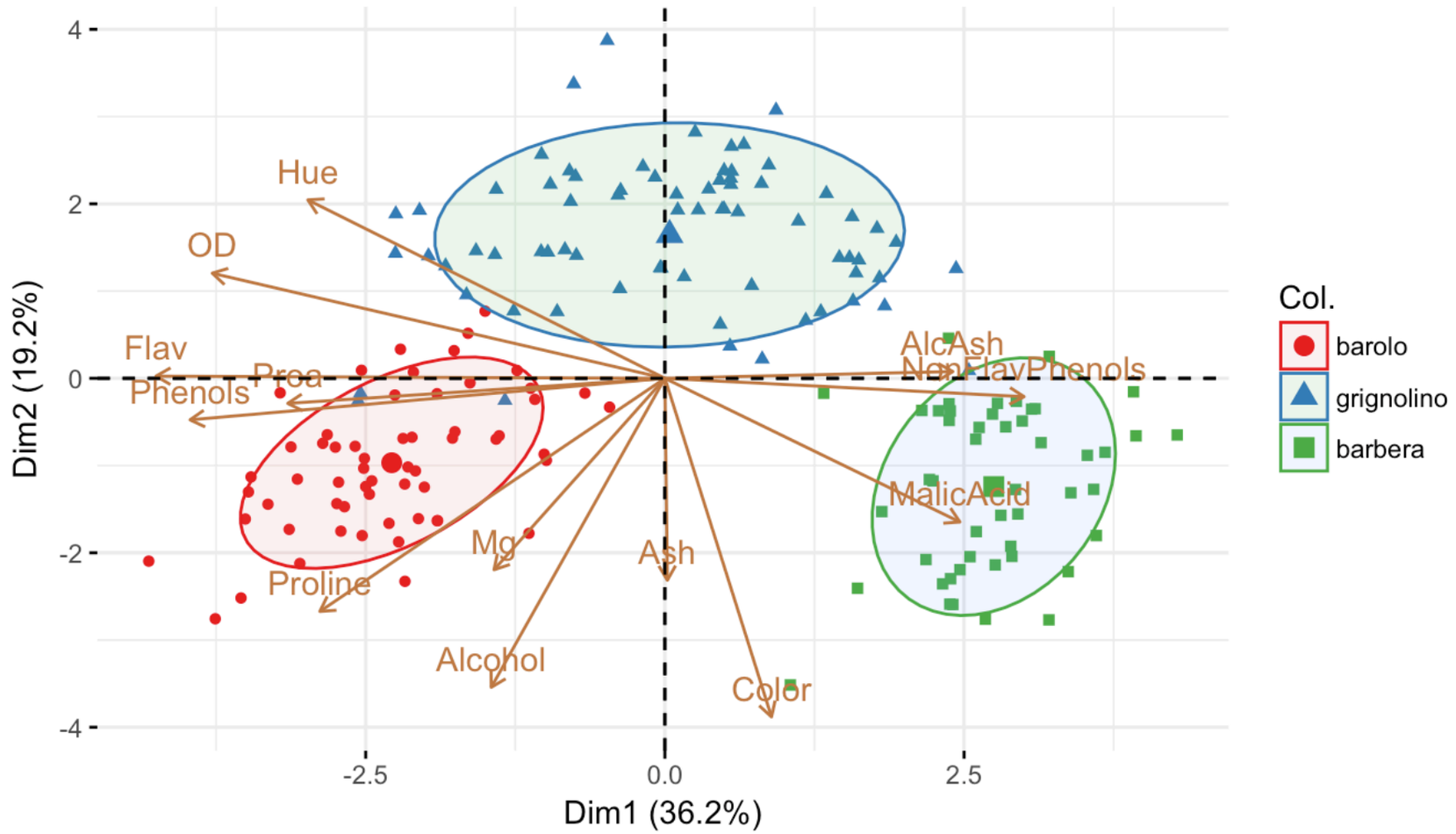


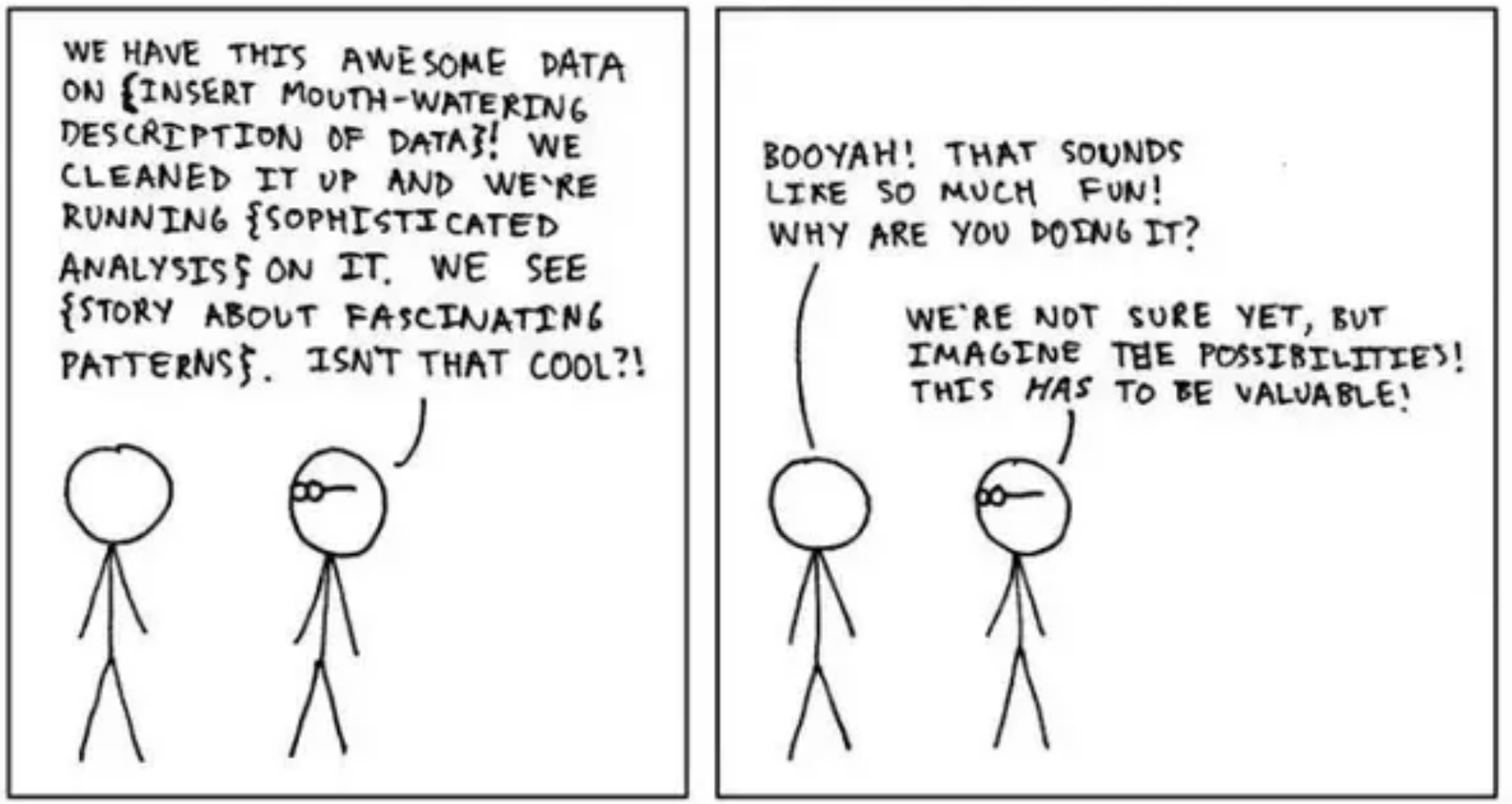
Multivariate Analysis

Lan Huong Nguyen

July 2, 2019

Outline

1. Introduction to Multivariate Analysis
2. Data matrices
3. Dimensionality reduction
4. PCA analysis from start to finish



Introduction

What is multivariate analysis?

- Most studies collect information on **multiple variables** repeated for each observation.
- Modern datasets are often **high-dimensional**, storing data on thousands to millions of variables (dimensions) per sample.

What is multivariate analysis?

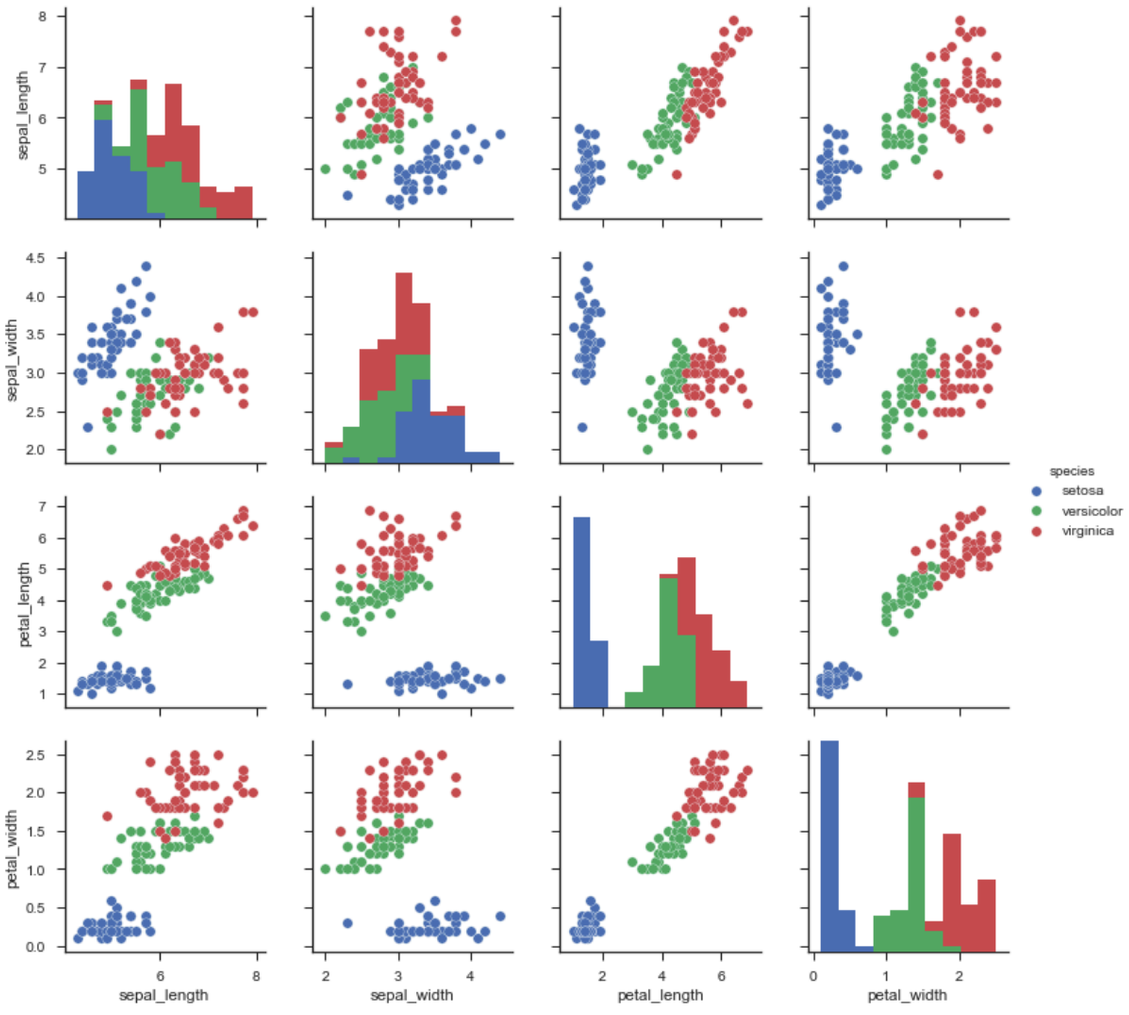
- Most studies collect information on **multiple variables** repeated for each observation.
- Modern datasets are often **high-dimensional**, storing data on thousands to millions of variables (dimensions) per sample.
- **Multivariate Analysis (MVA) is a technique to analyze more than one variable at a time.**

What is multivariate analysis?

- Most studies collect information on **multiple variables** repeated for each observation.
- Modern datasets are often **high-dimensional**, storing data on thousands to millions of variables (dimensions) per sample.
- Multivariate Analysis (MVA) is a technique to analyze more than one variable at a time.
- **MVA encompasses many statistical methods:**
  - PCA,
  - Correspondence Analysis,
  - Factor Analysis,
  - Multidimensional Scaling,
  - MANOVA, and many others.

Why do you need multivariate analysis?

- In a lot of cases, you are not interested in how a single variable behaves on its own, but **how all variables or a group of variables behave together jointly** or how they affect each other.

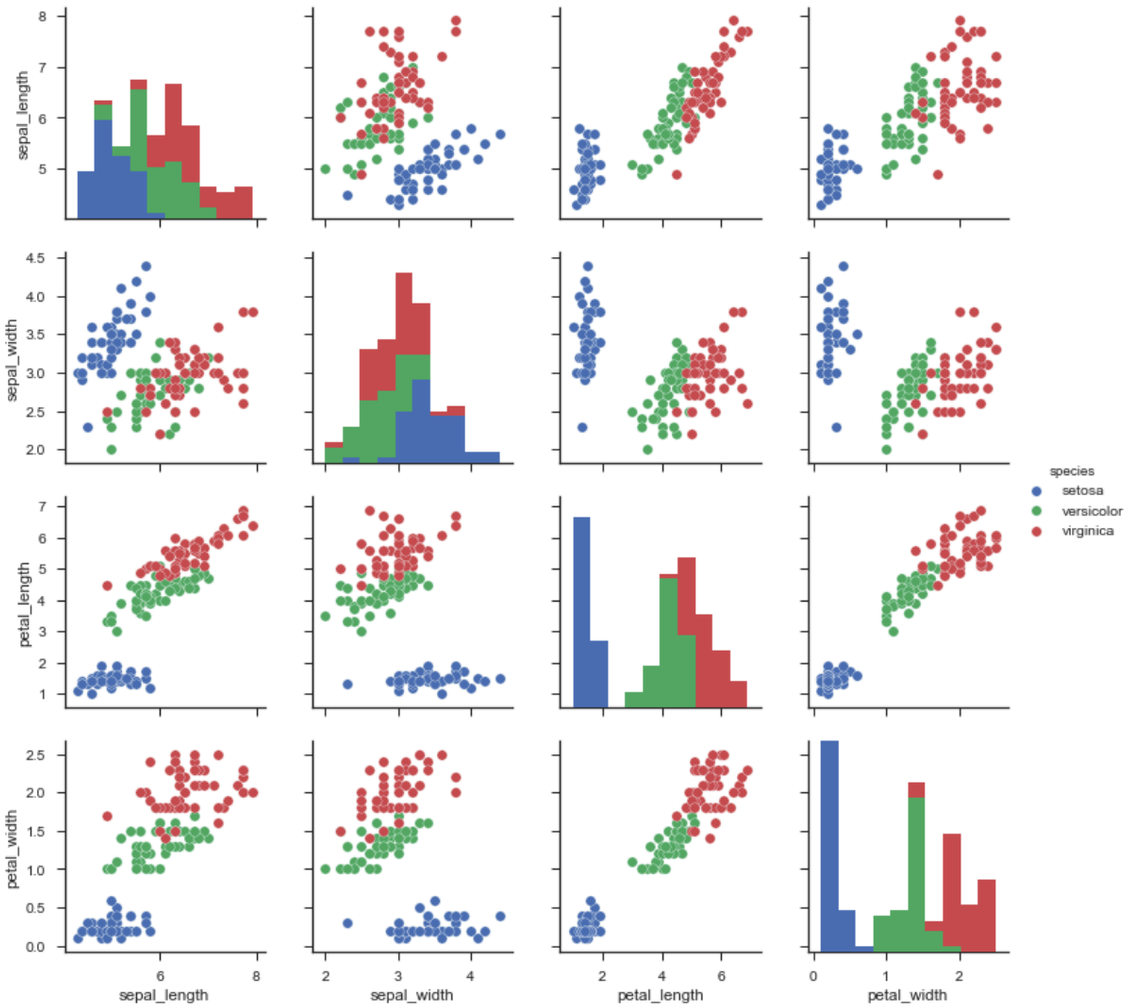


Why do you need multivariate analysis?

- In a lot of cases, you are not interested in how a single variable behaves on its own, but **how all variables or a group of variables behave together jointly** or how they affect each other.

- The main purpose of MVA is to **investigate connections or associations** between different measured variables.

- If all **variables are independent, then we should study them separately** one by one using standard univariate statistics.



Goals for this lecture

- See **examples of multivariate data** coming up in biological studies and others.
- Learn how to **preprocess, center and rescale data** before the analysis.
- Understand the **importance of correlations** between variables.
- Recognize why **dimensionality reduction** is useful.
- Build new variables, called **principal components** (PC), that are more useful than the original measurements.
- See **what is "under the hood" of PCA** and learn how to choose the number of principal components.
- Generate **helpful visualizations of PCA** results.
- **Run through a complete PCA analysis** from start to finish.

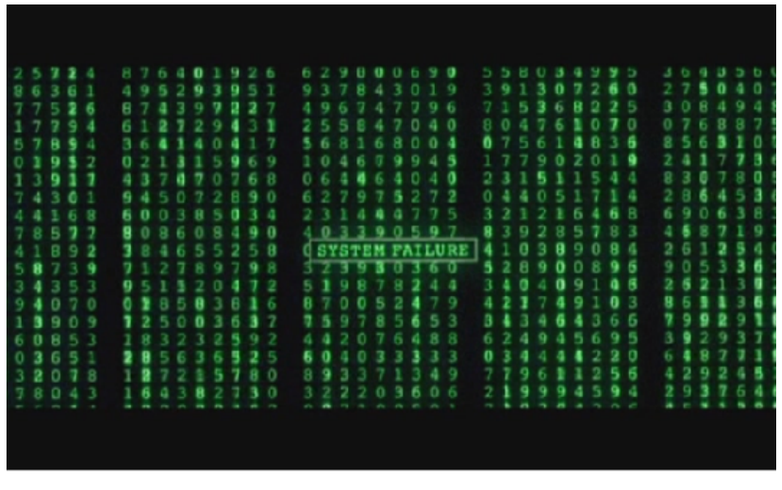
Datasets

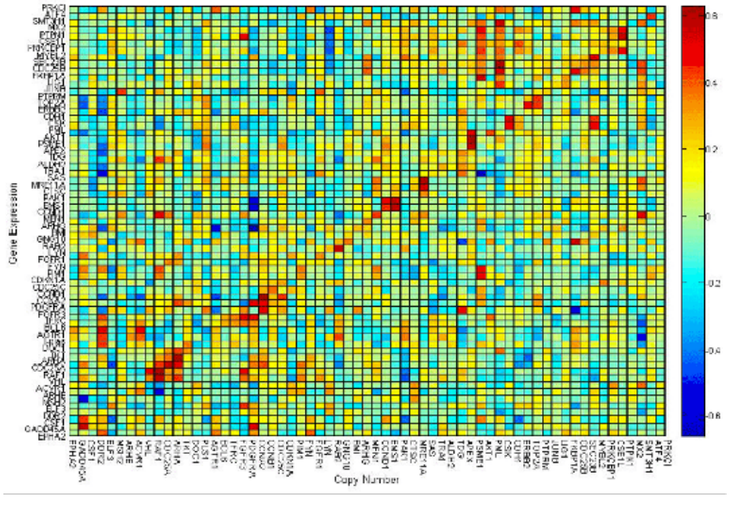
Data in a matrix format

- Datasets can often be organized in a form of a rectangular matrix.
- Usually, rows correspond to observations (e.g. samples, replicates, patients), and columns to variables (features characterizing the observations).

Data in a matrix format

- Datasets can often be organized in a form of a rectangular matrix.
- Usually, rows correspond to observations (e.g. samples, replicates, patients), and columns to variables (features characterizing the observations).
- Each matrix entry corresponds to a measurement of a particular feature of a particular sample.
- Variables might have different ranges or even different units.





A toy example

- Recall the **`mtcars`** data frame used in lecture on graphics.
- The dataset comes from 1974 *Motor Trend* US magazine and comprises multiple aspects of automobile design and performance for 32 automobiles.
- The data contains **both categorical and continuous variables** in various units.

```
head(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	4	4
Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
Hornet 4 D rive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.46	20.22	1	0	3	1

Clinical datasets

- Clinical datasets consist of patient data and can contain both numerical and categorical variables.
- Example: **`diabetes`** dataset was collected by Reaven and Miller (1979) to study the relationship among blood chemistry measures of glucose tolerance and insulin.
- Data measures glucose levels in the blood after fasting (`glufast`), after a test condition (`glutest`), steady state plasma glucose (`steady`) and steady state insulin (`insulin`) for 145 non-obese adults.
- The last variable is a categorical, indicating diagnostic group membership (1=overt diabetic, 2=chemical diabetic, 3=normal).

```
diabetes=read.table(url("http://bios221.stanford.edu/data/diabetes.txt"),
                    header=TRUE,row.names=1)
diabetes[1:4,]
```

diabetes

##	relwt	glufast	glutest	steady	insulin	Group
## 1	0.81	80	356	124	55	3
## 3	0.94	105	319	143	105	3
## 5	1	90	323	240	143	3
## 7	0.91	100	350	221	119	3

Microbial ecology

- Microbial species abundances are estimated using sequencing.
- Data is aggregated as a matrix of read counts:
  - Columns represent different bacterial species (or seq. variants),
  - Rows correspond to samples that were sequenced,
  - Entries are integers representing the number times a specific bacterial species was observed in each of the samples.

