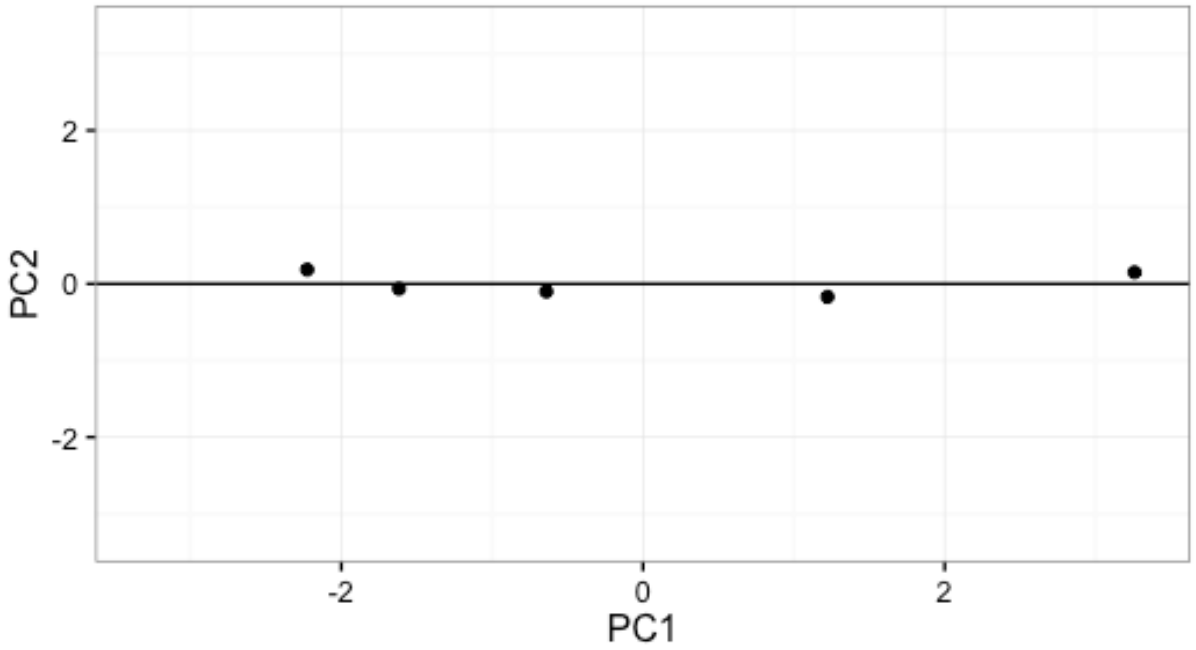
In this step there it is up to you to decide how many PC's you want to keep. Unfortunately, there is no rule how many you should keep or how much variance should be explained by your chosen components.

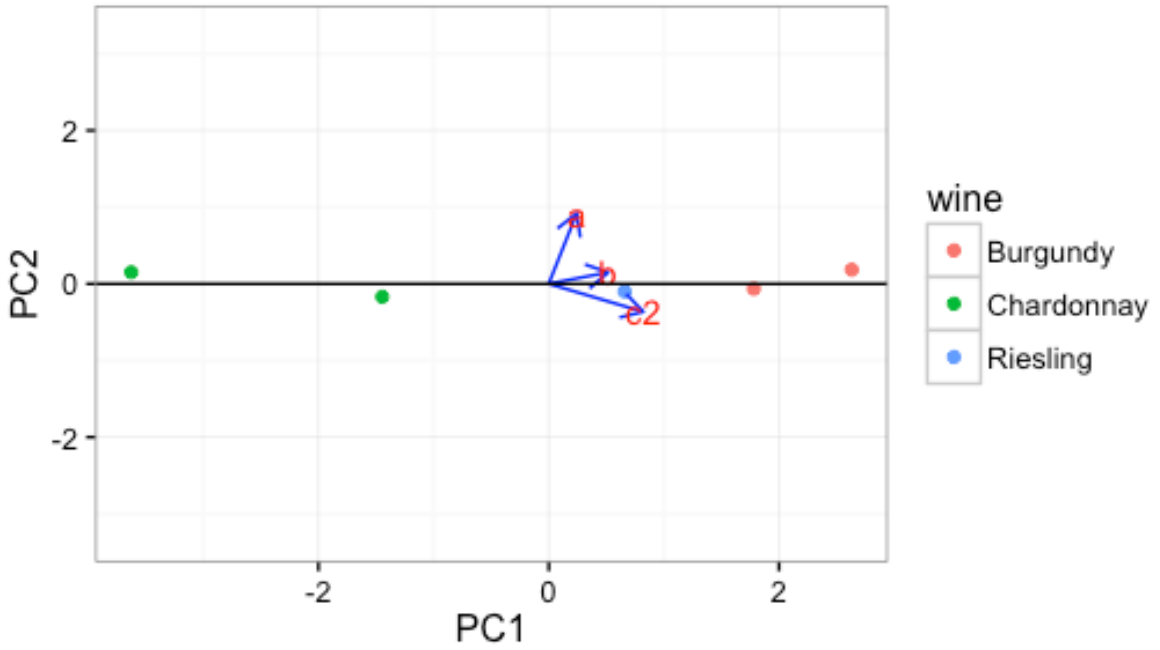
5. Transform/Rotate your data around picked PC's



