

Unsupervised learning

- HS2015 wurde in Woche 6 Classification eingeführt, Woche 7 war Test und Praktikum
- Daumenregel grob 30 slides
- Week1
 - Einführung
 - PCA

Statistisches Data Mining (StDM)

Woche 1

- *Oliver Dürr*
- Institut für Datenanalyse und Prozessdesign
- Zürcher Hochschule für Angewandte Wissenschaften
- oliver.duerr@zhaw.ch
- Winterthur, 20 September 2016

No laptops, no phones, no problems



Multitasking senkt Lerneffizienz:

- **Keine Laptops im Theorie-Unterricht Deckel zu oder fast zu (Sleep modus)**

Bewertung

- Es gibt eine freiwillige Zwischenprüfung. Die Prüfung dauert 45-60 Minuten und ergibt eine erste Vornote, die zu 15% in Endnote zählt, falls sie besser ist als die Note der Schlussprüfung.
- Die freiwillige Bearbeitung einer Hausarbeit ergibt eine zweite Vornote, die zu 10% in Endnote zählt, falls sie besser ist als die Note der Schlussprüfung
- Die Endprüfung ist obligatorisch und dauert 90 Minuten.
- Die Modulendnote ist der grössere Wert von
 $[0.1 \times \text{HA} + 0.15 \times \text{Zwischenprüfung} + 0.75 \times \text{Endprüfung}] (*)$ und
 $[0.15 \times \text{Zwischenprüfung} + 0.85 \times \text{Endprüfung}] (*)$ und
 $[0.1 \times \text{HA} + 0.9 \times \text{Endprüfung}] (*)$ und
 $[\text{Endprüfung}] (*)$.

Die Vorleistungen (Zwischenprüfung, Praktika) sind insofern fakultative, als sie nur dann zählt, wenn sie die Endnote verbessert. Es wird keine Nachprüfungen für die Vorleistungen geben. Die Endprüfung kann bei begründetem Fernbleiben (Krankheit mit Arztzeugnis, Militärdienst, etc.) nachgeholt werden.

ZP in der Woche 8 (8 November)

ISLR: <http://www.r-bloggers.com/in-depth-introduction-to-machine-learning-in-15-hours-of-expert-videos/>

In-depth introduction to machine learning in 15 hours of expert videos

September 23, 2014

By Kevin Markham

Like Share 2k Tweet 1,048

(This article was first published on [R - Data School](#), and kindly contributed to R-bloggers)

In January 2014, Stanford University professors Trevor Hastie and Rob Tibshirani (authors of the legendary **Elements of Statistical Learning** textbook) taught an **online course** based on their newest textbook, **An Introduction to Statistical Learning with Applications in R (ISLR)**. I found it to be an excellent course in statistical learning (also known as “machine learning”), largely due to the high quality of both the textbook and the video lectures. And as an R user, it was extremely helpful that they included R code to demonstrate most of the techniques described in the book.

If you are new to machine learning (and even if you are not an R user), I highly recommend reading ISLR from cover-to-cover to gain both a theoretical and practical understanding of many important methods for regression and classification. It is available as a **free PDF download** from the authors' website.

If you decide to attempt the exercises at the end of each chapter, there is a GitHub repository of **solutions provided by students** you can use to check your work.

As a supplement to the textbook, you may also want to watch the excellent **course lecture videos** (linked below), in which Dr. Hastie and Dr. Tibshirani discuss much of the material. In case you want to browse the lecture content, I've also linked to the PDF slides used in the videos.

Chapter 1: Introduction ([slides](#), [playlist](#))

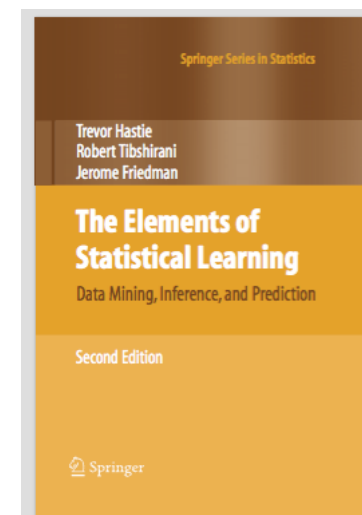
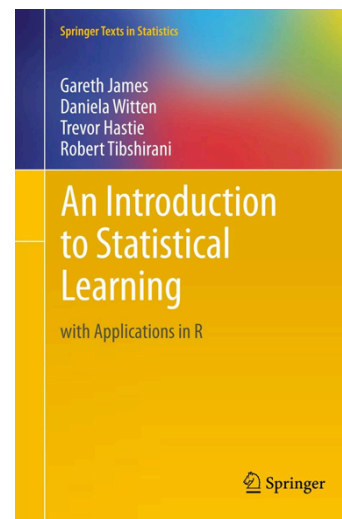
- **Opening Remarks and Examples** (18:18)
- **Supervised and Unsupervised Learning** (12:12)

Chapter 2: Statistical Learning ([slides](#), [playlist](#))