```
data("GlobalPatterns", package = "phyloseq")
GPOTUs = as.matrix(t(phyloseq::otu_table(GlobalPatterns)))
GPOTUs[1:4, 6:13]
```

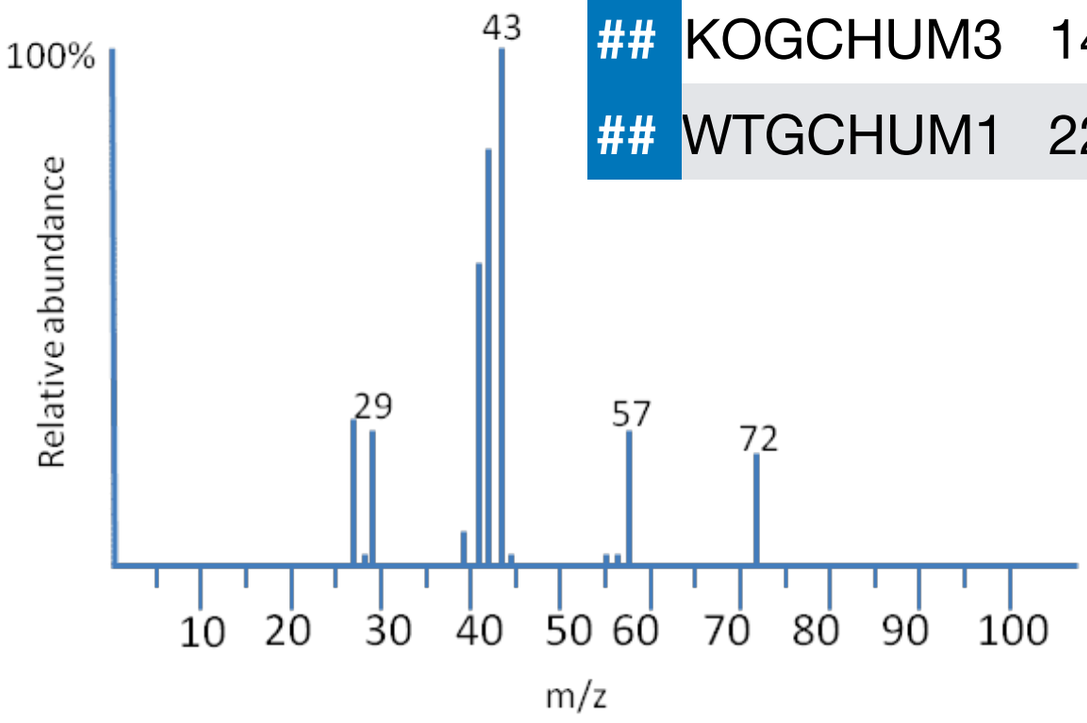
`GlobalPatterns` Microbial Abundance Data

##		246140	143239	244960	255340	144887	141782	215972	31759
##	CL3	0	7	0	153	3	9	0	0
##	CC1	0	1	0	194	5	35	3	1
##	SV1	0	0	0	0	0	0	0	0
##	M31Fcsw	0	0	0	0	0	0	0	0

Mass spectrometry

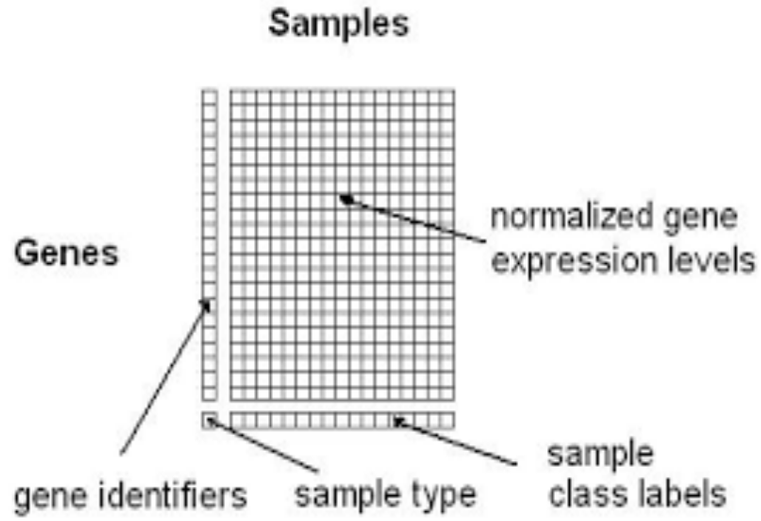
- Mass spectrometry measures **continuous intensities** quantifying the **concentration of small-molecule chemicals** found within a biological samples.
- Columns of the data matrix correspond to mass spectroscopy peaks or molecules identified by their m/z-ratios.
- Rows (samples) correspond to samples. Here, samples are taken from either knockout or wild-type mice.

##	146.099	148.705	310.151	132.451
## KOGCHUM1	29932.36	17055.70	1132.82	785.51
## KOGCHUM2	94067.61	74631.69	28240.85	5232.05
## KOGCHUM3	146411.33	147788.71	64950.49	10283.00
## WTGCHUM1	229912.57	384932.56	220730.39	26115.20



Expression Data

- **Cell Types Microarray:** Here, the rows are samples from different subjects and different T-cell types and the columns are expressed genes



	X3968	X14831	X13492	X5108	X16348	X585
HEA26_EFFE_1	-2.61	-1.19	-0.06	-0.15	0.52	-0.02
HEA26_MEM_1	-2.26	-0.47	0.28	0.54	-0.37	0.11
HEA26_NAI_1	-0.27	0.82	0.81	0.72	-0.9	0.75
MEL36_EFFE_1	-2.24	-1.08	-0.24	-0.18	0.64	0.01
MEL36_MEM_1	-2.68	-0.15	0.25	0.95	-0.2	0.17

continuous

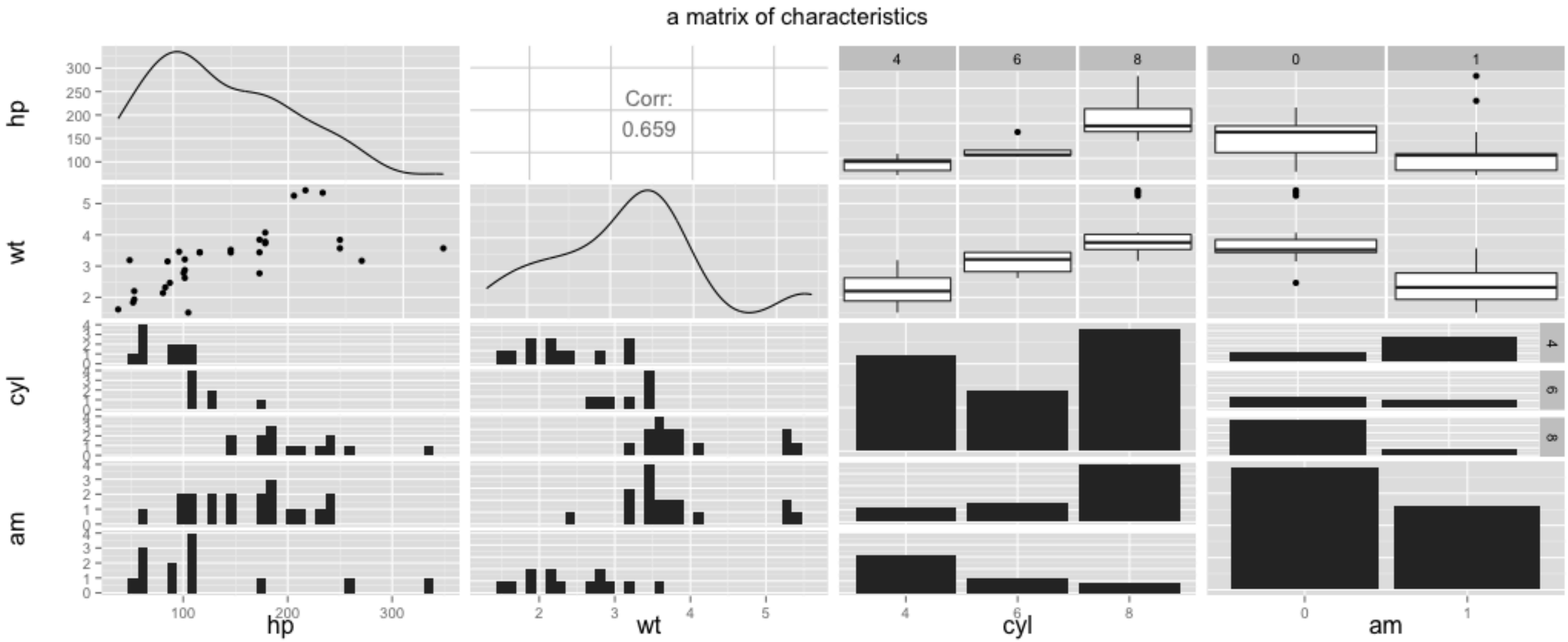
- **mRNA reads:** RNA-Seq transcriptome data report the number of sequence reads matching each gene in each of several biological samples.

	FBgn0000017	FBgn0000018	FBgn0000022	FBgn0000024	FBgn0000028	FBgn0000032
untreated1	4664	583	0	10	0	1446
untreated2	8714	761	1	11	1	1713
untreated4	3150	310	0	3	0	672
treated1	6205	722	0	10	0	1698
treated3	3334	308	0	5	1	757

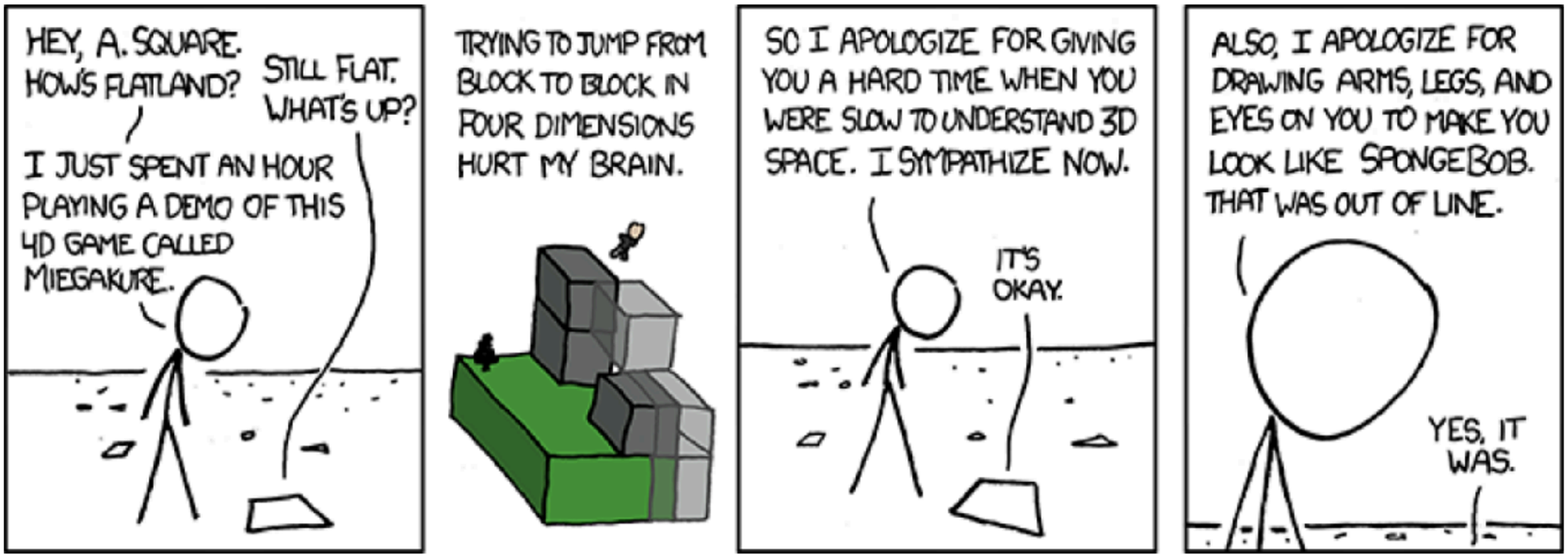
integers

Data summaries and visualization

- It is always beneficial to **start a multidimensional analysis by checking the simple one dimensional and two dimensional summary statistics**, we can visualize these using a graphics package that builds on 'ggplot2' called 'GGally'.



Data summaries



- If we study only a single variable (e.g. a particular column in data matrix), we are dealing with one dimensional data.
- Useful summaries for 1D data include:
  - **means or medians** (zero-dimensional summary of 1D data).
  - **histogram** showing the variable's distribution.

Correlation coefficients

In lecture 3 on graphics we studied two dimensional scatterplots.

When considering two variables **x** and **y** measured together on a set of observations, the **correlation coefficient measures how the variables co-vary linearly**.

This is a **single number summarizes two dimensional data**, its formula involves the sample mean of x and y, denoted $\bar{x}$ and $\bar{y}$.

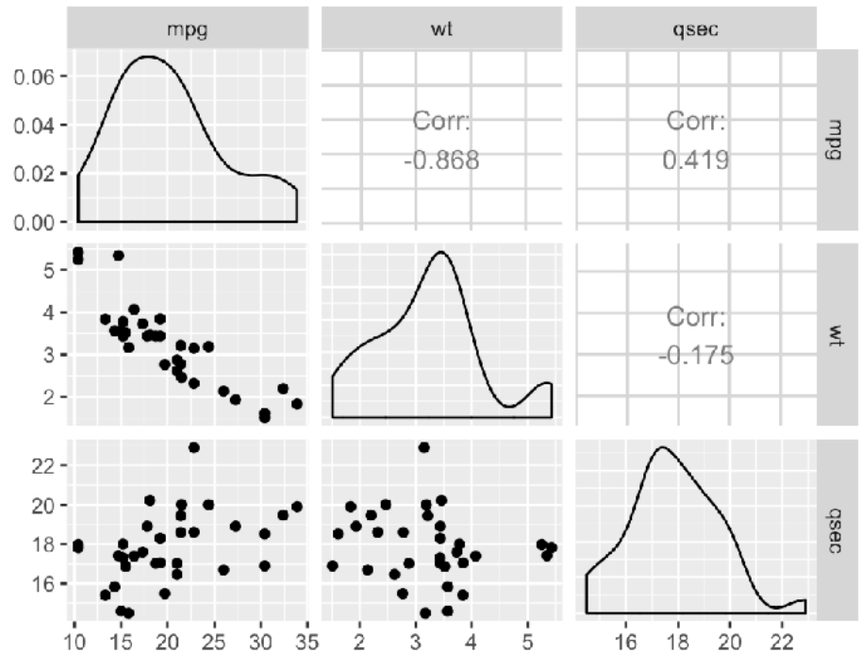
$$\hat{\rho} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Get the correlation coefficient (*r*) from your calculator or computer

- *r* has a value between -1 and +1:

ggally::ggpairs()

```
library(dplyr)
library(GGally)
my_mtcars <- select(mtcars, mpg, wt, qsec)
ggpairs(my_mtcars)
```




```
my_mtcars <- select(mtcars, mpg, wt, qsec, cyl, gear, am) %>%
  mutate(cyl = factor(cyl), gear = factor(gear), am = factor(am))
ggpairs(my_mtcars, aes(color = cyl))
```

Data preprocessing

Centering

- We usually **center the cloud of points around the origin**; the most common way of doing this is to make new variables whose means are all zero.
- However, sometimes centering is better achieved by making each column's median equal zero.
- **Centering using the median is a more robust** procedure; the result is much less influenced by outliers.

Scaling

- In many cases, different variables are measured in **different units, and at different scales (usually measured with variance).**
- For instance, `mpg` vary between 10 and 34, miles per gallon and car weight between 1.5k and 5.4k lbls.
- It would be hard to compare in their original form.

Centering and scaling in R

```
my_mtcars <- select(mtcars, mpg, wt, qsec, hp)
apply(my_mtcars, 2, mean)
```

```
##       mpg        wt      qsec        hp
##  20.09062   3.21725  17.84875 146.68750
```

```
apply(my_mtcars, 2, sd)
```

```
##        mpg         wt       qsec         hp
##  6.0269481  0.9784574  1.7869432 68.5628685
```

```
my_mtcars_centered = scale(my_mtcars, center = TRUE, scale = FALSE)
apply(my_mtcars_centered, 2, mean)
```

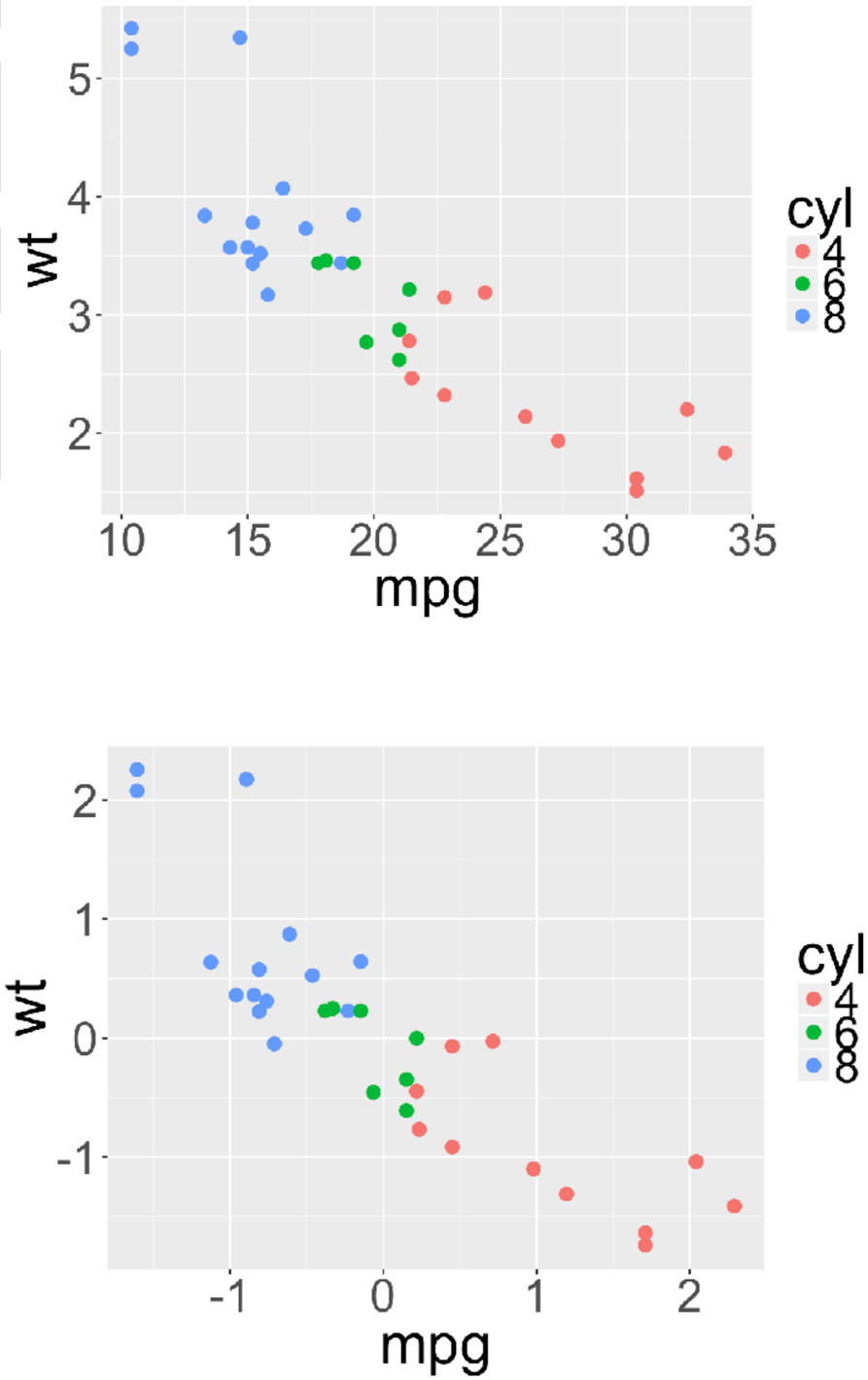
```
##          mpg           wt         qsec           hp
## 4.440892e-16 3.469447e-17 9.436896e-16 0.000000e+00
```

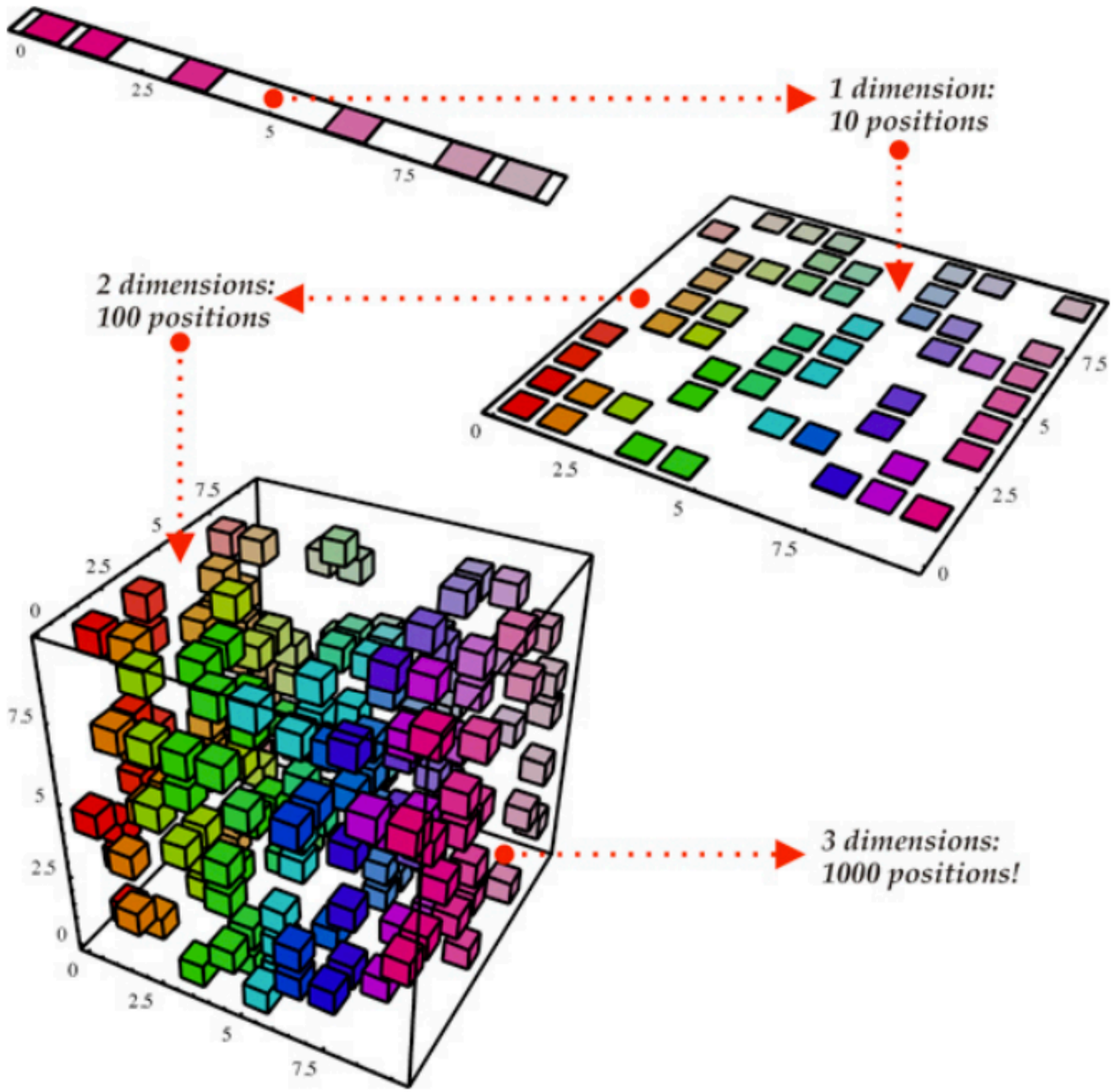
```
apply(my_mtcars_centered, 2, sd)
```

```
##        mpg         wt       qsec         hp
##  6.0269481  0.9784574  1.7869432 68.5628685
```

```
my_mtcars_centered_scaled = scale(my_mtcars)
apply(my_mtcars_centered_scaled, 2, mean)
```