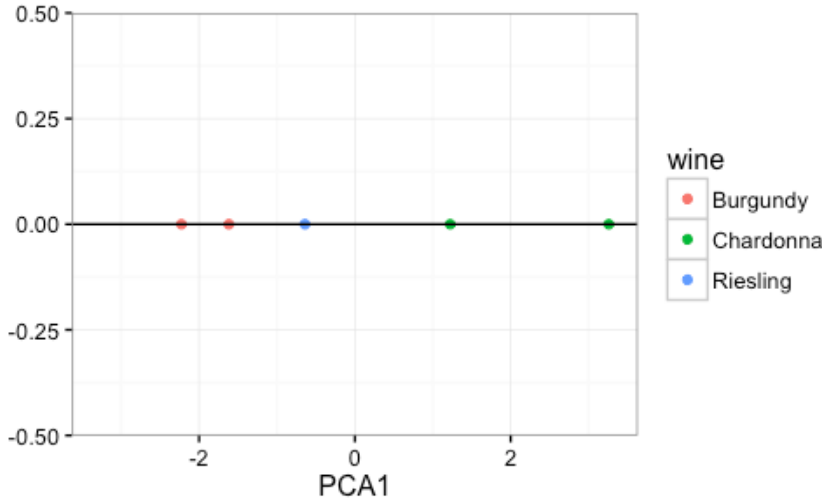
Dimensions reduced from 3 to 2!!

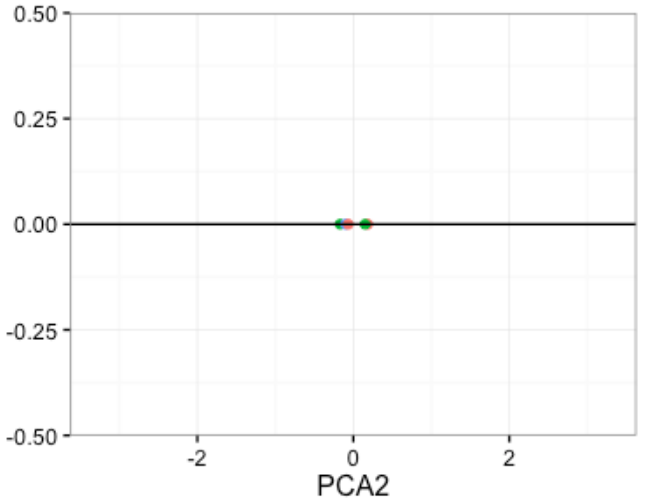
5. Transform/Rotate your data around picked PC's