- **Statistical Learning and Regression** (11:41)
- **Curse of Dimensionality and Parametric Models** (11:40)

ISLR: Introduction to statistical learning

ESL: Elements of statistical learning (goes beyond this course)



Both books are free, see website on the left

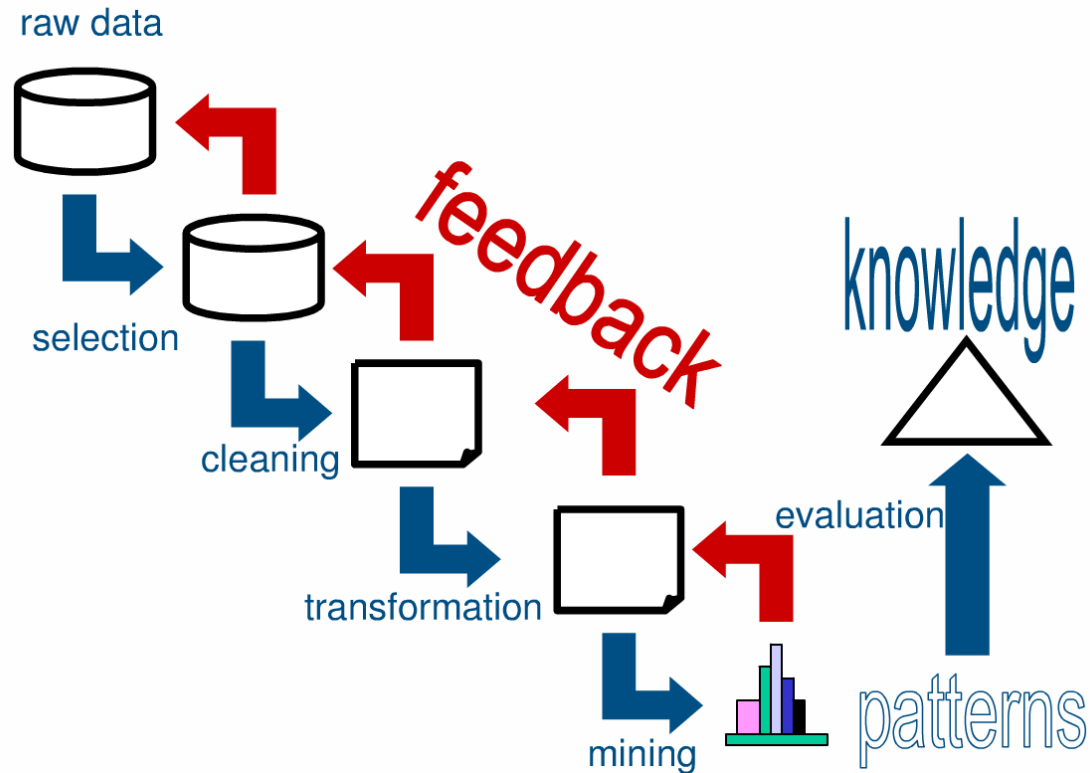
Definition: Data Mining

- Many definitions
 - "The nontrivial extraction of implicit, previously unknown, and potentially useful information from data" [1].
 - "The science of extracting useful information from large data sets or databases" [2]
 - In Principal
 - Understand Data and / or
 - Make Predictions
- The model is usually not so important
- Many very similar disciplines:
 - Machine Learning, Statistical Learning, Data Science, Data Fishing

[1] W. Frawley and G. Piatetsky-Shapiro and C. Matheus, Knowledge Discovery in Databases: An Overview. AI Magazine , Fall 1992, pgs 213-228

[2] D. Hand, H. Mannila, P. Smyth: Principles of Data Mining. MIT Press, Cambridge, MA, 2001.

The obvious data mining process



Usama Fayyad : “From Data Mining to Knowledge Discovery in Database”
1996

Example of data sets

Data: The Iris Data set

Iris Setosa



Iris Virginica



Iris Versicolor



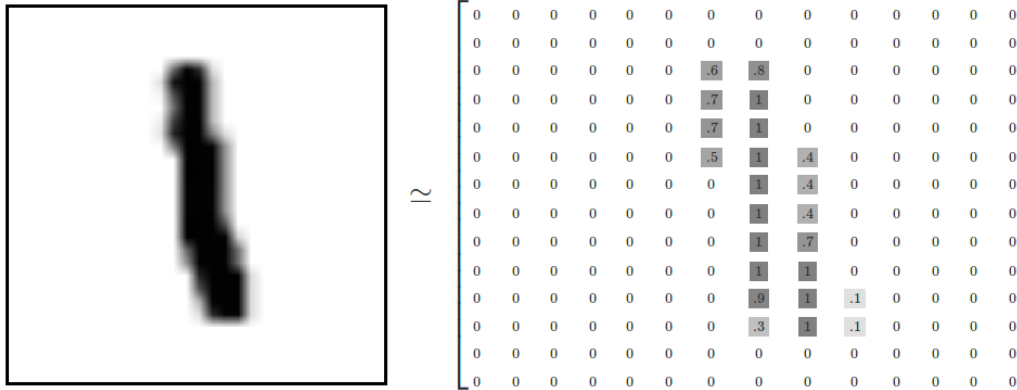
5 Features: 4 numerical, 1 class

150
Flowers

Blume	Type	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	setosa	5.1	3.5	1.4	0.2
2	setosa	4.9	3	1.4	0.2
3	virinica	3.3	3.2	1.6	0.5
4	setosa	5.1	3.5	1.4	0.2
...
150	virinica	4.9	3	1.4	0.2