```
##          mpg           wt         qsec           hp
## 7.112366e-17 4.681043e-17 5.299580e-16 1.040834e-17
```





Dimensionality Reduction

Why do you need to reduce dimensions?

- Most of real life **datasets are now high dimensional**
  e.g. genetic sequencing data, medical records data, user internet activity data etc.

Why do you need to reduce dimensions?

- Most of real life **datasets are now high dimensional**
  e.g. genetic sequencing data, medical records data, user internet activity data etc.
- D.R. can serve as a feature extraction method that reduces the number of variables without the loss of signal.
  It can be used to:
  - compress the data
  - remove redundant features and noise
  - increase accuracy of learning methods by avoiding overfitting and the curse of dimensionality

Why do you need to reduce dimensions?

- Most of real life **datasets are now high dimensional**
  e.g. genetic sequencing data, medical records data, user internet activity data etc.
- D.R. can serve as a feature extraction method that reduces the number of variables without the loss of signal.
  It can be used to:
  - compress the data
  - remove redundant features and noise
  - increase accuracy of learning methods by avoiding overfitting and the curse of dimensionality
- Common methods for dimensionality reduction include:
  PCA, CA, ICA, MDS, Isomaps, Laplacian Eigenmaps, tSNE.

A practical guideline to dimensionality reduction

PLOS COMPUTATIONAL BIOLOGY

EDUCATION

Ten quick tips for effective dimensionality reduction

Lan Huong Nguyen[1], **Susan Holmes**[2]*

1 Institute for Mathematical and Computational Engineering, Stanford University, Stanford, California, United States of America, **2** Department of Statistics, Stanford University, Stanford, California, United States of America

* susan@stat.stanford.edu

Learn how to explore your data efficiently using dimensionality reduction.

Make sure you interpret the results of DR correctly!

Lower-Dimensional Projections

- We will use geometrical projections that take **points in higher dimensional spaces and project them down onto lower dimensions**.
- We are not trying to predict one particular variables' value. Instead, we are interested in an **unsupervised learning task** of inferring latent (hidden) patterns and structures "unlabeled" data.
- In particular, we will explain in detail one of techniques called **Principal Component Analysis**.
- PCA is primarily an **exploratory technique that produces maps** that show the relations between variables and between observations in a useful way.

History of PCA

- PCA was invented in **1901 by Karl Pearson** as a way to reduce a two-variable scatterplot to a single coordinate,
- It was again independently developed by Harold Hotelling in the1930s. Statisticians in that period used it to summarize a battery of psychological tests run on the same subjects, by constructing overall scores that summarize many variables at once.
- This idea of principal scores inspired the name Principal Component Analysis. PCA is an unsupervised learning technique; it treats all variables as having the same status, as did clustering.

Karl Pearson

Harold Hotelling

Projecting 2D-data on a line

- In general when reducing/summarizing two dimensional data (a plane) to one (a line) we **lose information.**
- Our goal is to **keep as much information as possible**.
- There are many ways to project a point cloud on a line.

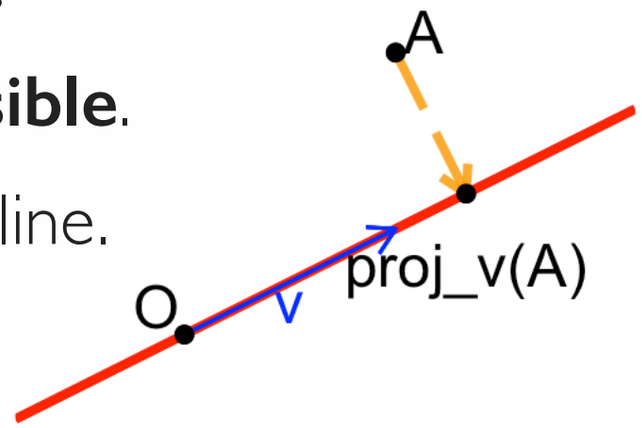


Figure 7.5: Point A is projected onto the red line generated by the vector v. The dashed projection line is perpendicular (or **orthogonal**) to the red line. The intersection point of the projection line and the red line is called the orthogonal projection of A onto the red line generated by the vector v.

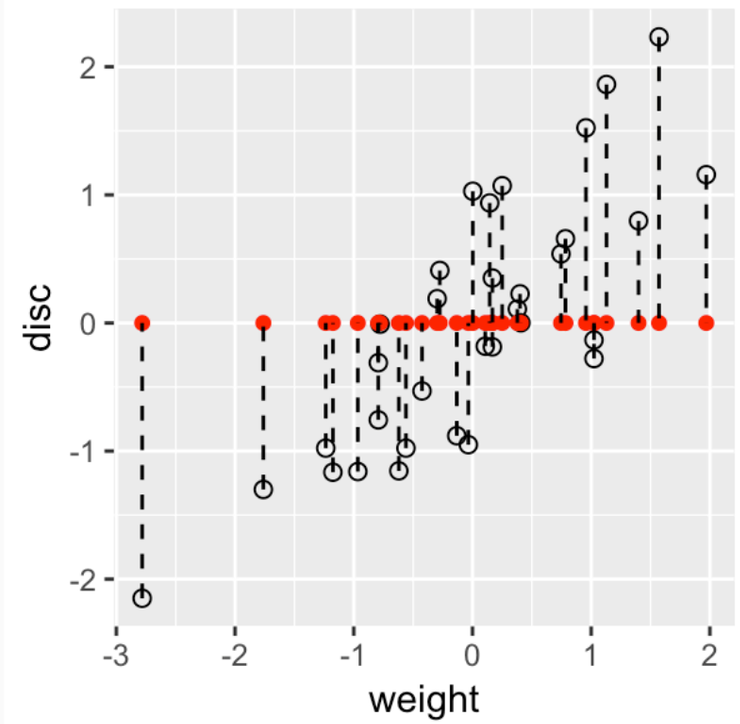


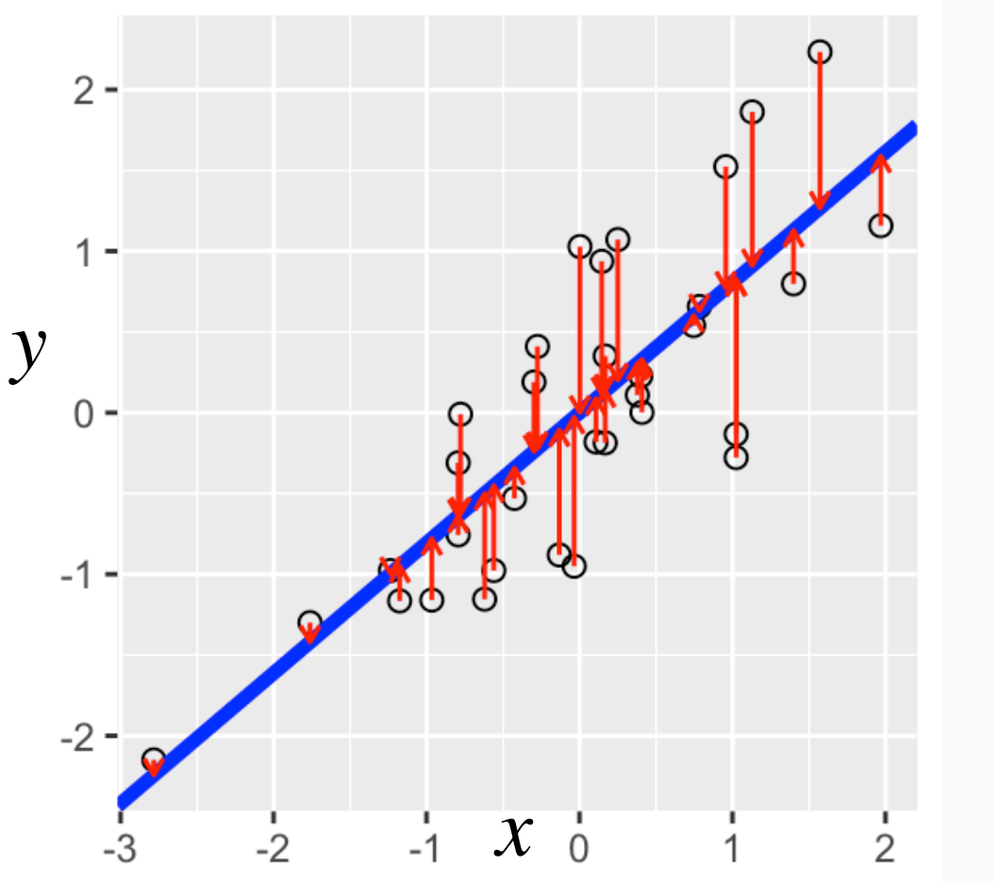
Figure 7.6: Scatterplot of two variables showing the projection on the horizontal x axis (defined by $y = 0$) in red and the lines of projection appear as dashed.

If we do it just by using the original coordinates, for instance the x coordinate as we did on the left, we lose all the information about the second one.

Regression Line

- One of the most widely known method of projecting 2D data on a line is **regression.**
- Regressing y variable on x variable minimizes error in y direction
- **Linear regression** is a **supervised** method that gives preference minimizing the residual sum of squares in one direction (direction of the response variable).

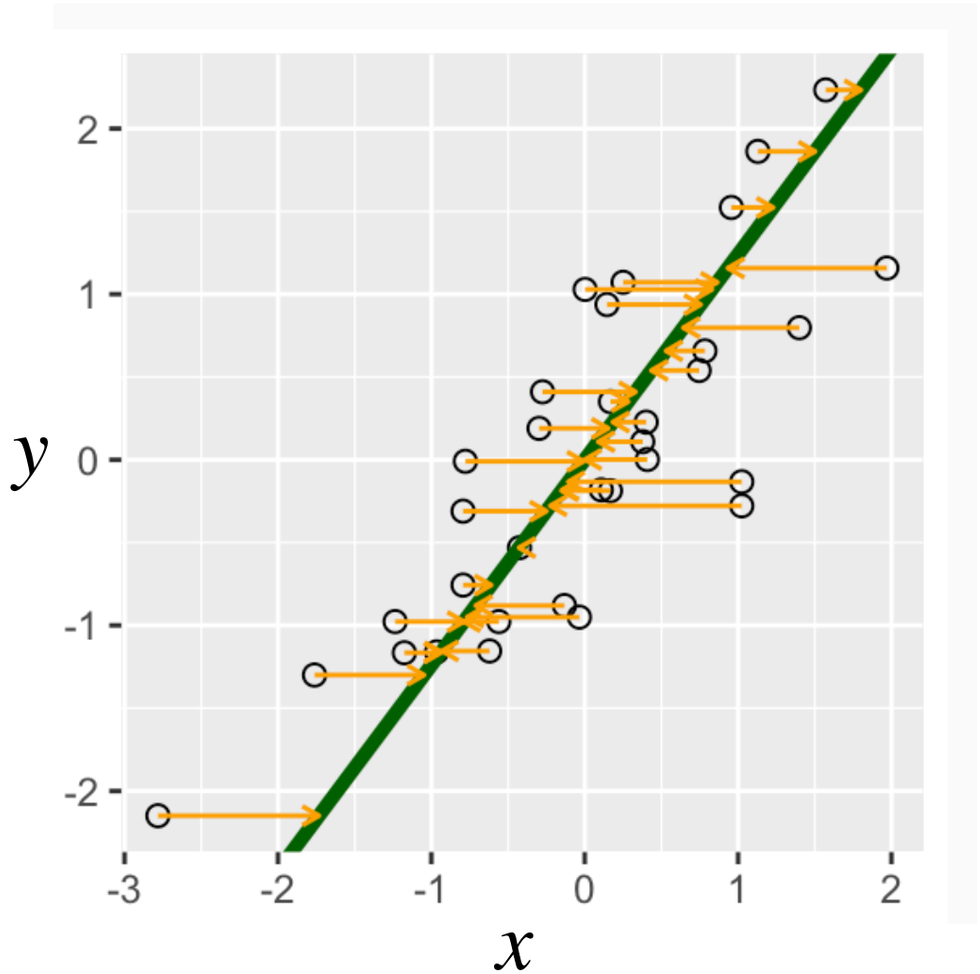
$$y = a + bx$$



Regression Line

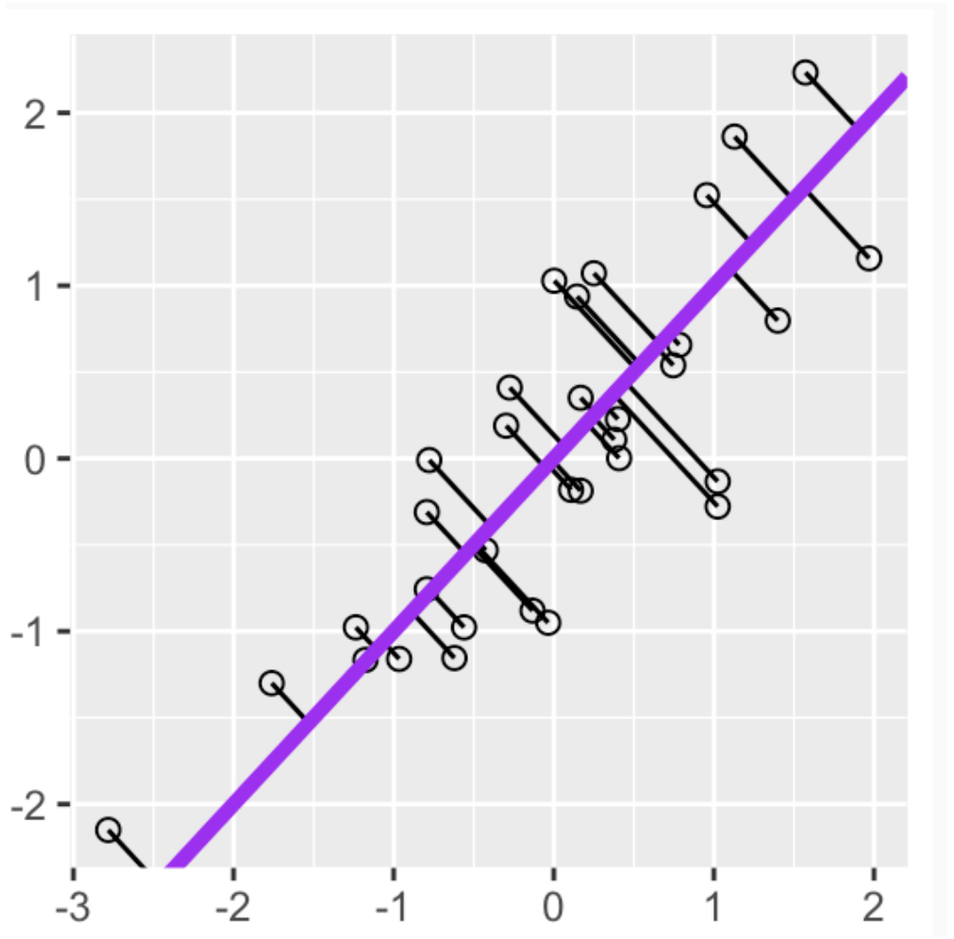
- Of course we can also regress in another direction, i.e. regress x on y:

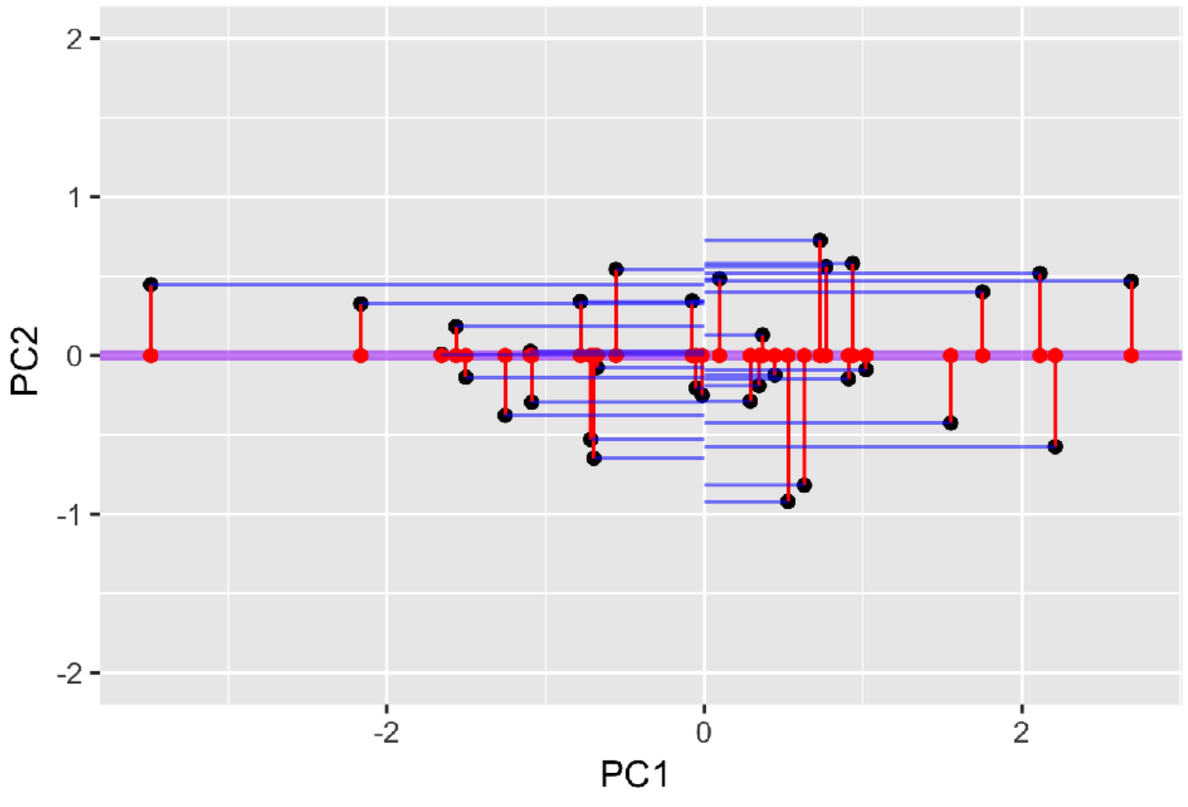
$$x = a + by$$



Principal Component Line

- Line **minimizing error in both horizontal and vertical directions**
- We are in fact minimizing the diagonal projections onto the line from each point





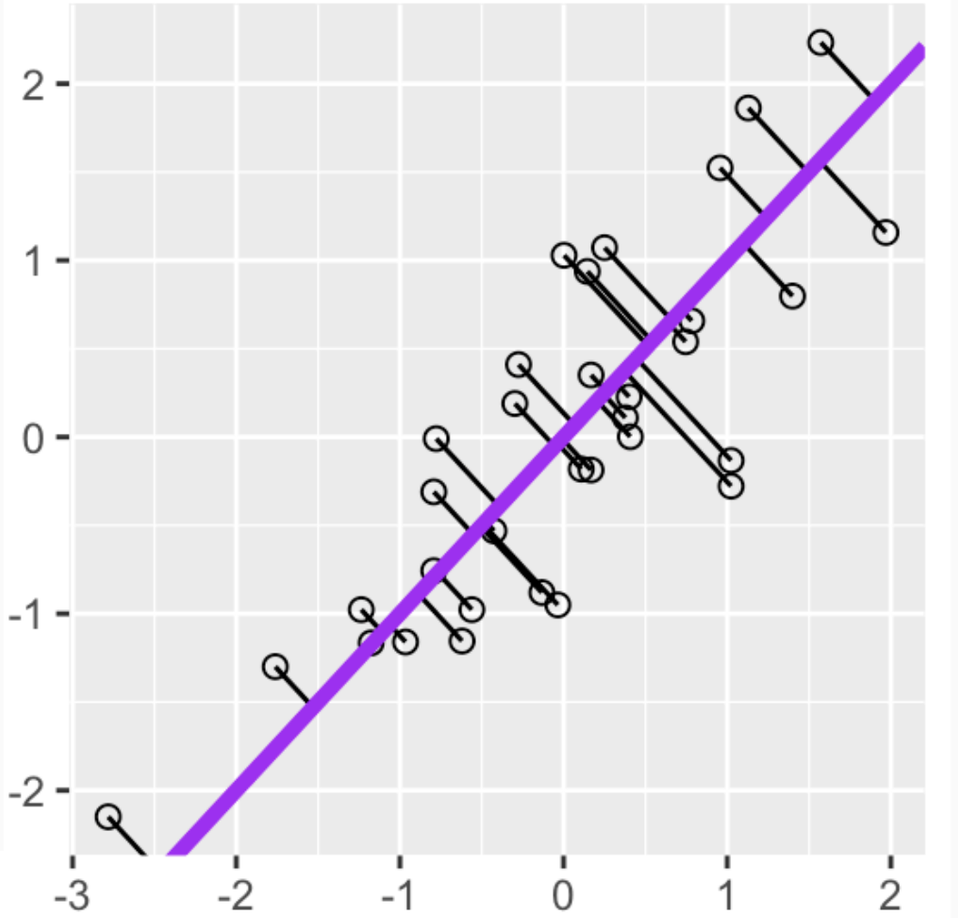
Principal Component Line

► **Question 7.7**

a) What is particular about the slope of the purple line?
b) Redo the plots on the original (unscaled) variables. What happens?

► **Solution**

The lines computed here depend on the choice of units. Because we have made the standard deviations equal to one for both variables, the PCA line is the diagonal that cuts exactly in the middle of both regression lines. Since the data were centered by subtracting their means, the line passes through the origin $(0, 0)$.



Variance along the line

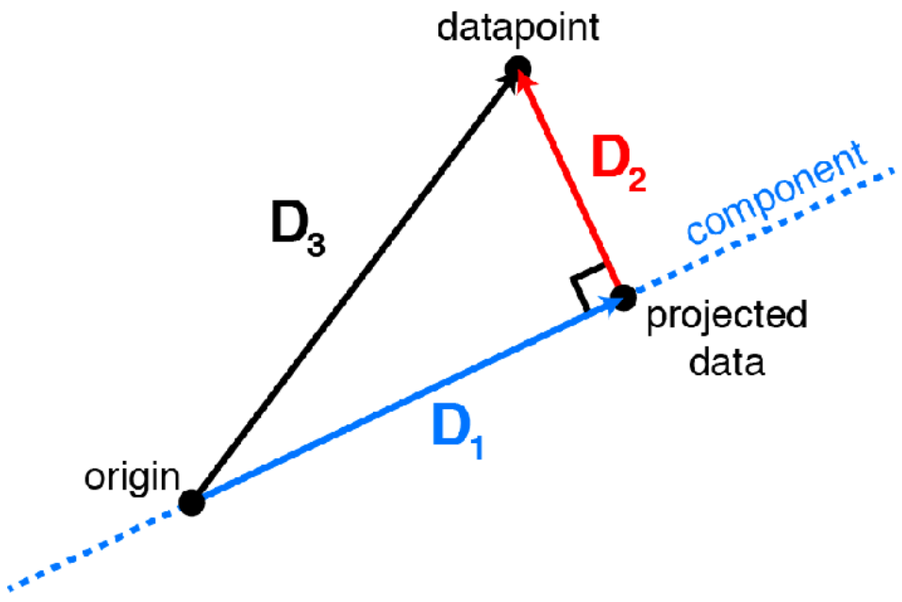
- **Pythagoras' theorem**:
  - The **total variability** of the points is measured by the sum of squares of the distance of the points to the center of gravity, which is the origin (0,0) if the data are centered.

Variance along the line

- **Pythagoras' theorem**:
  - The **total variability** of the points is measured by the sum of squares of the projection of the points onto the center of gravity, which is the origin (0,0) if the data are centered.
  - The total variance is sometimes called **the inertia of the point cloud**. This **inertia can be decomposed into the sum of the squares of the projections onto the line plus the variances along that line**.

Variance along the line

- **Pythagoras' theorem**:
  - The **total variability** of the points is measured by the sum of squares of the distance of the points to the center of gravity, which is the origin (0,0) if the data are centered.