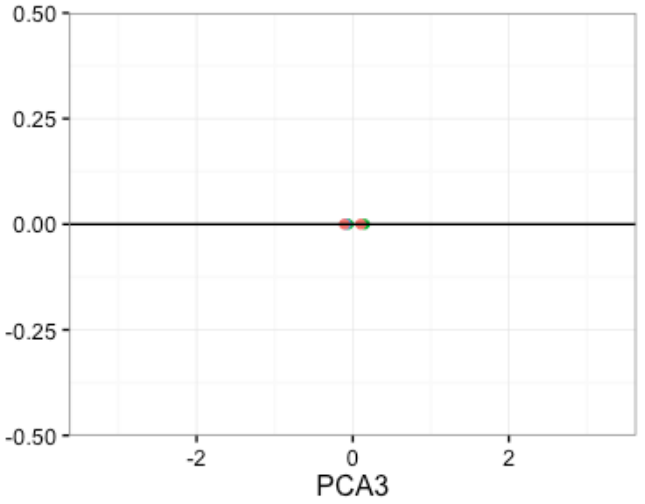




5. Transform/Rotate your data around picked PC's







PCA1 – best discriminator

PCA2 – worse discriminator

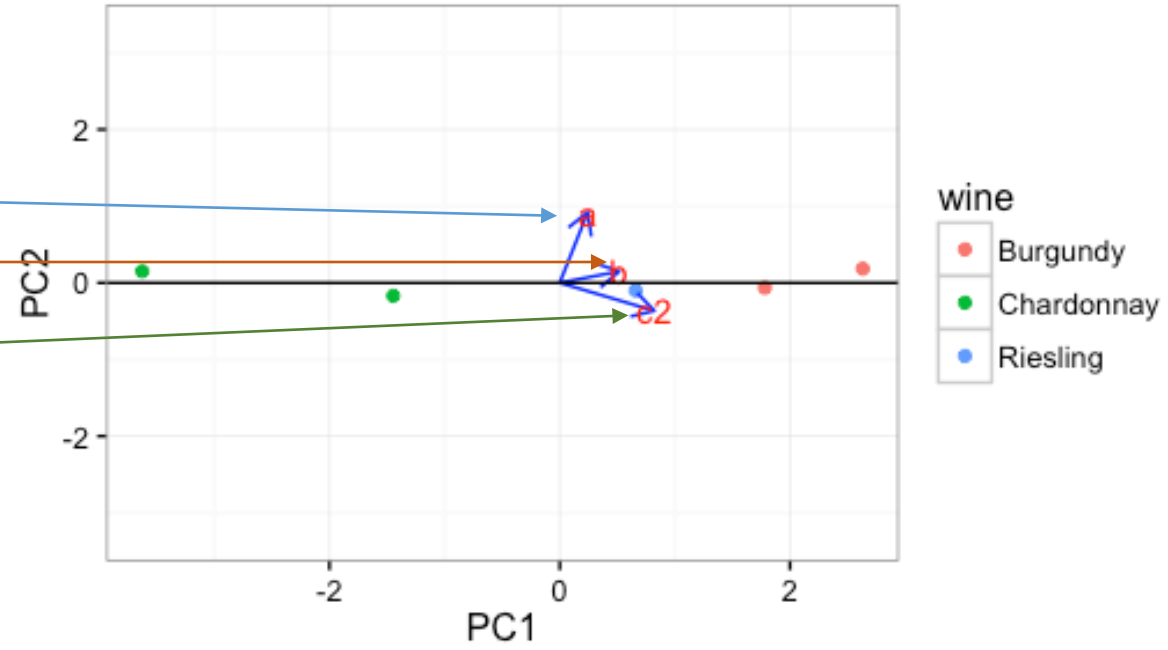
PCA3 – worst discriminator

But: the fewer dimensions the more loss of information…

6. How to interpret your PCA's

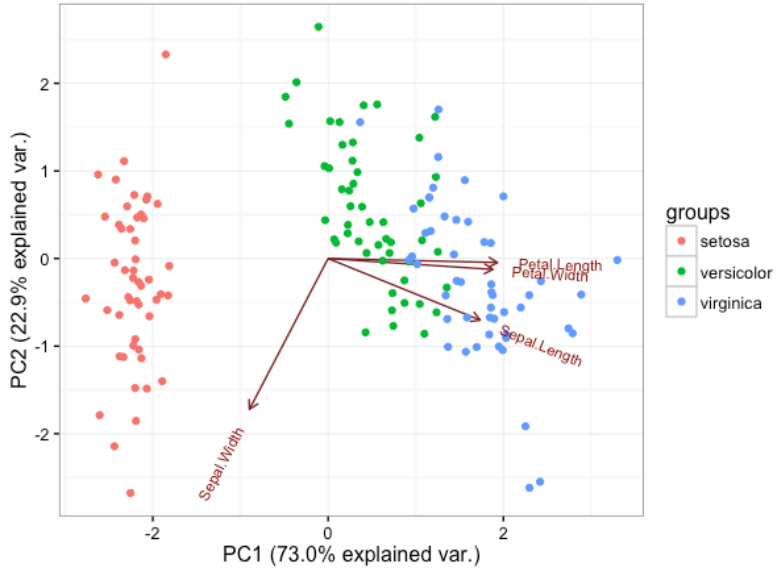
Loadings of the PCA's show importance of original factors for data discrimination