  - The total variance is sometimes called **the inertia of the point cloud**. This **inertia can be decomposed into the sum of the squares of the projections onto the line plus the variances along that line**.
  - For a fixed variance, **minimizing the projection distances also maximizes the variance** along that line. Often we define the first principal component as the line with maximum variance.
- We saw that the **principal component** minimizes the distance to the line, and it **also maximizes the variance of the projections** along the line.



$$D_3^2 = D_1^2 + D_2^2$$

initial variance = remaining variance + lost variance

$$\|a_i\|^2 = \|w_i c\|^2 + \|a_i - w_i c\|^2$$

this is constant — maximize this **or** minimize this

Linear Combinations

- The PC line we found in the previous section could be written

$$PC = \frac{1}{2}x + \frac{1}{2}y$$

- **Principal components are linear combinations of variables that were originally measured,** they provide a new coordinate system.
- This is analogous to what you do when e.g. making a healthy juice mix, you can follow a recipe.

$$V = 2 \times \text{Beets} + 1 \times \text{Carrots} + \frac{1}{2}\,\text{Gala} + \frac{1}{2}\,\text{GrannySmith} + 0.02 \times \text{Ginger} + 0.25\,\text{Lemon}$$

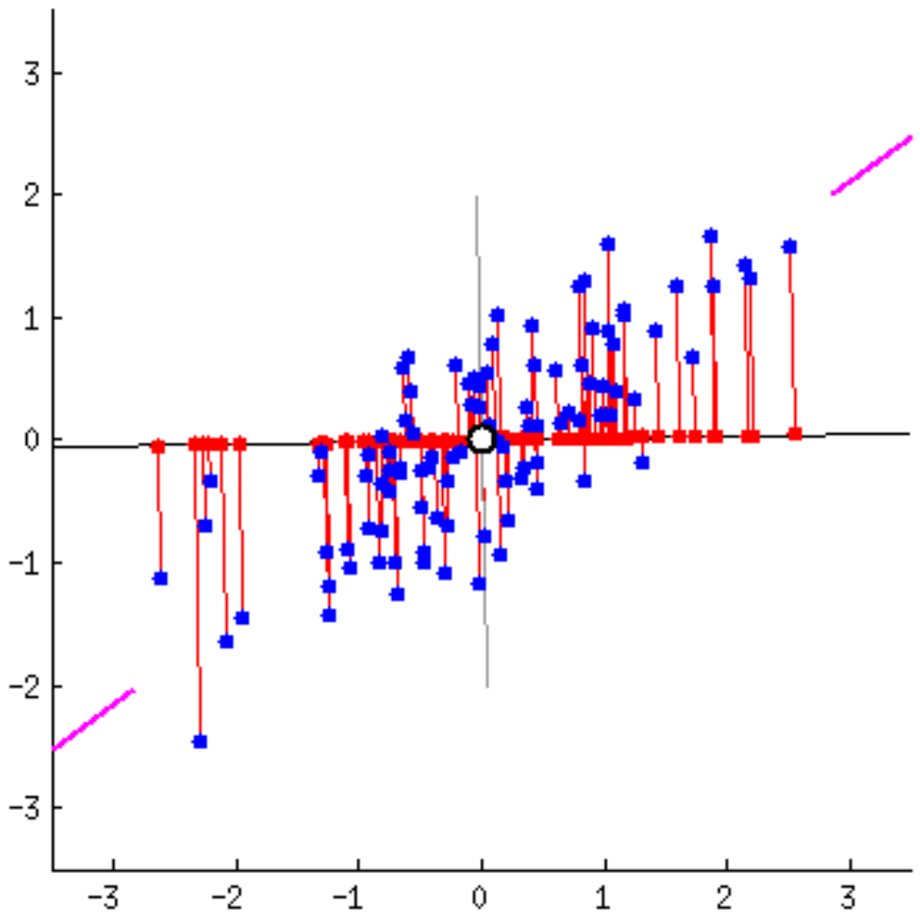
ingredients

- 2 pounds beets (about 6 medium), trimmed, peeled, cut into 1" pieces
- 1 pound carrots (about 4 large), trimmed, peeled, cut into 1" pieces
- 1 Gala or Empire apple (about 8 ounces), cored, cut into 1" pieces
- 1 Granny smith apple (about 8 ounces), cored, cut into 1" pieces
- 1 3" piece fresh ginger, peeled, chopped into 1" pieces
- 3 tablespoons fresh lemon juice

The above recipe is a linear combination of individual ingredients.

The juice mix PC would be a new variable, with coefficients (2,1, 0.5,0.5,0.02,0.25) for each original ingredient, called loadings.

Optimal Line



Source (also includes a great "*dinner-table*" explanation of PCA):
https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

Optimal Line

- In higher dimensions (> 2), **a linear combination of variables also defines a line.**
- As we saw in 2D case, there are many ways to choose lines onto which we project the data, there is however a 'best' line for our purpose.
- In PCA, we use the fact that the total sums of squares of the distances between the points and the origin can be **decomposed into the distance to the line and the variance along the line**.
- We saw that the principal component **minimizes the distance to the line**, and it also **maximizes the variance of the projections** along the line.
- **Why is this a good idea?** Let's look at another example of a projection from 3D to 2D, demonstrating what happens in human vision

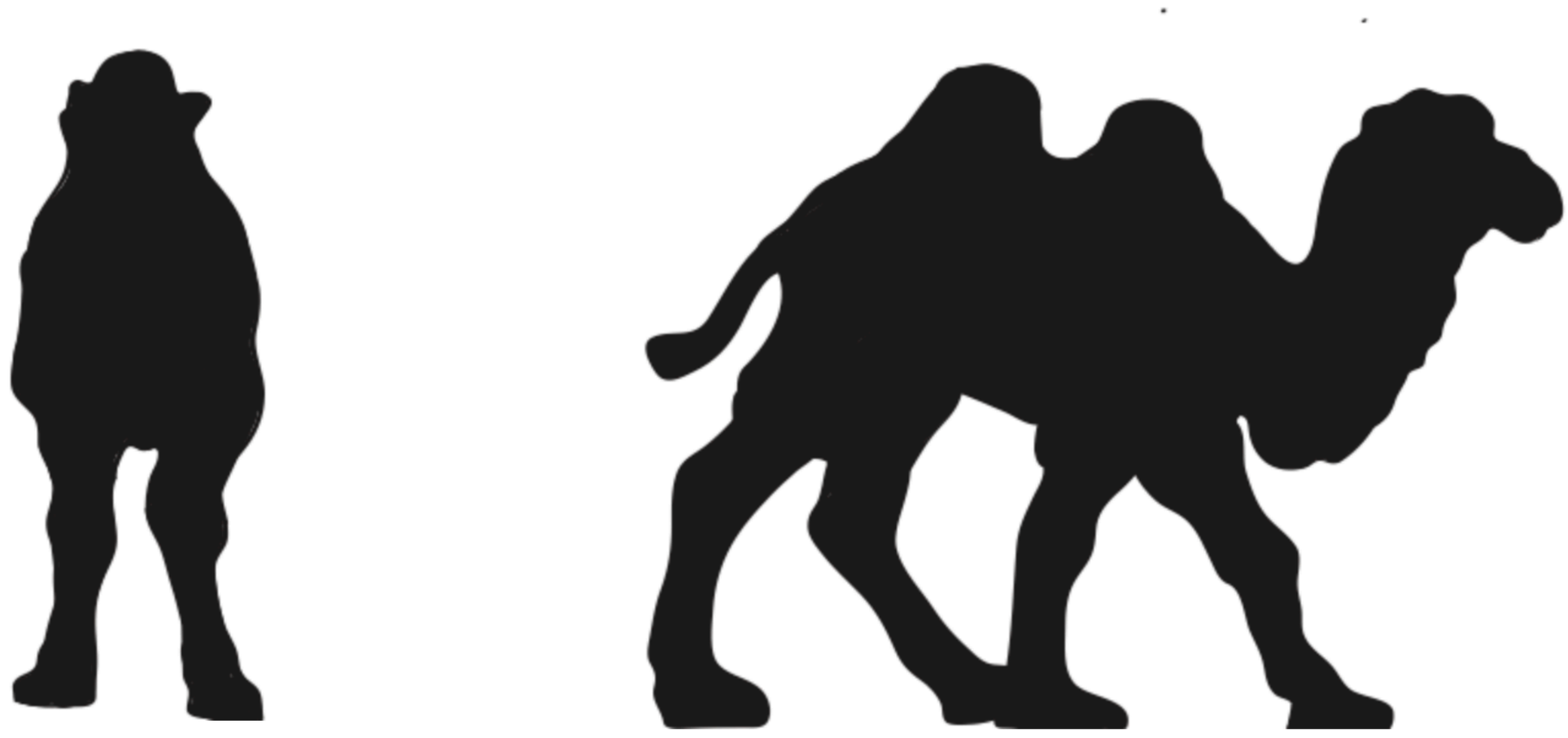
Good Projections

- What is this?

Mystery Image

Good Projections

- What is this?



- Which projection do you think is better?
- It's the **projection that maximizes the area of the shadow** and an equivalent measurement is the sums of squares of the distances between points in the projection, **we want to see as much of the variation as possible, that's what PCA does.**

Principal Component Analysis

- Now, generalized the ideas from before to projection of data with more than 2D.
- PCA is an unsupervised learning technique because it treats all variables as having the same status.
- PCA is visualization technique which produces maps of both variables and observations.
- PCA is a linear technique meaning both that we look for linear relations between variables and that it is based on functions that are linear in the variables

$$f(ax+by)=af(x)+b(y)$$

and thus particularly easy to compute.

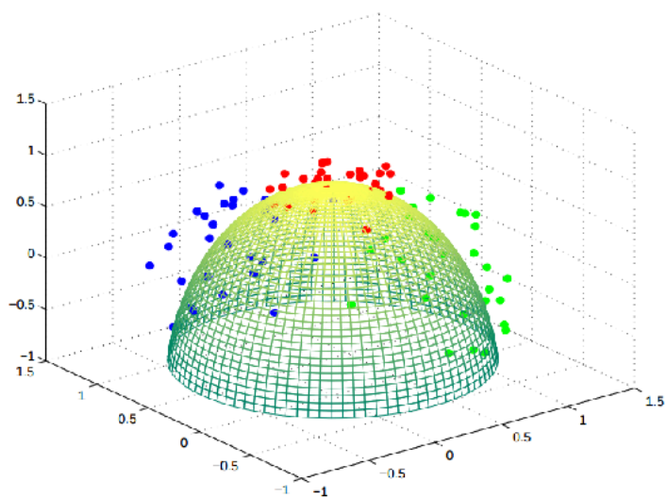


FIGURE 14.15. *Simulated data in three classes, near the surface of a half-sphere.*

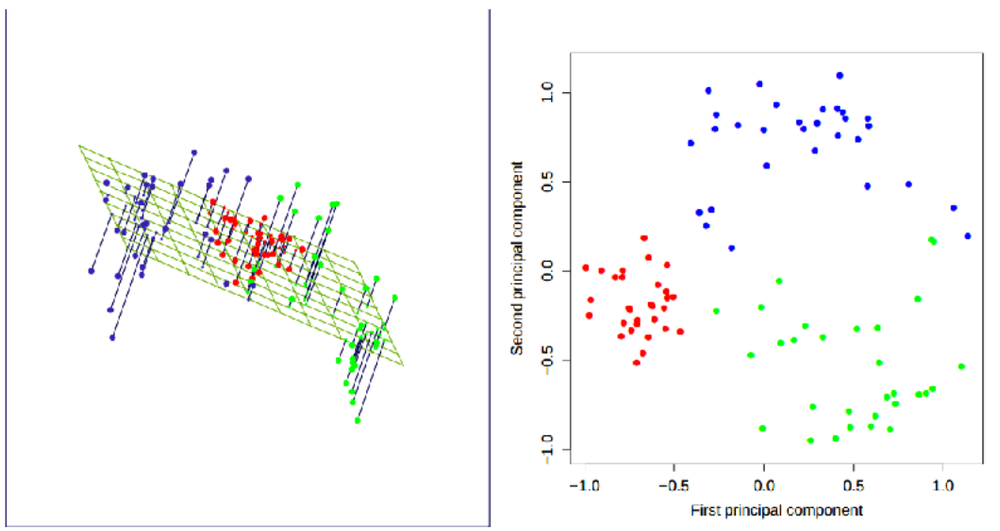


FIGURE 14.21. *The best rank-two linear approximation to the half-sphere data. The right panel shows the projected points with coordinates given by* $\mathbf{U}_2\mathbf{D}_2$*, the first two principal components of the data.*

Principal Component Analysis

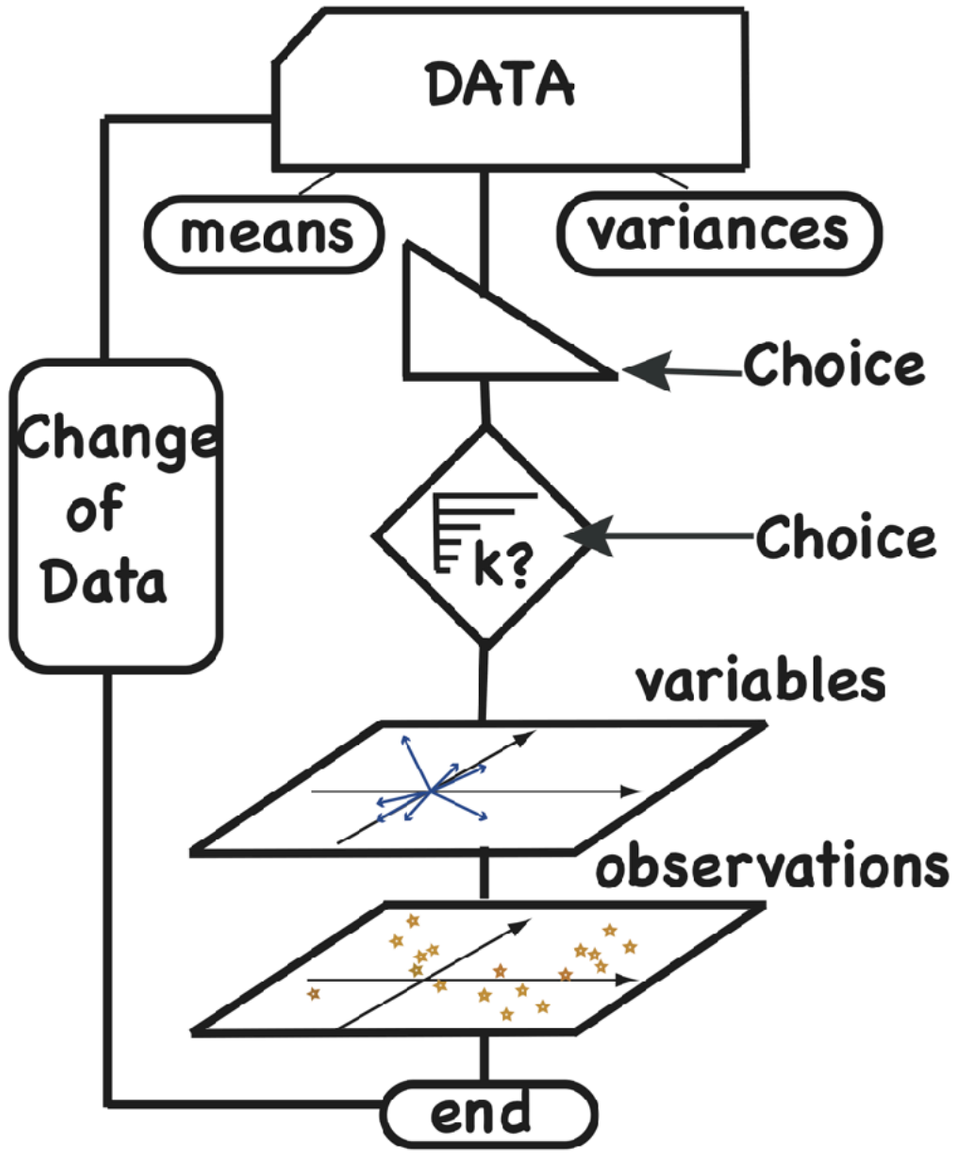
PCA is based on the principle of the following iterative procedure:

1. finding the axis showing the largest variability,
2. removing the variability in that direction,
3. then, iterating to find the next best orthogonal axis so on.

It turns out, we **do not have to run iterations**, all the axes can be found in one operation called the Singular Value Decomposition.

In the diagram, we can see an important step is the choice of k — the number of components relevant to the data.

The number of components k, is also the rank of the data approximation matrix we chose, we will see an explanation of what this means in later slides.



Finding Components Iteratively

- PCA transformation is defined in such a way that the **first PC accounts for as much of the variability in the data as possible**, then each **successive PCs explains the highest remaining variance possible under** the constraint that they are orthogonal to the preceding ones.

Finding Components Iteratively

- PCA transformation is defined in such a way that the first PC accounts for as much of the variability in the data as possible, then each successive PC explains the highest remaining variance possible under the constraint that it is orthogonal to the preceding components.
- Suppose, Σ is an empirical covariance matrix for variables; if the data matrix is already centered: $\Sigma = XX^t/(n-1)$
- The first principal component is defined as:

$$\arg\max_{u \in R^p} \frac{u^t \Sigma u}{u^t u}$$

Finding Components Iteratively

- PCA transformation is defined in such a way that the first PC accounts for as much of the variability in the data as possible, then each successive PC explains the highest remaining variance possible under the constraint that it is orthogonal to the preceding components.
- Suppose, Σ is an empirical covariance matrix for variables; if the data matrix is already centered: $\Sigma = XX^t/(n-1)$
- The first principal component is defined as: $\arg\max_{u \in R^p} \frac{u^t \Sigma u}{u^t u}$
- Then, we subtract the first component approximation from the data:

$$X^{new} = X - Xuu^t$$

- Then, we repeat the process for X^{new} to find subsequent components with a constraint that **new components are orthogonal to all the previous ones.**

Rank of a Matrix

- The rank of a matrix is **the minimum number of orthogonal vectors are needed** to express the columns or rows as their (orthogonal vectors') linear combination.
- It is often the case that datasets are *low-rank* i.e. some of the columns/rows are *redundant* and can be expressed as linear combinations of the remaining ones.
- When a matrix is low-rank or approximately low-rank (with added noise), **dimensionality reduction can be used to reduce redundancy.**
- In PCA, we will see that a the projection of the data is just a linear combination of the original variables.
- One can keep just a subset of k principal components, and obtain a projection which is a **rank-k approximation of the original data.**

Rank-one matrices

- In the textbook you can find an example of a rank-one matrix:

$$X = \begin{pmatrix} 2 & 4 & 8 \\ 4 & 8 & 16 \\ 6 & 12 & 24 \\ 8 & 16 & 32 \end{pmatrix} = u * t(v) = u * v^t \qquad \text{where} \qquad u = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} \text{ and } v = \begin{pmatrix} 2 \\ 4 \\ 8 \end{pmatrix}$$

PCA as Matrix Decomposition

- PCA could be computed using a procedure called :
  Singular Value Decomposition of centered data matrix, $X_{\text{cntr}} = X - \overline{X}$,
  to make things simpler, we assume data was already centered, $X_{\text{cntr}} = X$

- SVD decomposes a matrix into a **product of orthogonal and diagonal matrices**:
  where S is a diagonal matrix with singular values and U, V are orthogonal matrices $(U^t U = I)$, whose columns are left and right singular vectors respectively

$$X = USV^t$$

- The projection of the data to the principal coordinates is called component scores:

$$T = XV = USV^tV = US$$

The component loadings, V, store the coefficients for principal components as the linear combination of the original variables, just as we saw in the case of the principal component line computations.

more details: https://en.wikipedia.org/wiki/Singular-value_decomposition

PCA as Matrix Decomposition

- The same principal components can be obtained by **Eigenvalue Decomposition** of empirical covariance matrix for variables, Σ .
- Eigenvalue decomposition **applies only to square matrices.**
- Recall that $\Sigma = XX^t/(n-1)$ for centered matrices.
- The eigendecomposition is:

$$(n-1)\Sigma = XX^t = Q\Lambda Q^{-1}$$

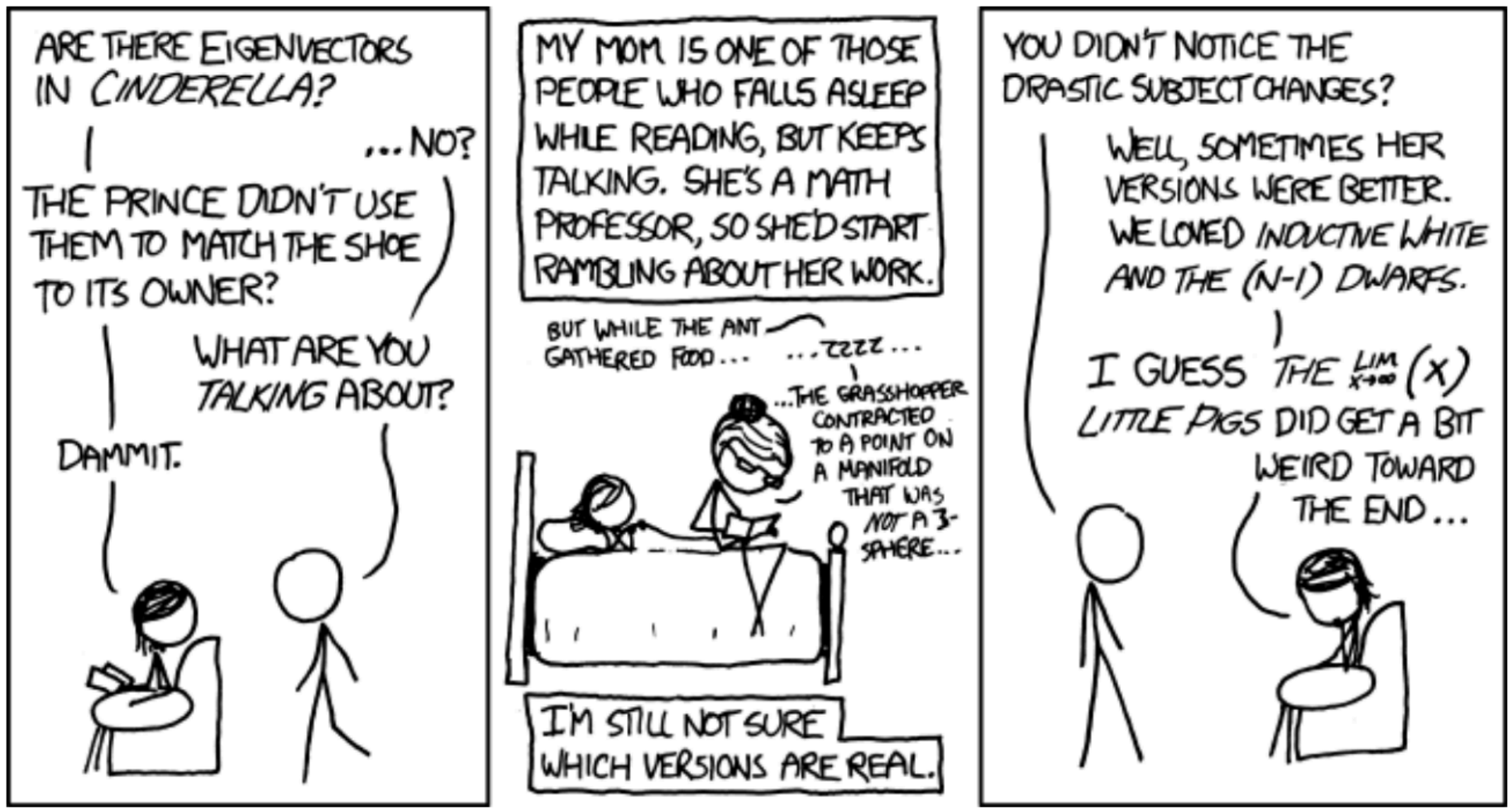
where is Λ a diagonal matrix with **eigenvalues**, and Q is an orthogonal matrix whose columns referred to as **eigenvectors**.

- Again, we have the data projection: $T = XV$
- Also, $Q\Lambda Q^{-1} = XX^t = (USV^t)(VSU^t) = US^2U^t$, so we have:

$$Q = U \quad \text{and} \quad \Lambda = S^2$$

mode details : https://en.wikipedia.org/wiki/Eigenvalue_decomposition

What are eigenvalues and eigenvectors?



$$Av = \lambda v$$

Consider a linear transformation in form of a matrix A, then:

- the **eigenvector**, v, is a vector whose only **magnitude but not direction changes after transformation**,
- and **eigenvalue** is the scalar indicating **the change in magnitude** for the eigenvector after transformation

Why are eigenvalues important?

- Recall from before: $T = XV$ and $X^tX = (VSU^t)(USV^t) = V\Lambda V^t$

- The **covariance matrix for the principal scores is:**

$$T^tT = (V^tX^t)XV = V^t(X^tX)V = V^t(V\Lambda V^t)V = \Lambda$$

a diagonal matrix of eigenvalues,

- which means that **the new (PC) features are uncorrelated and their variances are given by the eigenvalues.**

- Since, eigenvalues are variances of corresponding PCs, they also indicate the **relative importance of the PCs,** which should determine **the aspect ratio of the PCA plots.**

What is the maximum number of PCs?

- Suppose, we have a dates with n samples and p variables, $X \in \mathbb{R}^{n \times p}$.
- The maximum possible number of principal components is less than or equal to the minimum of the number of samples and original variables.

$$K \leq \min(n, p)$$

- Suppose we have 5 samples with 23,000 genes measured on them, what is the dimensionality of these data?

Scree Plot

- Usually, **not all of PCs are informative**, the data matrix can be approximately low-rank, and higher PCs correspond to noise.
- **Choosing the number of PCs to retain is an important** part of a PCA procedure.

Scree Plot

- Usually, not all of PCs are informative, the data matrix can be approximately low-rank, and higher PCs correspond to noise.
- Choosing the number of PCs to retain is an important part of a PCA procedure.
- Unlike clustering, **the number of components should be chosen after completing the analysis.** The choice of k, the number of PCs, for the data projection requires looking at a **screeplot give the magnitude of the eigenvalues**.

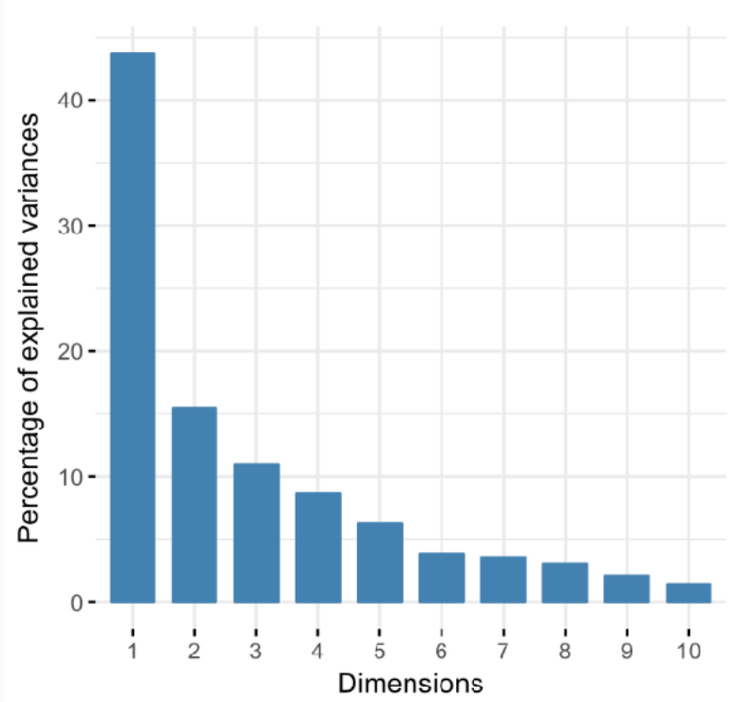


Figure 7.29: Screeplot showing the eigenvalues for the mice data.

Scree Plot

- Usually, not all of PCs are informative, the data matrix can be approximately low-rank, and higher PCs correspond to noise.
- Choosing the number of PCs to retain is an important part of a PCA procedure.
- Unlike clustering, the number of components should be chosen after completing the analysis. The choice of k, the number of PCs, for the data projection requires looking at a **screeplot give the magnitude of the eigenvalues**.
- There are situations when the **PCs are ill-defined: when two or three successive PCs have very similar variances:**

Then, vectors are not meaningful individually and one cannot interpret their loadings. **A very slight change in one observations could give a completely different set of three vectors**.

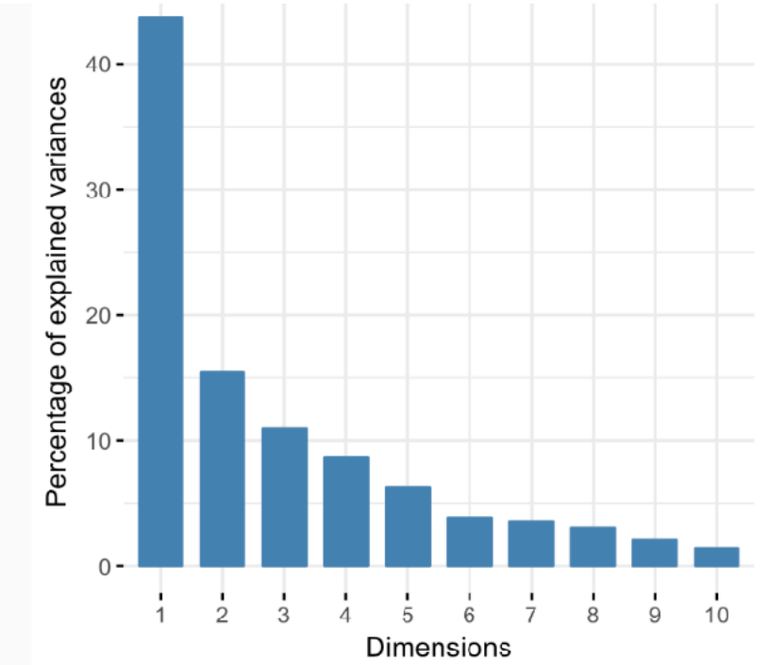


Figure 7.29: Screeplot showing the eigenvalues for the mice data.

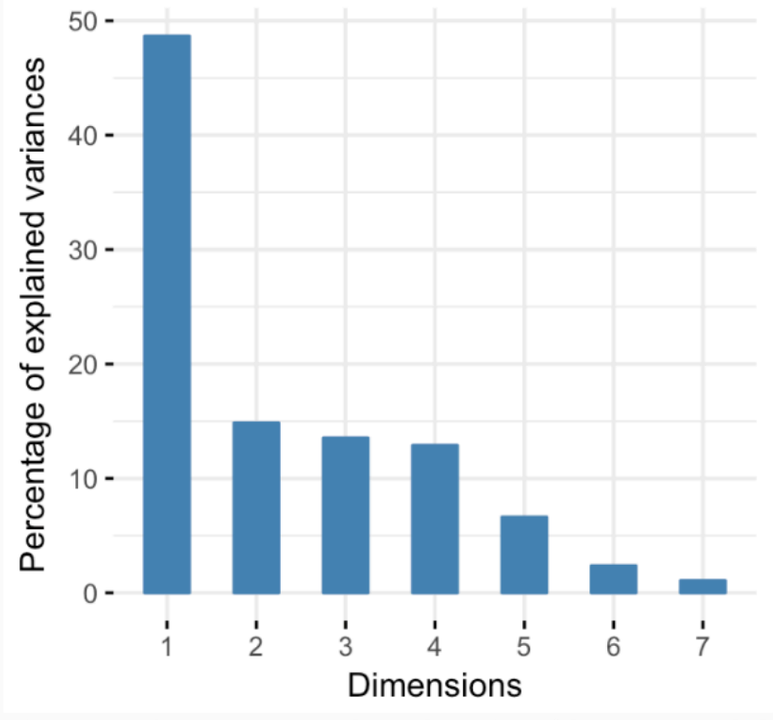
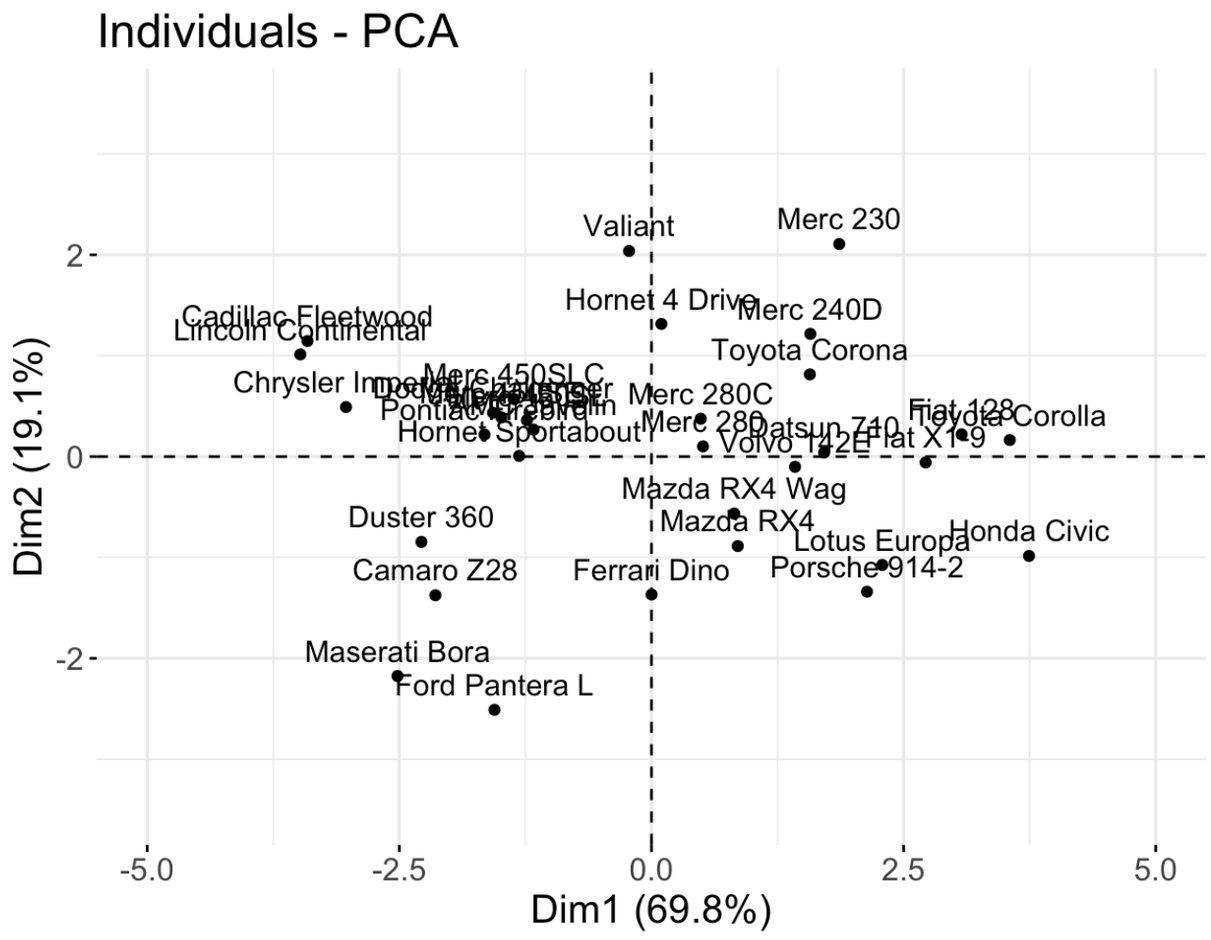


Figure 7.26: A screeplot showing 'dangerously' similar variances. Choosing to cutoff at a hard threshold of 80% of the variance would give unstable PC plots. With so such cutoff, the axes corresponding to the 3D subspace of 3 similar eigenvalues are unstable and cannot be individually interpreted.

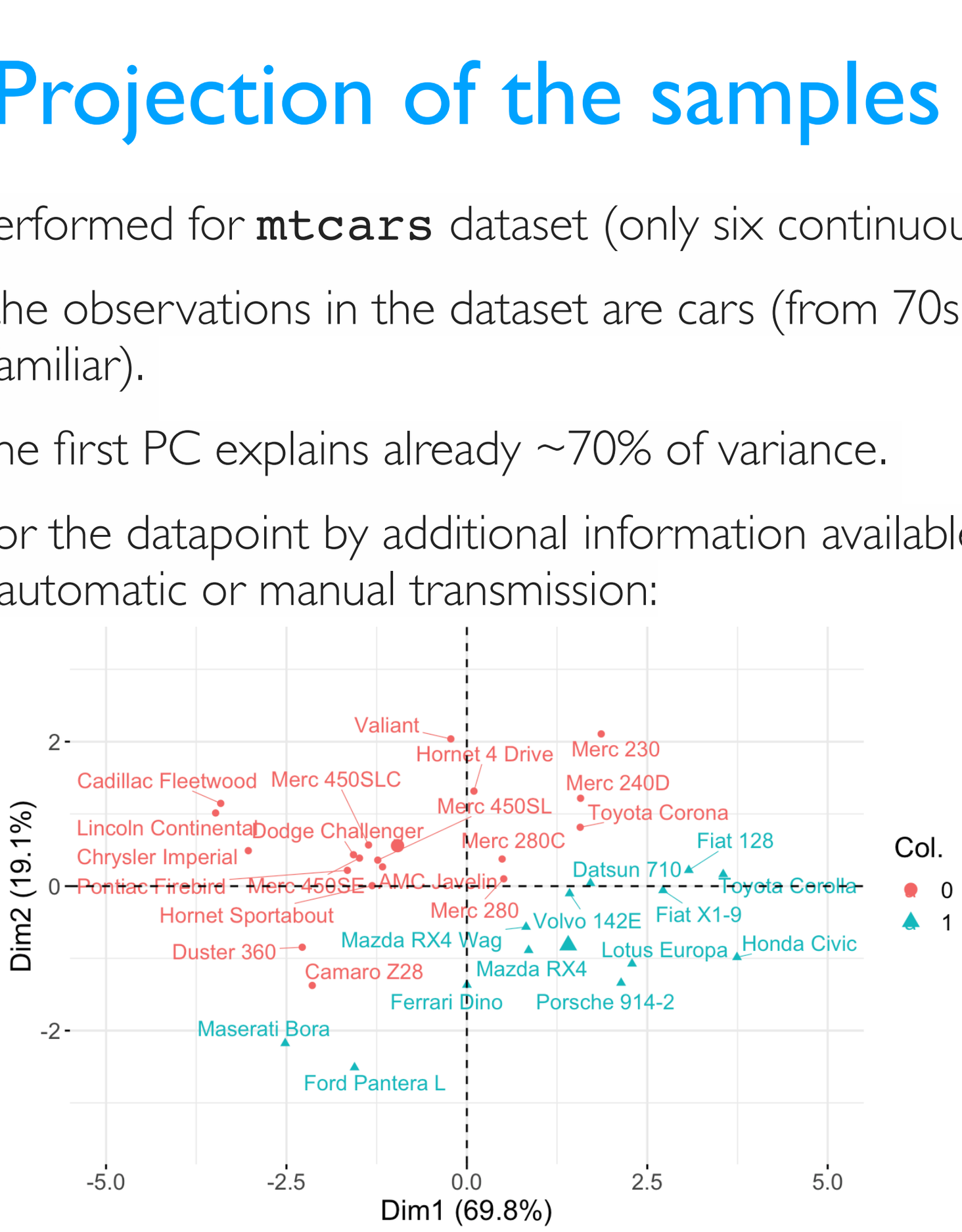
Projection of the samples

- PCA was performed for **`mtcars`** dataset (only six continuous variables).
- Recall that the observations in the dataset are cars (from 70s so they might not sound familiar).
- Note that the first PC explains already ~70% of variance.
- We can color the datapoint by additional information available, e.g. whether the car has automatic or manual transmission:



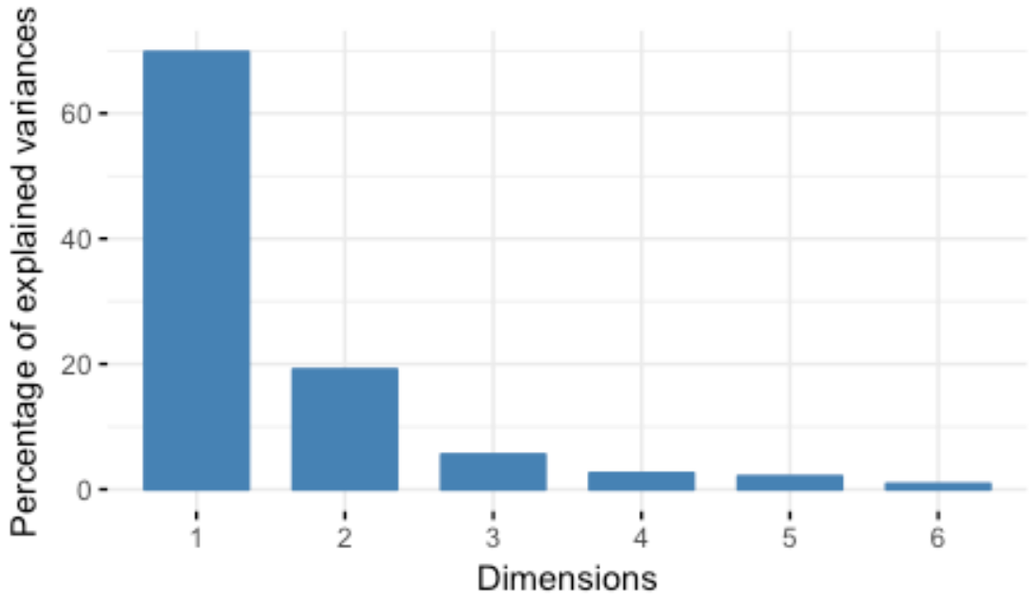
Projection of the samples

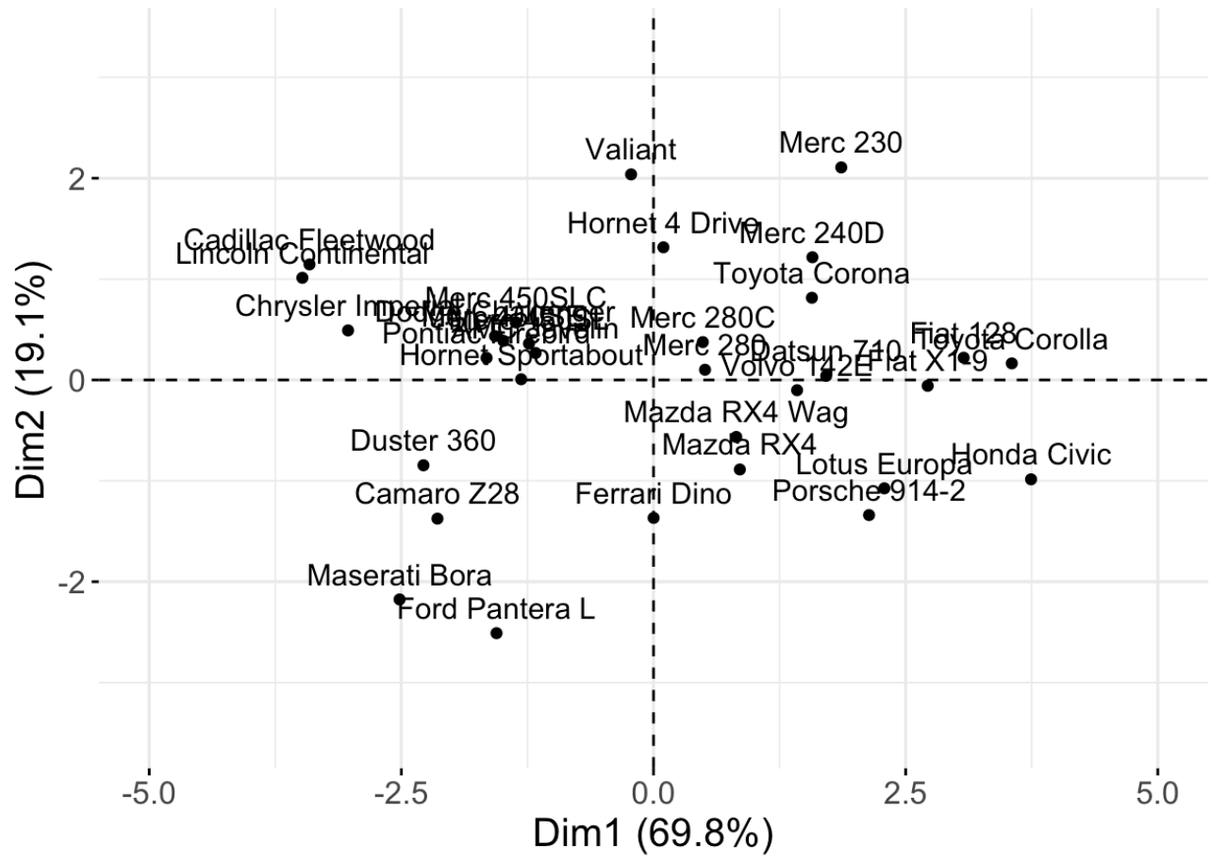
- PCA was performed for **`mtcars`** dataset (only six continuous variables).