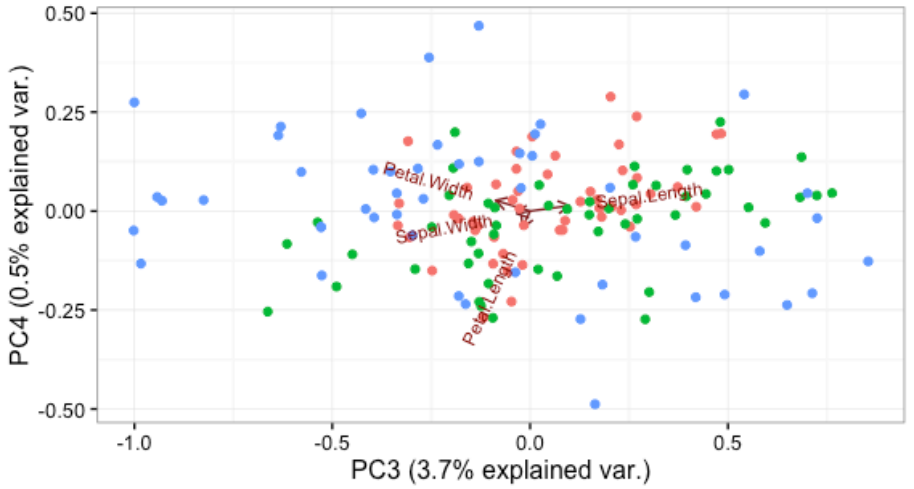
	PC1	PC2
a	0.2429645	0.9190116
b	0.5156843	0.1487096
c	0.8216070	-0.3651069



7. How to do it in R

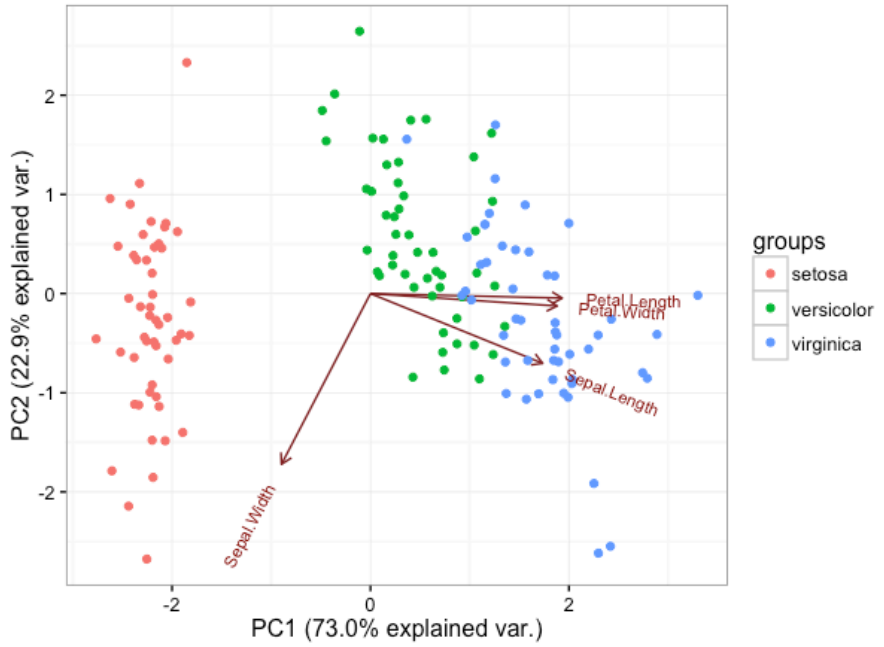
```
library(devtools)
install_github("vqv/ggbiplot")
library(ggbiplot)
iris <- iris
pca_iris <- prcomp(iris[,1:4], scale=T)
species <- iris$Species
pca_iris
summary(pca_iris)
ggbiplot(pca_iris, choices = 1:2,obs.scale = 1, var.scale = 1, groups = species)+
theme_bw()
ggbiplot(pca_iris, choices = 3:4,obs.scale = 1, var.scale = 1, groups = species)+
theme_bw()
```





7. How to do it in R

ggbiplot(pca_iris, choices = **1:2**,obs.scale = 1,
var.scale = 1, groups = species)+
theme_bw()



ggbiplot(pca_iris, choices = **3:4**,obs.scale = 1,

var.scale = 1, groups = species)+

theme_bw()