- Recall that the observations in the dataset are cars (from 70s so they might not sound familiar).
- Note that the first PC explains already ~70% of variance.
- We can color the datapoint by additional information available, e.g. whether the car has automatic or manual transmission:



Eigenvalues and aspect Ratio

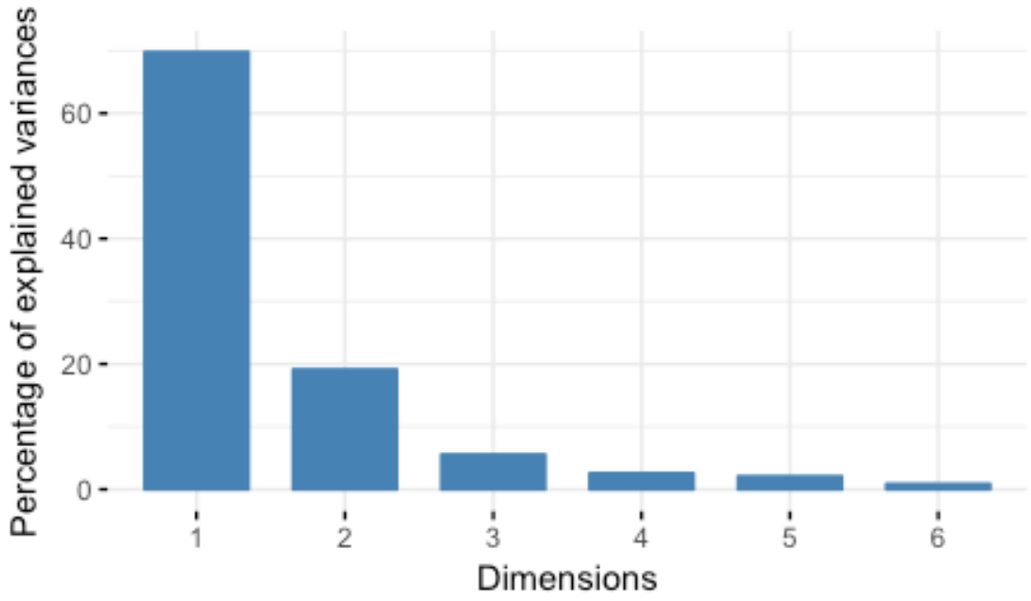
- Remember, that we said **the variance of PCs are given by corresponding eigenvalues.**

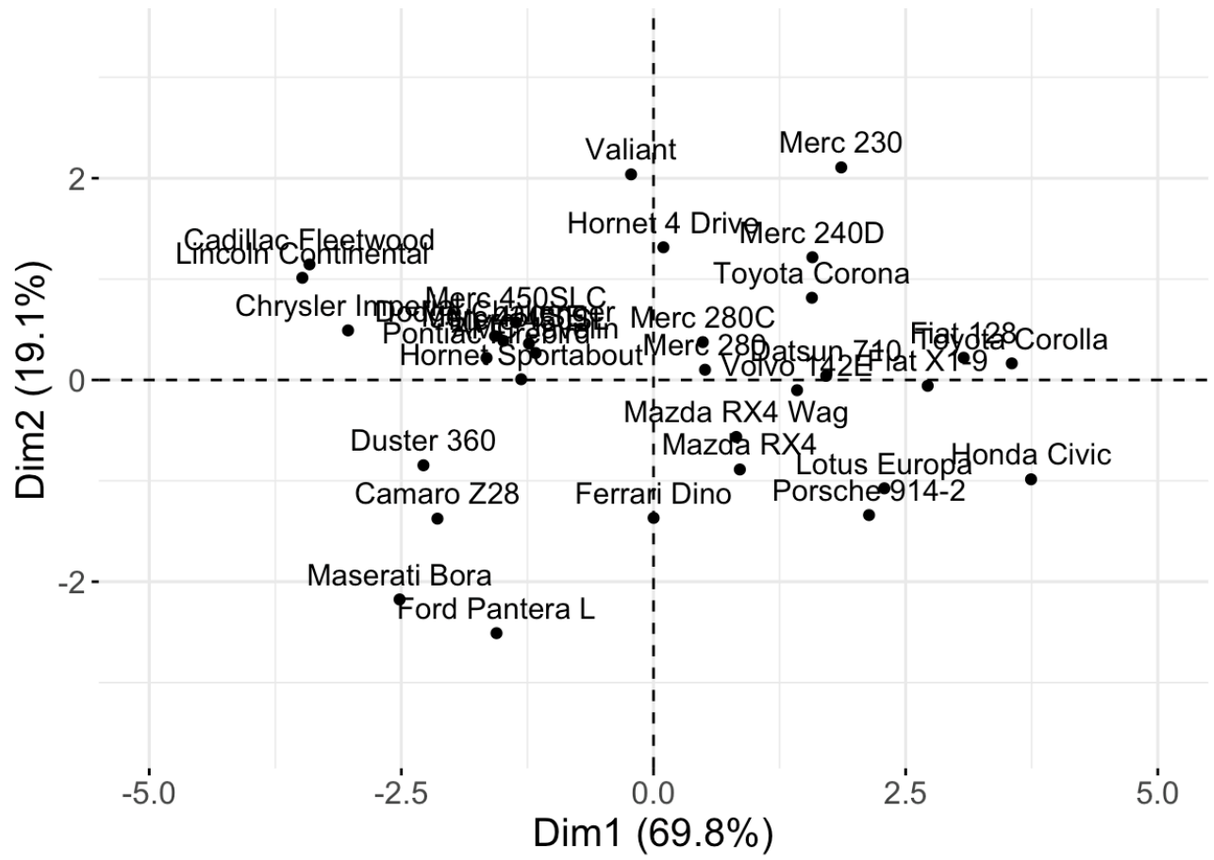




Eigenvalues and aspect Ratio

- Remember, that we said the variance of PCs are given by corresponding eigenvalues.
- Relative eigenvalues (divided by the total sum of all eigenvalues) give **the fraction of variance explained.**

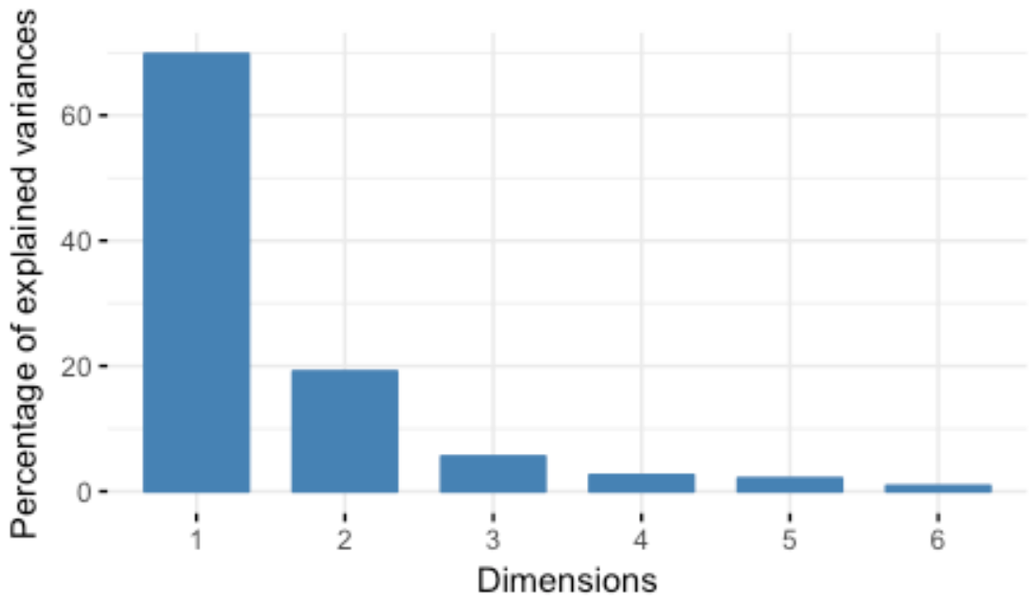


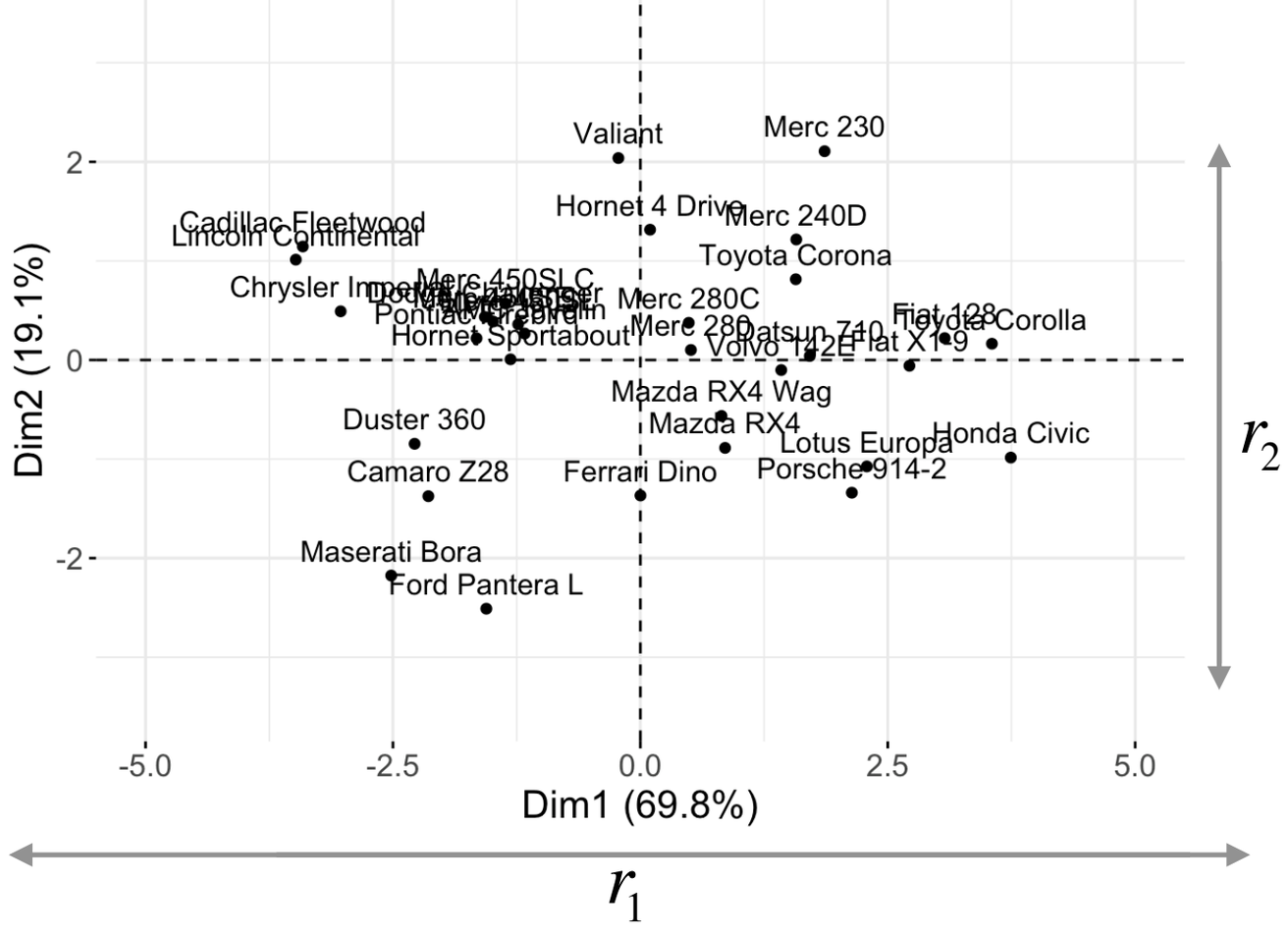


Eigenvalues and aspect Ratio

- Remember, that we said the variance of PCs are given by corresponding eigenvalues.
- Relative eigenvalues (divided by the total sum of all eigenvalues) give the fraction of variance explained.
- Thus, the aspect ratio of the PCA samples projection map should reflect **the ratio between the square root of eigenvalues.**

$$r_1 / r_2 = \sqrt{\lambda_1 / \lambda_2}$$





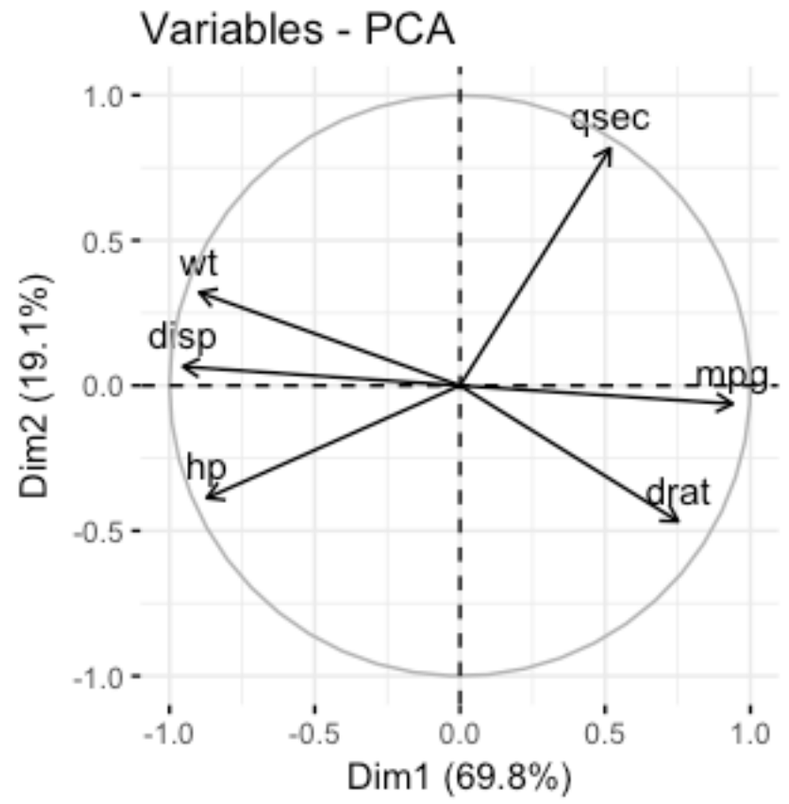
Correlation Circles for variables

- We use six continuous variables from the **`mtcars`** dataset.

mpg	Miles/(US) gallon
disp	Displacement (cu.in.)
hp	Gross horsepower
drat	Rear axle ratio
wt	Weight (1000 lbs)
qsec	first 1/4 mile time

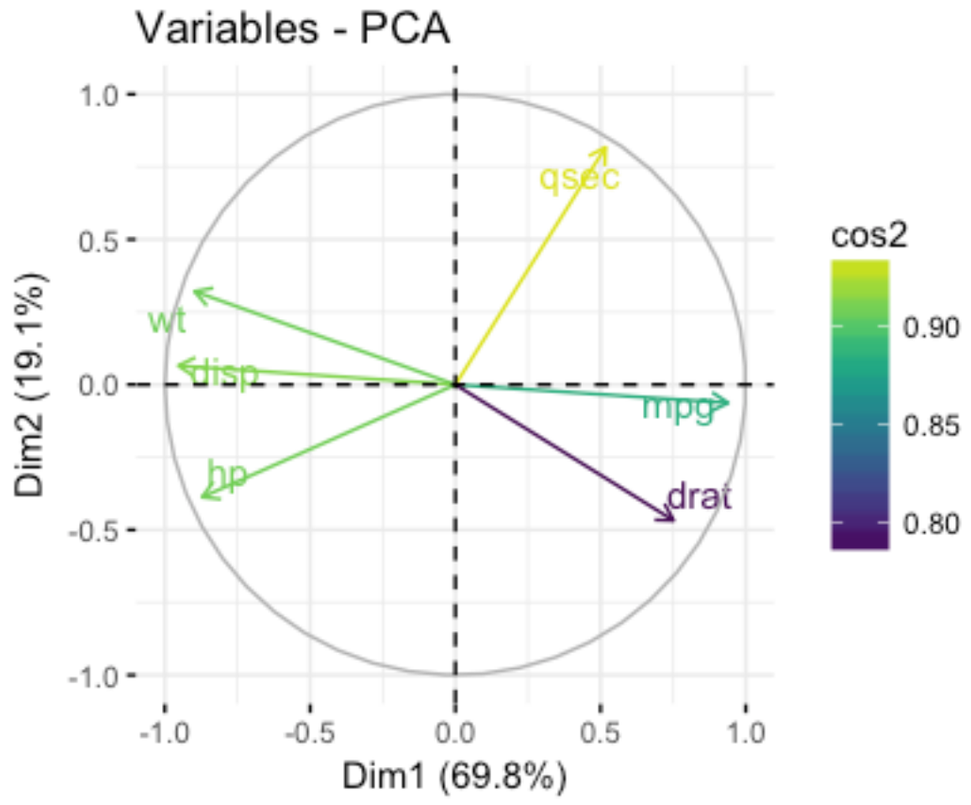
Correlation Circles for variables

- We use six continuous variables from the `mtcars` dataset.
- **Correlation circle** represents original data variables by **their correlations with the principal components.** Note the range of the values on x-y axes.



Correlation Circles for variables

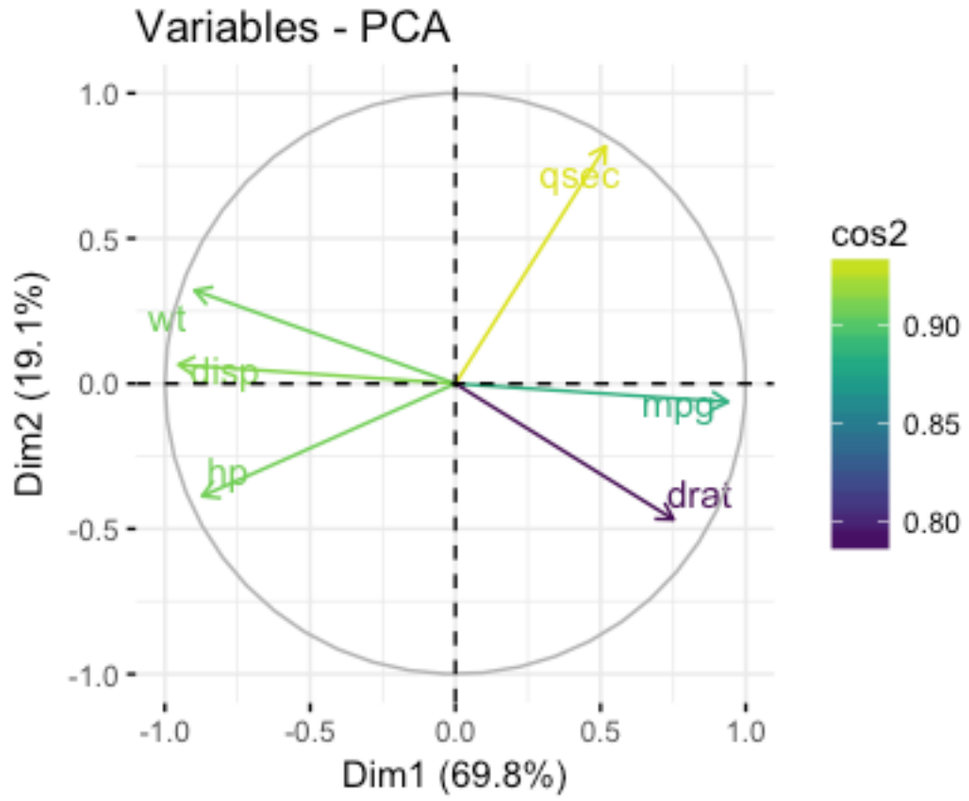
- We use six continuous variables from the **`mtcars`** dataset.
- Correlation circle represents original data variables by their correlations with the principal components. Note the range of the values on x-y axes.
- The **distance between variables and the origin measures the quality of their representation**; ones far away from the origin are well represented.



Correlation Circles for variables

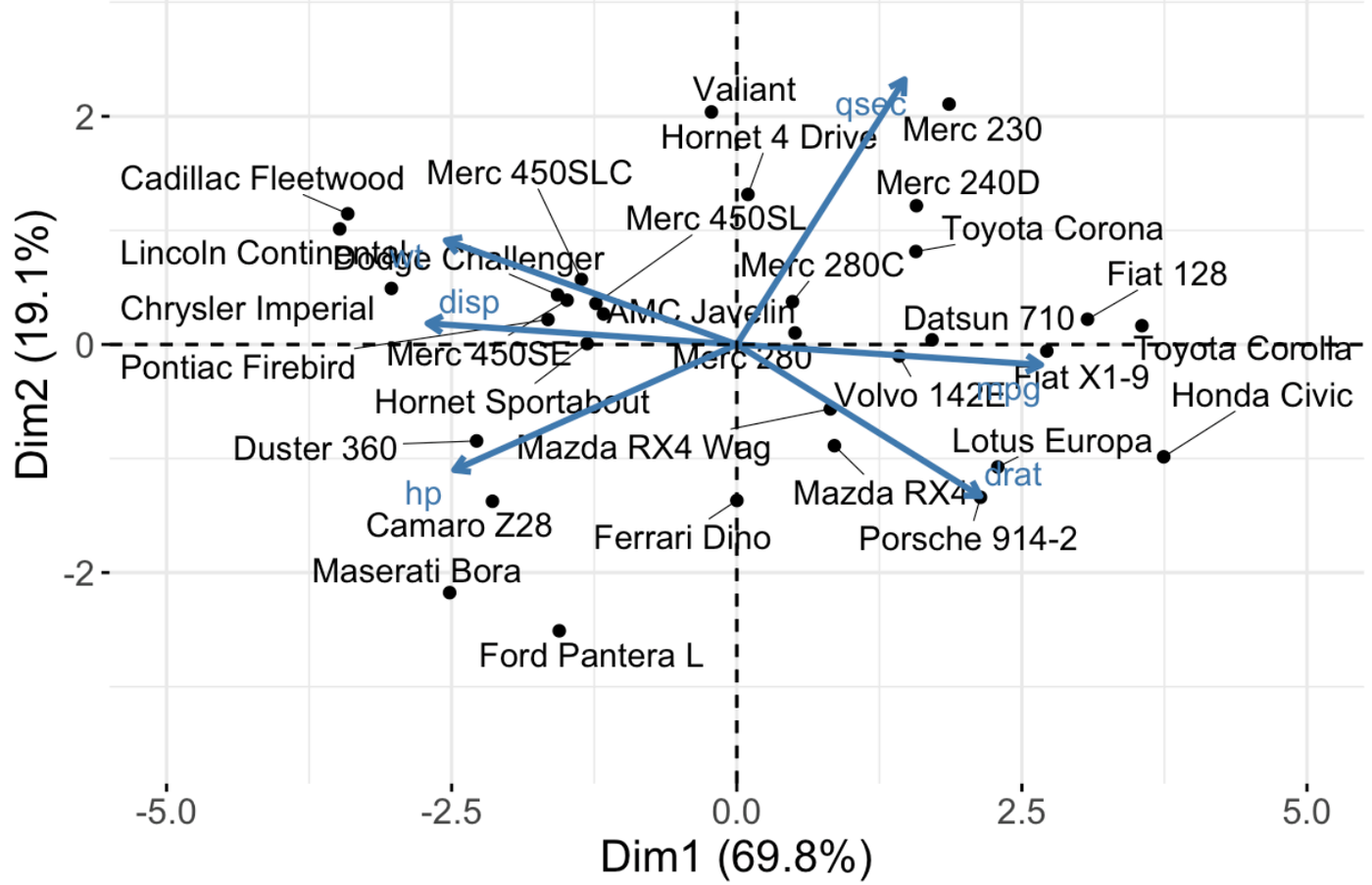
- We use six continuous variables from the **`mtcars`** dataset.
- Correlation circle represents original data variables by their correlations with the principal components. Note the range of the values on x-y axes.
- The distance between variables and the origin measures the quality of their representation; ones far away from the origin are well represented.
- The **angles between vectors are** interpreted as **correlations between** variables. Positively correlated variables are grouped together. Negatively correlated variables are positioned on opposite sides of the plot origin.

	mpg	wt	disp	hp	qsec	drat
mpg	1	-0.9	-0.8	-0.8	0.4	0.7
wt	-0.9	1	0.9	0.7	-0.2	-0.7
disp	-0.8	0.9	1	0.8	-0.4	-0.7
hp	-0.8	0.7	0.8	1	-0.7	-0.4
qsec	0.4	-0.2	-0.4	-0.7	1	0.1
drat	0.7	-0.7	-0.7	-0.4	0.1	1



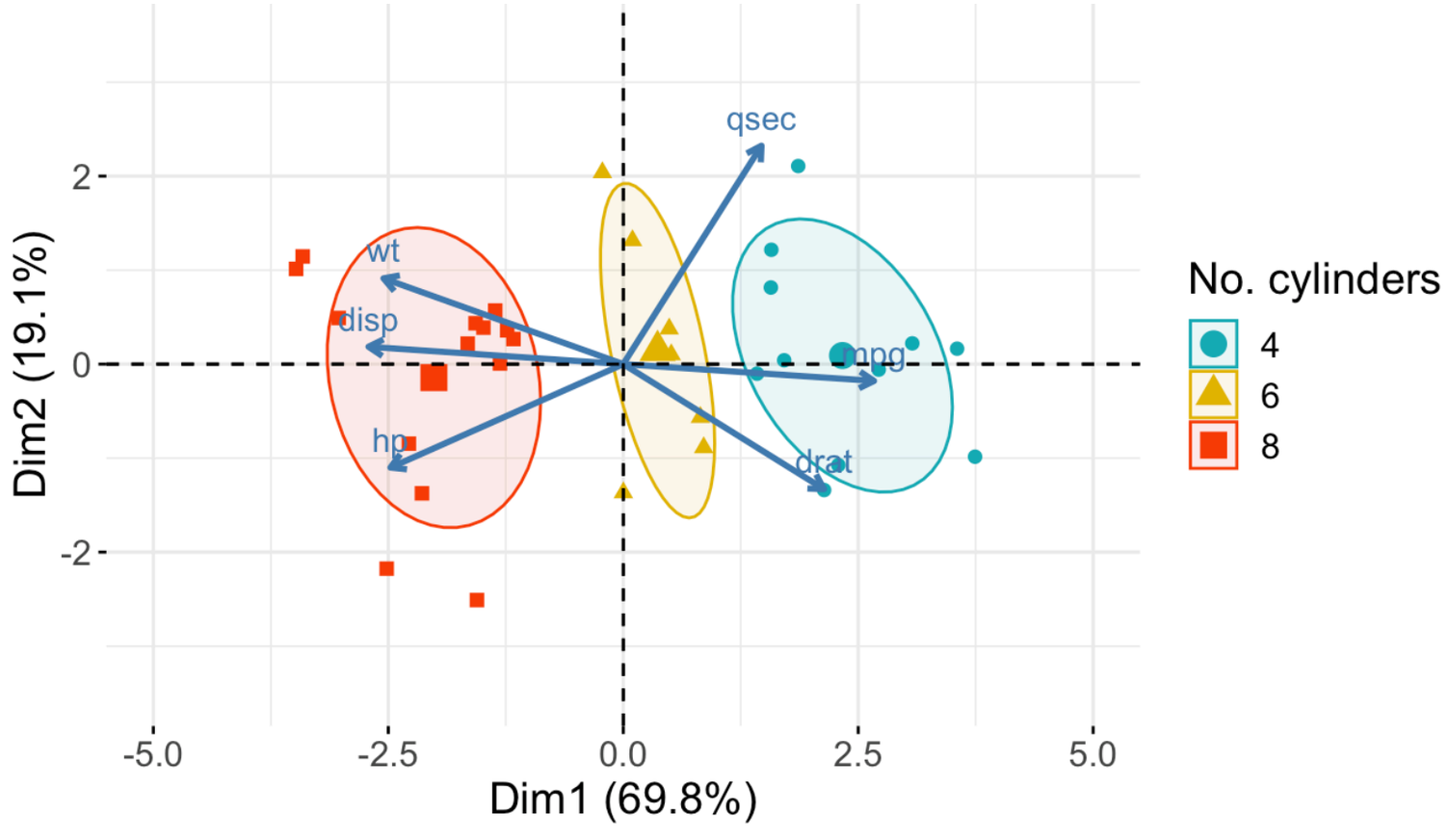
PCA Biplot

- It is often useful to have both the variables and the samples on the same map.
- This simultaneous representation of both the observation and samples is called **a biplot.**
- Note that, the biplot is only useful when there is a **low number of variables and observations** in the data set; otherwise the final plot would be overcrowded.
- The coordinate of individuals and variables are not constructed on the same space. Therefore, in the biplot, you should mainly **focus on the direction of variables but not on their absolute positions.**



PCA Biplot

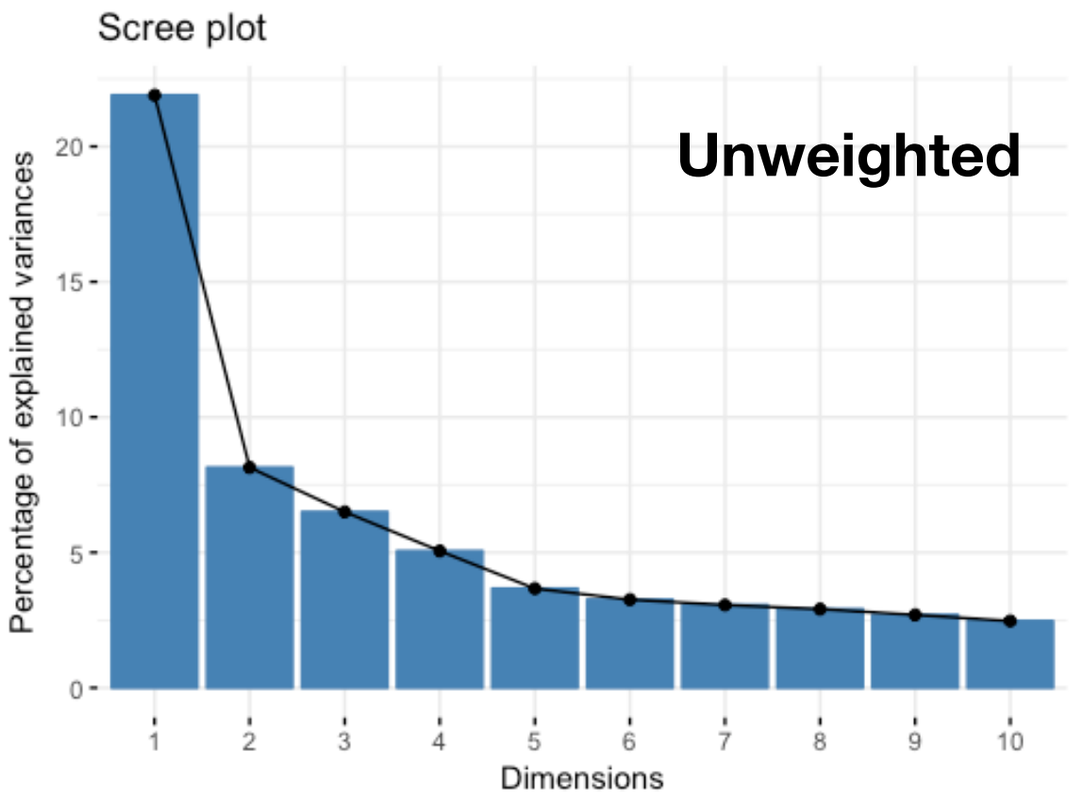
- Roughly speaking a biplot can be interpreted as follows:
  - an sample that is on the same side of a given variable has a high value for this variable,
  - an sample that is on the opposite side of a given variable has a low value for this variable.
- In this example, we can see that the cars with eight cylinders tend to have hight horsepower and weigh a lot, also they have low mileage per hour. Cars with four cylinders are the exact opposite, which makes sense!

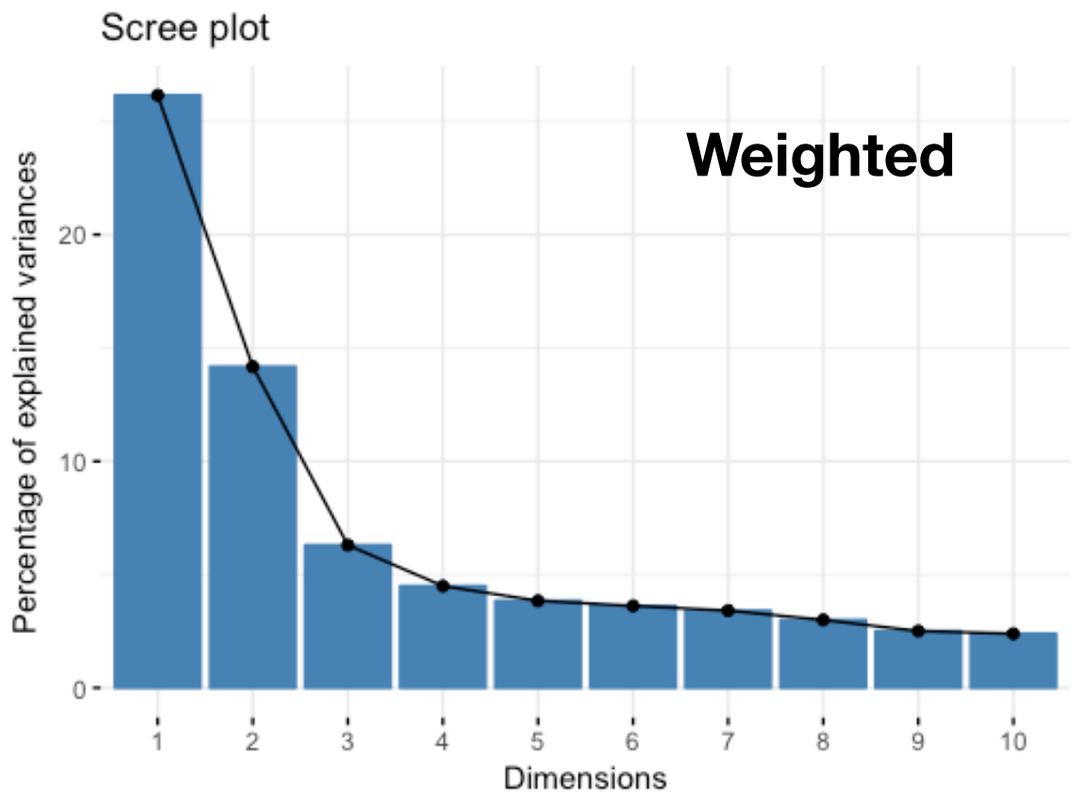


Weighted PCA

- Sometimes we want to see variability between different groups or observations but need to weight them.
- This can happen when heterogeneous groups are imbalanced i.e. they do not contain the same number of observation.
- In these cases, we can **weight the observations by the inverse of the group size.**
- Let's consider an example of a microarray dataset, **`Hiiragi2013`**, from a study by Ohnishi et al. (2014), on early development of mouse embryos (we have worked with it before).
- The data has class imbalance:

E3.25	E3.5 (EPI)	E3.5 (PE)	E4.5 (EPI)	E4.5 (PE)
36	11	11	4	4

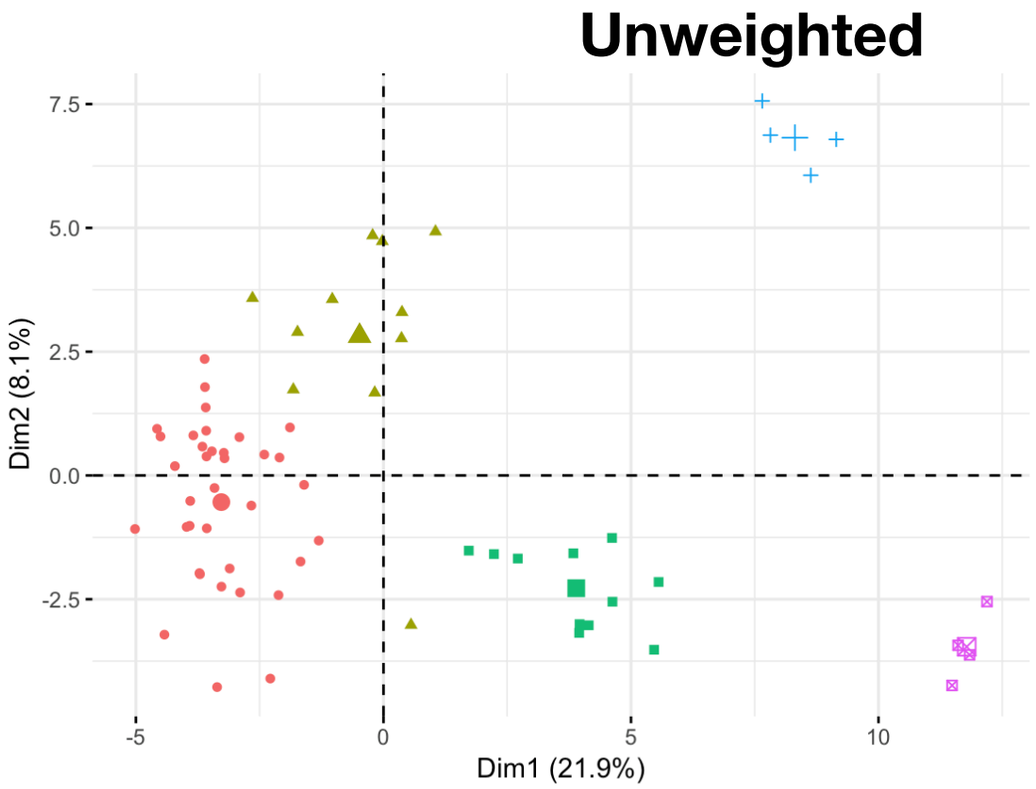


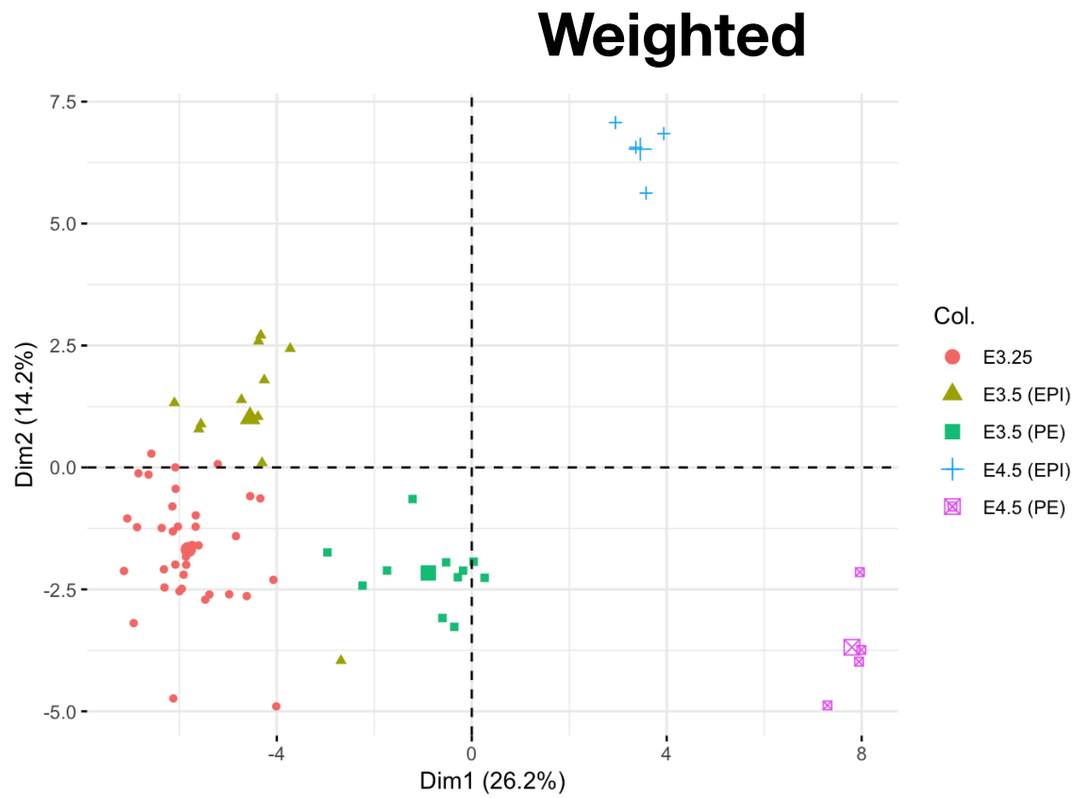


Weighted PCA

- Sometimes we want to see variability between different groups or observations but need to weight them.
- This can happen when heterogeneous groups are imbalanced i.e. they do not contain the same number of observation.
- In these cases, we can **weight the observations by the inverse of the group size.**
- Let's consider an example of a microarray dataset, **`Hiiragi2013`**, from a study by Ohnishi et al. (2014), on early development of mouse embryos (we have worked with it before).
- The data has class imbalance:

E3.25	E3.5 (EPI)	E3.5 (PE)	E4.5 (EPI)	E4.5 (PE)
36	11	11	4	4





Running PCA in R

Wine Dataset

- This dataset contains **chemical measurements on different wines.**
- We also have a information the class of wine each one belongs to, but we will not use it for any computations, just for results interpretation after completing the analysis.
- If you ever want to hear a *dinner table-appropriate* explanation of PCA, I highly recommend looking up this stats-stackexchange post which is also about wines!

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

	Alcohol	MalicAcid	Ash	AlcAsh	Mg	Phenols	Flav	NonFlav Phenols	Proa	Color	Hue	OD	Proline
1	14.23	1.71	2.43	15.6	127	2.8	3.06	0.28	2.29	5.64	1.04	3.92	1065
2	13.2	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.4	1050
3	13.16	2.36	2.67	18.6	101	2.8	3.24	0.3	2.81	5.68	1.03	3.17	1185
4	14.37	1.95	2.5	16.8	113	3.85	3.49	0.24	2.18	7.8	0.86	3.45	1480
5	13.24	2.59	2.87	21	118	2.8	2.69	0.39	1.82	4.32	1.04	2.93	735
6	14.2	1.76	2.45	15.2	112	3.27	3.39	0.34	1.97	6.75	1.05	2.85	1450

Center and Scale the Data

```
load("wine.RData")
apply(wine, 2, mean)
```

```
##        Alcohol      MalicAcid            Ash         AlcAsh             Mg
##     13.0006180      2.3363483      2.3665169     19.4949438     99.7415730
##        Phenols           Flav NonFlavPhenols           Proa          Color
##      2.2951124      2.0292697      0.3618539      1.5908989      5.0580899
##            Hue             OD        Proline
##      0.9574494      2.6116854    746.8932584
```

```
apply(wine, 2, sd)
```

```
##        Alcohol      MalicAcid            Ash         AlcAsh             Mg
##      0.8118265      1.1171461      0.2743440      3.3395638     14.2824835
##        Phenols           Flav NonFlavPhenols           Proa          Color
##      0.6258510      0.9988587      0.1244533      0.5723589      2.3182859
##            Hue             OD        Proline
##      0.2285716      0.7099904    314.9074743
```

```
wine_scaled = scale(wine)
apply(wine_scaled, 2, mean)
```

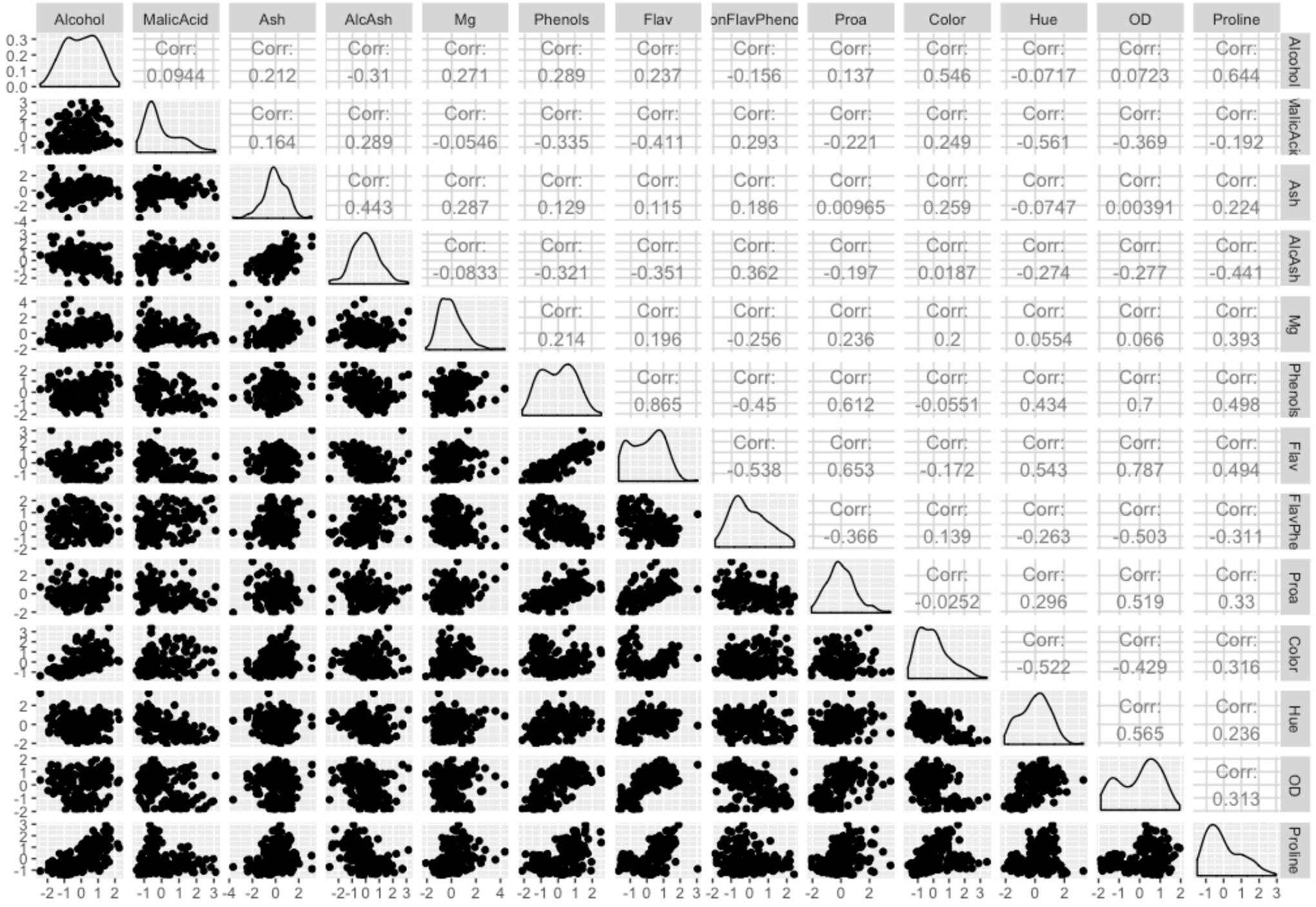
```
##        Alcohol      MalicAcid            Ash         AlcAsh             Mg
##  -8.591766e-16  -6.776446e-17   8.045176e-16  -7.720494e-17  -4.073935e-17
##        Phenols           Flav NonFlavPhenols           Proa          Color
##  -1.395560e-17   6.958263e-17  -1.042186e-16  -1.221369e-16   3.649376e-17
##            Hue             OD        Proline
##   2.093741e-16   3.003459e-16  -1.034429e-16
```

```
apply(wine_scaled, 2, sd)
```

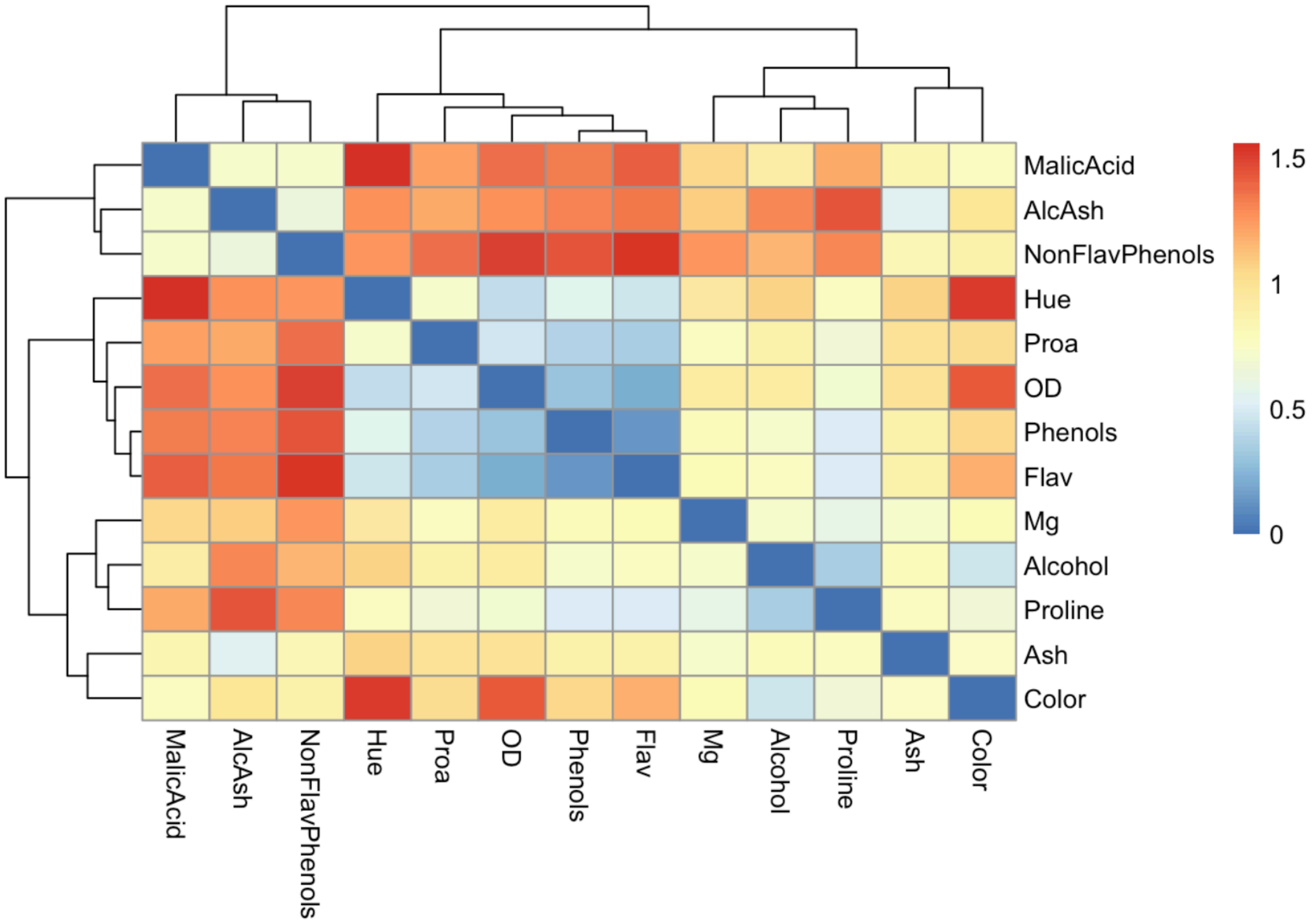
```
##        Alcohol      MalicAcid            Ash         AlcAsh             Mg
##              1              1              1              1              1
##        Phenols           Flav NonFlavPhenols           Proa          Color
##              1              1              1              1              1
##            Hue             OD        Proline
##              1              1              1
```

2D summaries

```
wine_scaled <- as.data.frame(wine_scaled)
GGally::ggpairs(wine_scaled)
```



```r
library(pheatmap)
pheatmap(1 - cor(wine))
```



Use R functions to compute PCA

- To compute PCA in R, you can use any of the following functions: **`princomp, prcomp, ade4::dudi.pca`** or even **`svd`**
- Here we will use **`dudi.pca`** from **`ade4`** package, for others see the textbook.
- Note that, there is no need to center and scale the data ahead of time!

```
library(ade4)
winePCA = dudi.pca(wine, nf = 5, scale = TRUE, center = TRUE, scannf=FALSE)
```

```
class(winePCA)
```

```
## [1] "pca"  "dudi"
```

```
names(winePCA)
```

```
##  [1] "tab"  "cw"   "lw"   "eig"  "rank" "nf"   "c1"   "li"   "co"   "l1"
## [11] "call" "cent" "norm"
```

- Note that we set nf = 5, that is we only retain 5 PCs. The rest will not be included in the result.
- The output of **`dudi.pca()`** is of class both "pca" and "dudi", but is basically a list containing many elements
- scan = FALSE surpasses automatic printing of the the screeplot

Elements of the Output

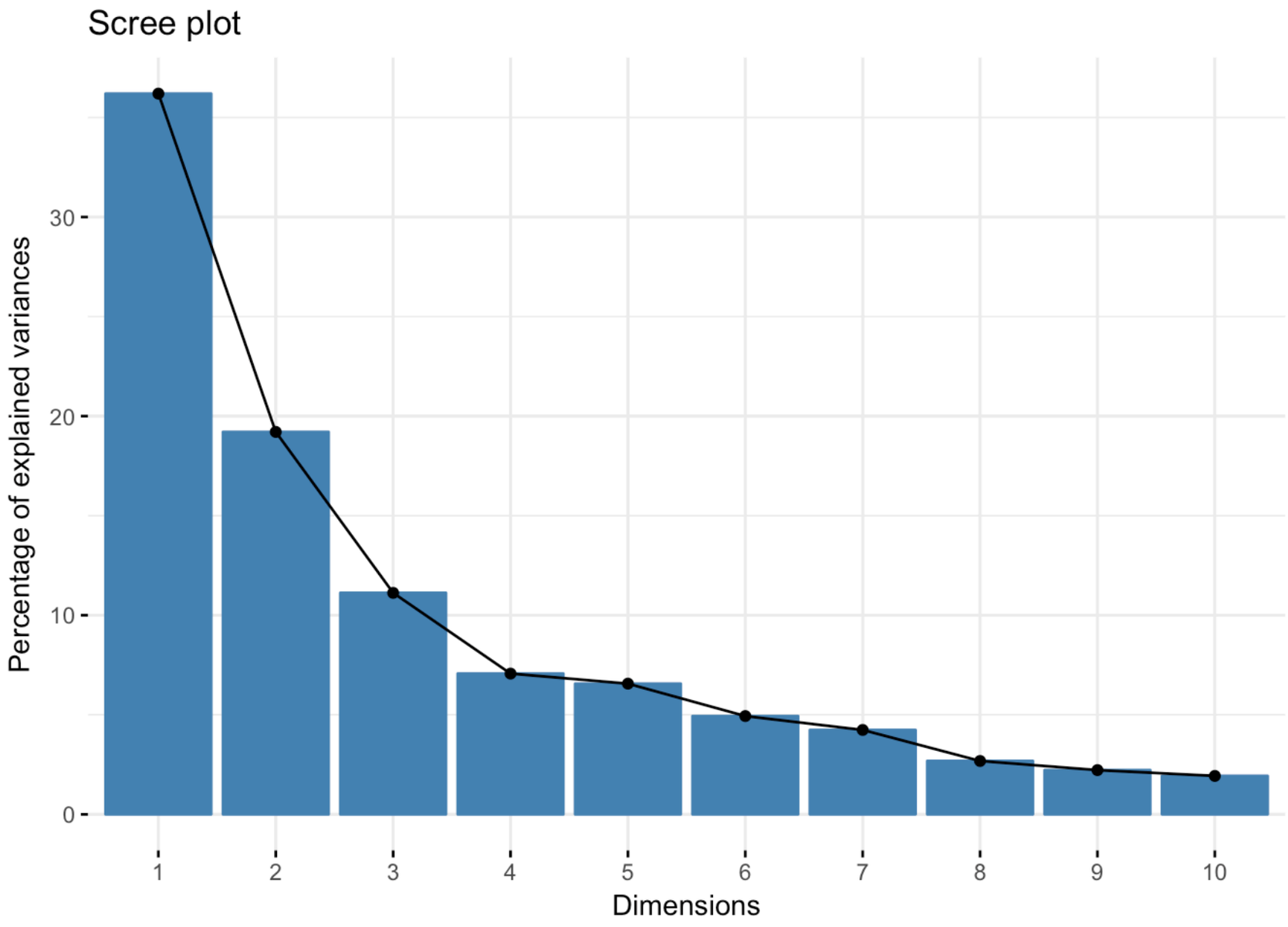
```
names(winePCA)
```

```
##  [1] "tab"  "cw"   "lw"   "eig"  "rank" "nf"   "c1"   "li"   "co"   "l1"
## [11] "call" "cent" "norm"
```

tab	the data frame to be analyzed depending of the transformation arguments (center and scale)
cw	the column weights
lw	the row weights
eig	**the eigenvalues**
rank	the rank of the analyzed matrice
nf	the number of kept factors
c1	the column normed scores i.e. the principal axes
l1	the row normed scores
co	**the column coordinates**
li	**the row coordinates i.e. the principal components**
call	the call function
cent	the p vector containing the means for variables (Note that if center = F, the vector contains p 0)
norm	the p vector containing the standard deviations for variables i.e. the root of the sum of squares deviations of the values from their means divided by n (Note that if norm = F, the vector contains p 1)

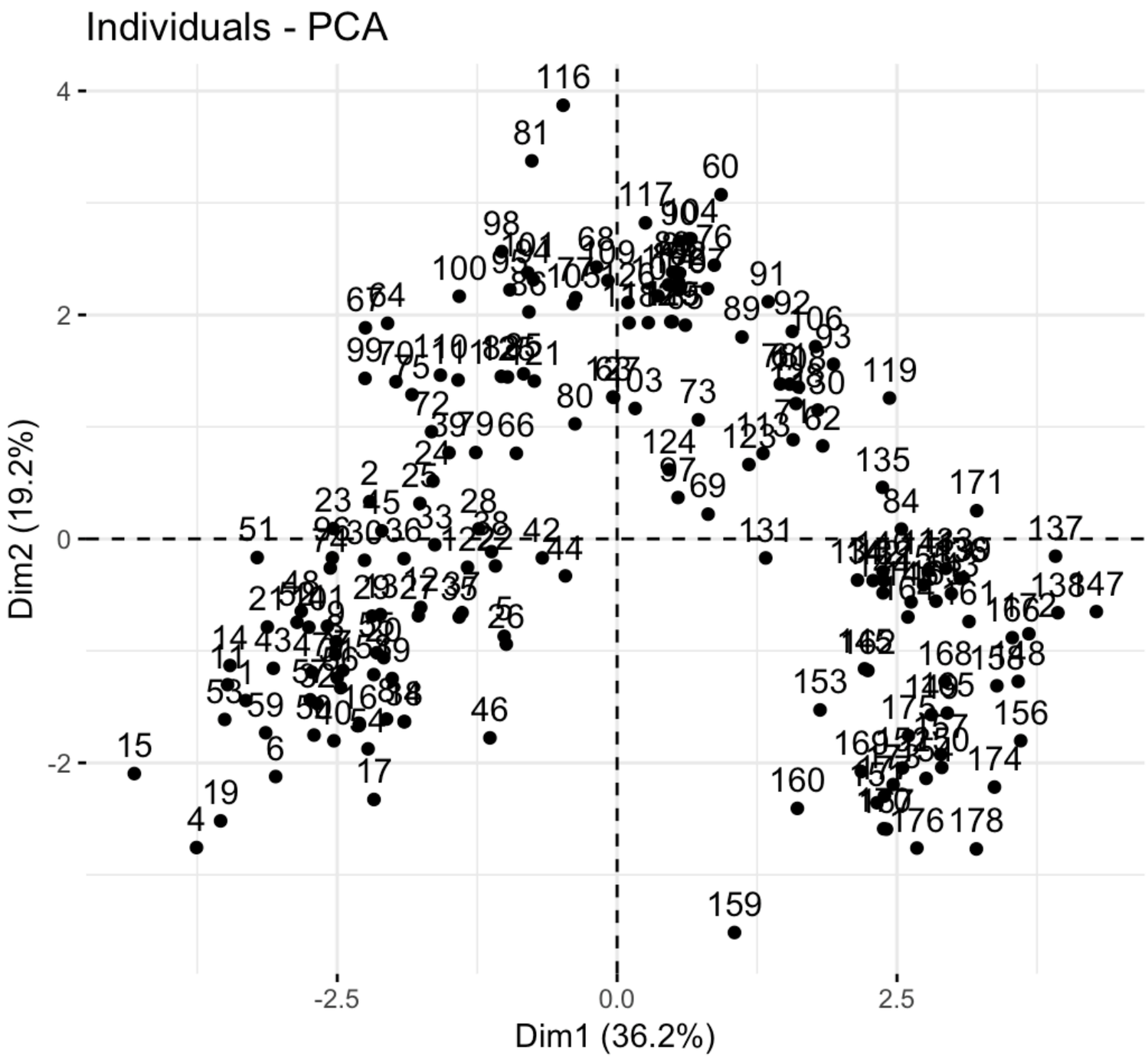
Scree Plot

```
library(factoextra)
fviz_eig(winePCA)
```



Sample Projection

```
fviz_pca_ind(winePCA) +
  coord_fixed()
```

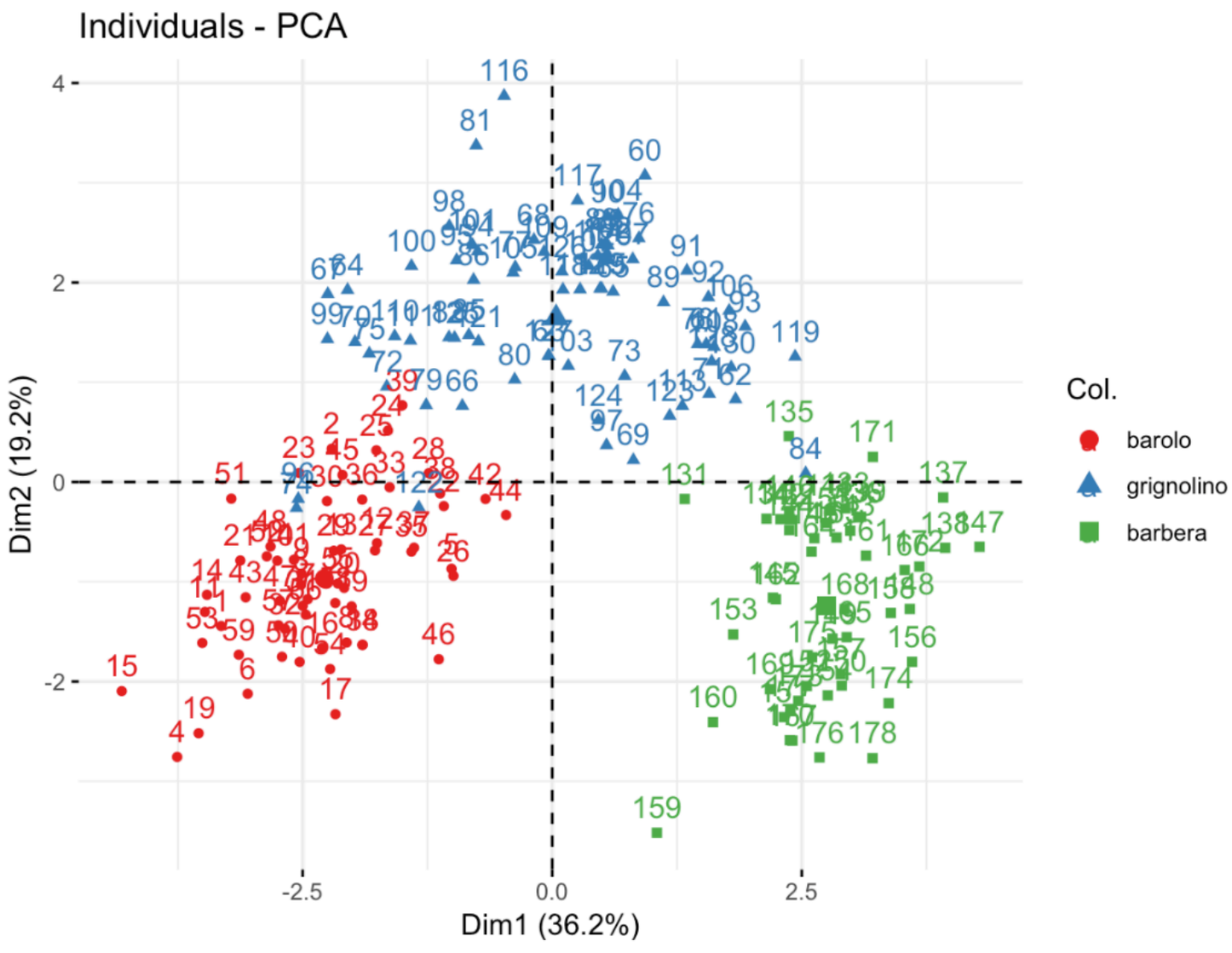


Sample Projection with Coloring by Covariates

```
load("wineClass.RData")
table(wine.class)
```

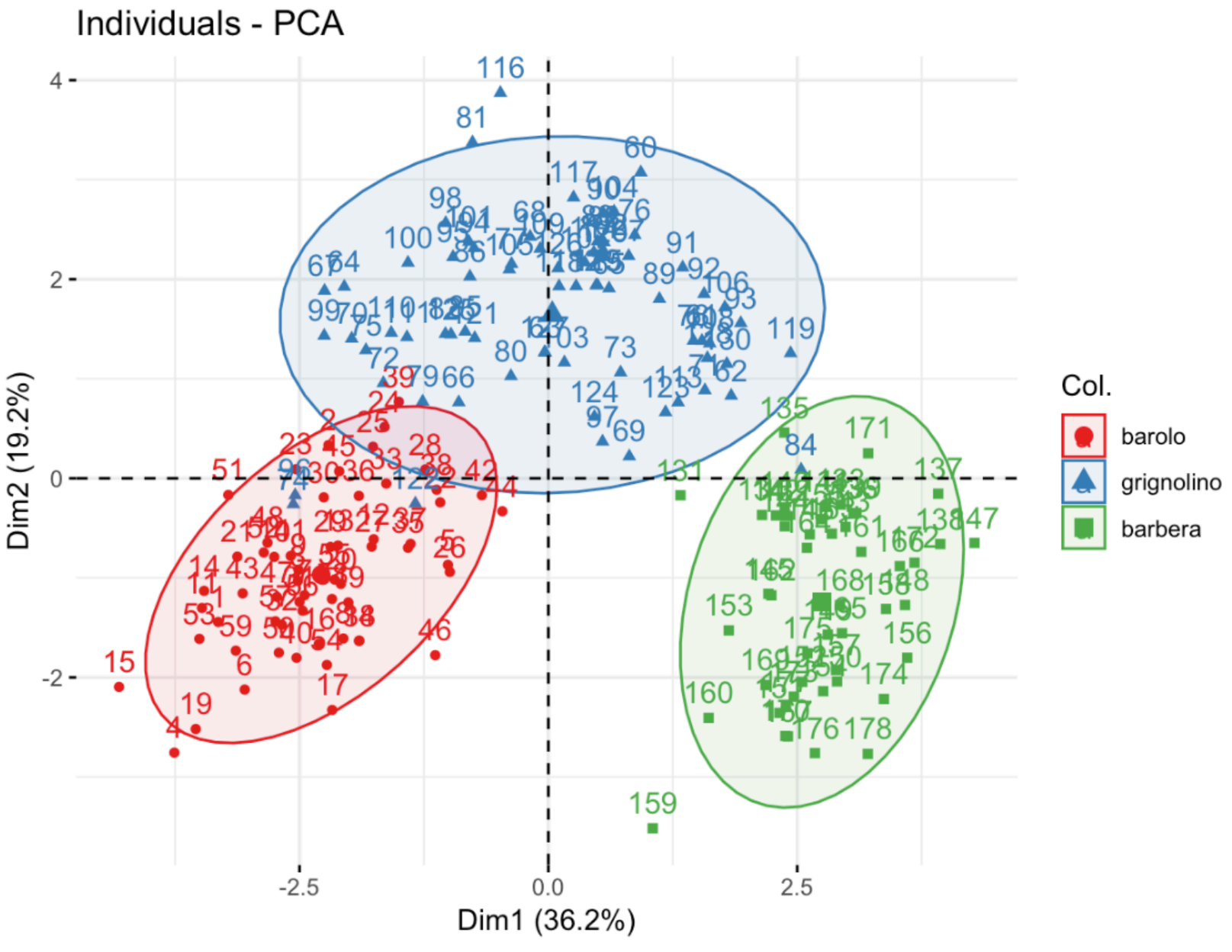
```
## wine.class
##     barolo grignolino    barbera
##         59         71         48
```

```
fviz_pca_ind(winePCA, col.ind = wine.class,
  palette = c("#E41A1C", "#377EB8", "#4DAF4A")) +
  coord_fixed()
```



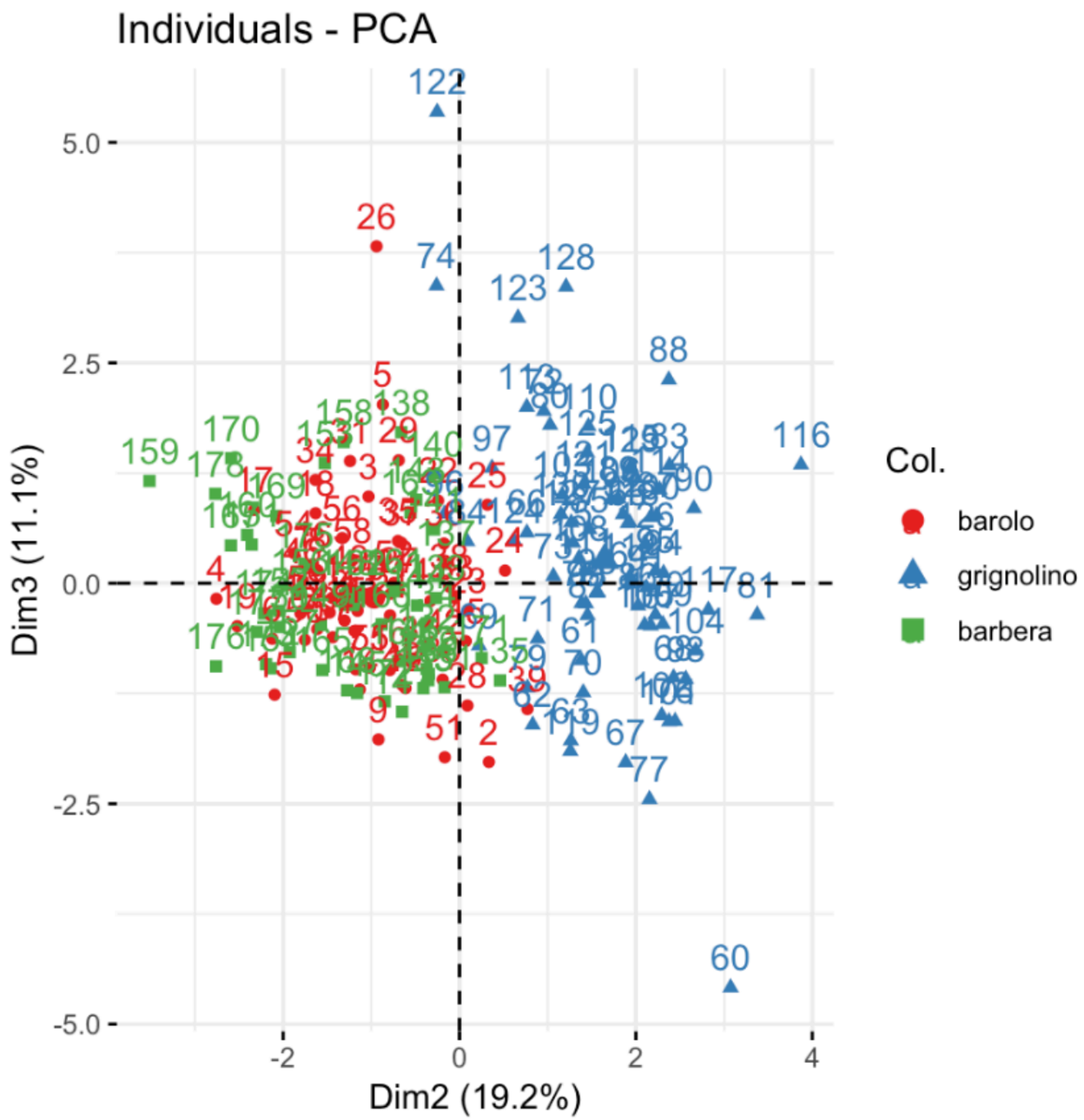
Including Ellipses

```
fviz_pca_ind(winePCA, col.ind = wine.class,
  palette = c("#E41A1C", "#377EB8", "#4DAF4A"),
  addEllipses = TRUE, ellipse.level = 0.9) +
  coord_fixed()
```



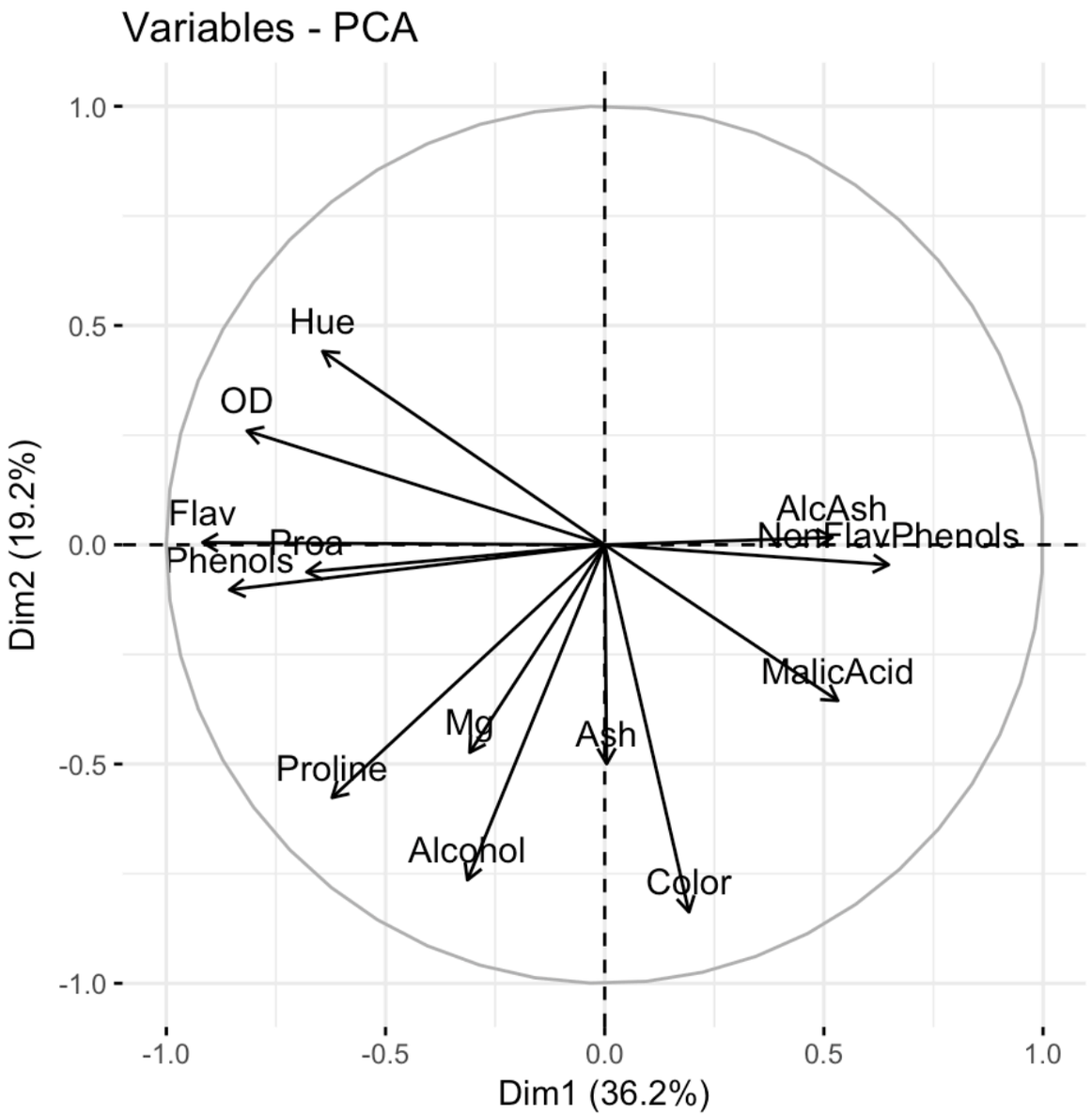
Sample Projection on Other Axes

```
fviz_pca_ind(winePCA, axes = 2:3, col.ind = wine.class,
  palette = c("#E41A1C", "#377EB8", "#4DAF4A")) +
  coord_fixed()
```



Correlation Circle for Variables

```
fviz_pca_var(winePCA)
```

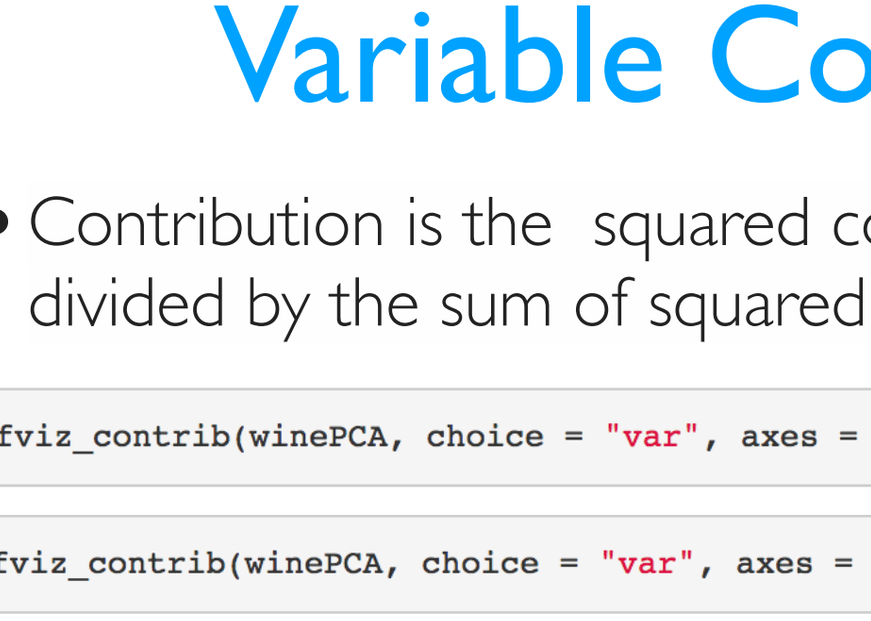


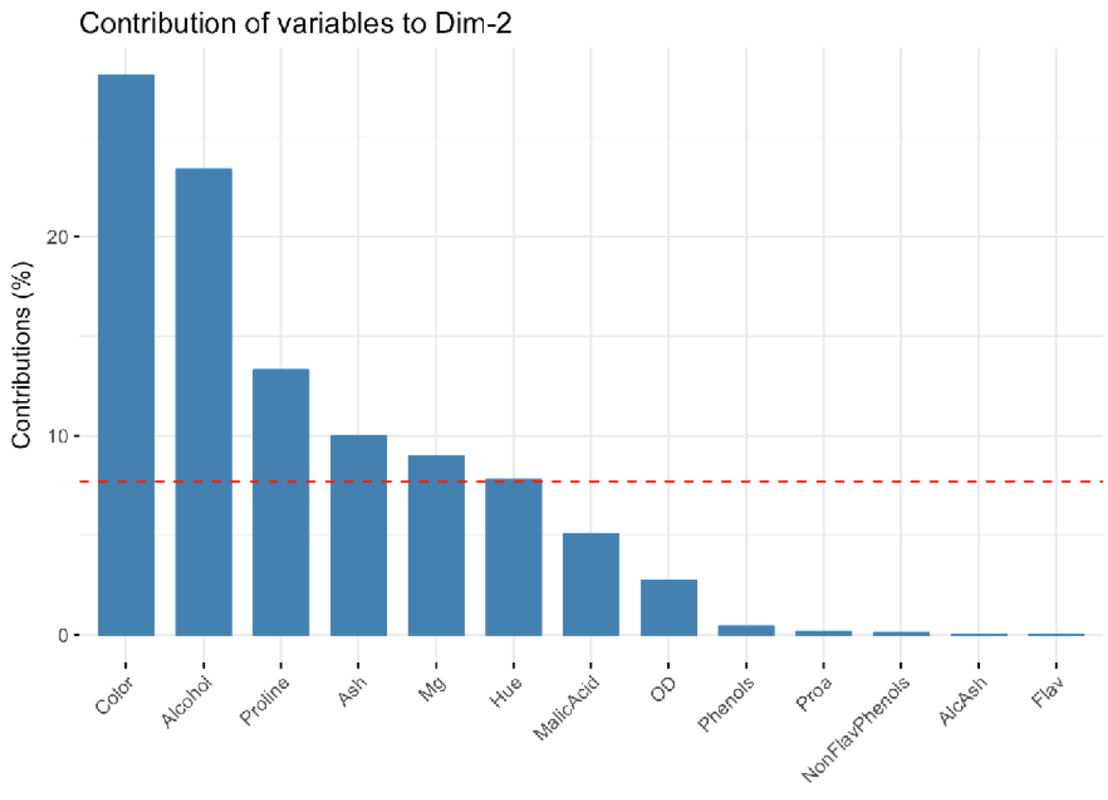
Variable Contribution to PCs

- Contribution is the squared correlation of the variable to the dimension divided by the sum of squared correlations for all variables.

```
fviz_contrib(winePCA, choice = "var", axes = 1)
```

```
fviz_contrib(winePCA, choice = "var", axes = 2)
```

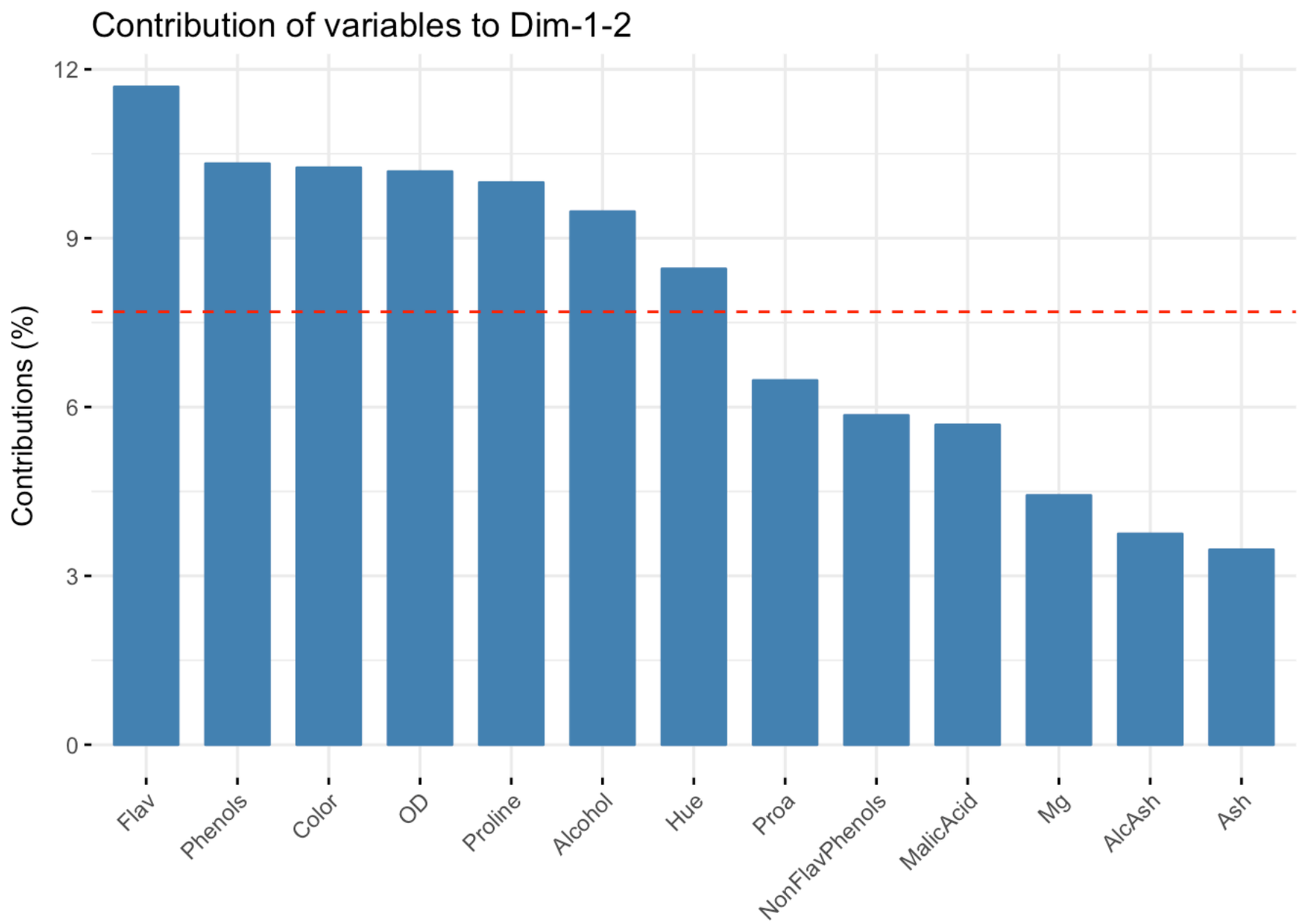




The red dashed line on the graph above indicates the expected average contribution, as if the contribution of the variables was even.

Variable Contribution to PCs

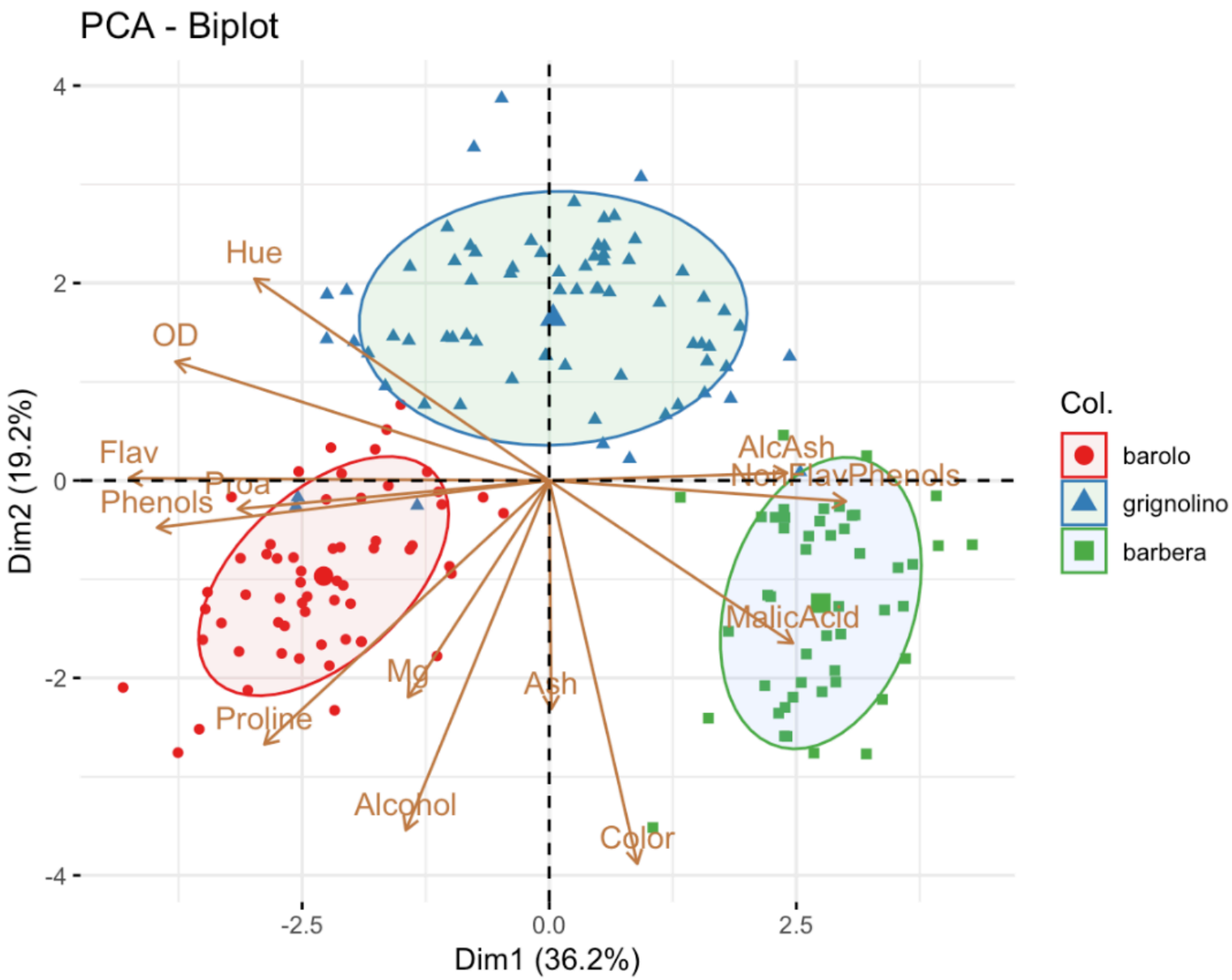
```
fviz_contrib(winePCA, choice = "var", axes = 1:2)
```



The red dashed line on the graph above indicates the expected average contribution, as if the contribution of the variables was even.

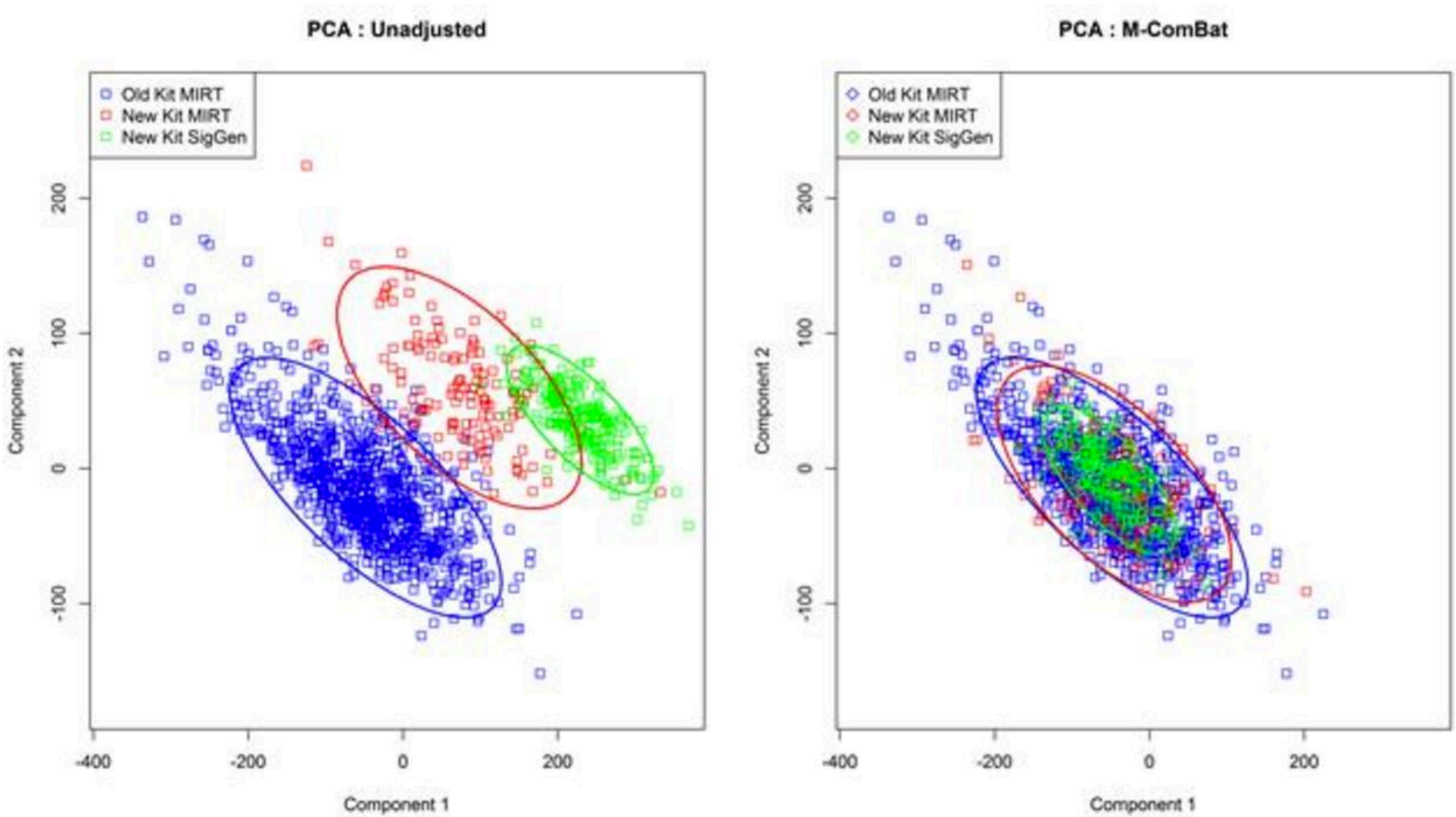
PCA Biplot with Everything Together

```
fviz_pca_biplot(
  winePCA, geom = "point",
  col.ind = wine.class,
  col.var = "#c07d44",
  addEllipses = TRUE, ellipse.level = 0.7) +
  coord_fixed() +
  scale_color_brewer(palette = "Set1")
```



Discovering Batch Effects

- PCA can be used to discover batch effects
- If there are batch effect you can use tools such as ComBat which are available in **`sva`** package .



Source: Stein et al. 2014

Summary

- **Multivariate data requires conscious preprocessing,** to make the variances of the variables comparable and their centers at the origin.
- When data contains many numerical variables, we can still make useful graphical representations by making **projections on lower dimensions** (planes and 3D are the most frequently used).
- **PCA searches for new 'more informative' variables** which are linear combinations of the original ones.
- PCA is based on finding singular value decompositions (SVD) of the matrix X, which is equivalent to the eigenanalysis of X'X. The squares of the singular values are the equal to the eigenvalues and to the variances of the new variables.
- You need to **plot eigenvalues to guide your decision on how many axes to discard** and still able reproduce the signal in the data.
- **Eigenvalues which are quite close can give rise to PC scores which are highly unstable**. In these situations, you may need to use interactive three or four dimensional projections.
- Interpretation of PCA is facilitated by covariates measured on the observations.
- **PC axes should be scaled according to the variances explained.**

Further Reading

- The best way to deepen your understanding of SVD is to read Chapter 7 of *Strang (2009)*.
- Complete textbook on PCA and related method, *Mardia, Kent, and Bibby (1979)*, is a standard text that covers all multivariate methods in a classical way, with linear algebra and matrices.
- *Jolliffe (2002)* is a booklong treatment of everything to do with PCA with extensive examples.
- Improvements to the interpretation and stability of PCA can be obtained by adding a penalty that minimizes the number of nonzero coefficients that appear in the linear combinations. *Zou, Hastie, and Tibshirani (2006)* and *D. M. Witten, Tibshirani, and Hastie (2009)* have developed sparse versions of principal components, and their packages **`elasticnet`** *(Zou and Hastie 2012)* and **`PMA`** *(D. Witten et al. 2009)* provide implementations in R .