Fisher, R.A. "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188 (1936)

MNIST Handwritten Digits



$28 \times 28 = 784$ features

Row Names	Digit	Pixel 1	Pixel 2	Pixel 3	Pixel 4	Pixel 5	...	Pixel 256
Sample16	0	0	0	0	0	1		0
Sample78	8	0	1	1	1	1		0
Sample79	3	1	1	1	1	1	...	0
Sample80	2	0	0	0	1	1		1
Sample81	1	0	0	0	0	0		0
Sample82	2	0	0	0	0	1		1
Sample83	4	0	0	0	0	1		1

60000 cases

Document data (Bag of Words coding)

- Each document is described by the occurrence of words

~30'000 Features

~1000 Documents

Document	Type	Viagra		Loerrach	Pizza
1	SPAM	9		0	0
2	NO SPAM	1		4	1
3	SPAM	3		1	0
...
1000	SPAM	4		1	1

- Quite successful
- Better codings such as word2vec (maybe later)

Word Vector

- Other data has to be transformed in vectors first...

```
model['dog'].shape
```

```
(100,)
```

```
model['dog'][:10]
```

```
array([ 0.05753701,  0.0585594 ,  0.11341395,  0.02016246,  0.11514406,  
        0.01246986,  0.00801256,  0.17529851,  0.02899276,  0.0203866 ])
```

‘dog’ → Big Complex Machine → (0,10,20,002,1,0...,334)

Words „embedded“ into 100-dimensional real vector space, words with similar meanings tend to end up in close proximity (and distance vectors are interpretable as well!)

Allgemeine Struktur der Daten

- Daten-Objekte und deren Attribute (wie Excel Sheet)
- Ein Variable oft auch Feature ist eine Eigenschaft einen Objekts
 - Beispiel: Augenfarbe, Kelchblattbreite, etc.
 - Für Klassifizierung gibt es oft ein ausgezeichnetes Attribut (Klassenattribut)
- Eine Sammlung von Attributen beschreibt ein Objekt
 - Objekt auch: Individuum, Instanz, Fall, Datum, Record oder Beispiel

p Features

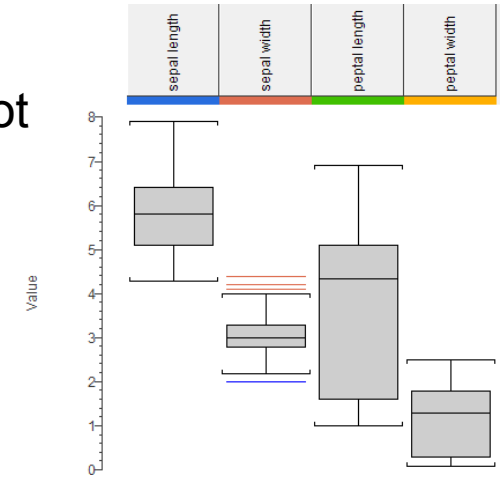
Blume	Type	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	setosa	5.1	3.5	1.4	0.2
2	setosa	4.9	3	1.4	0.2
3	virginica	3.3	3.2	1.6	0.5
4	setosa	5.1	3.5	1.4	0.2
...
150	virginica	4.9	3	1.4	0.2

n Objekte

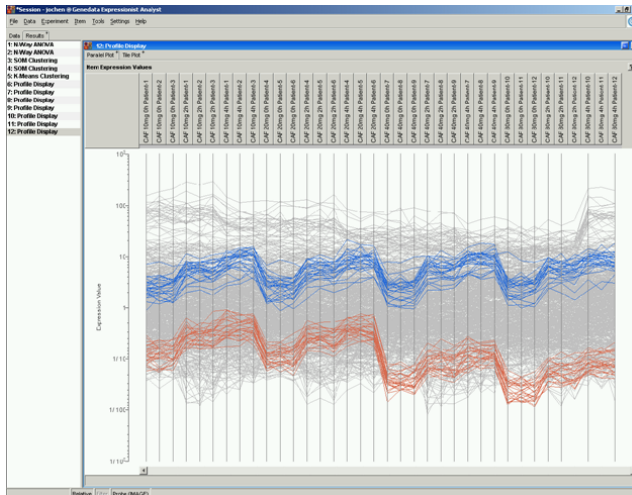
Methoden des DM: Visualisierung

- Beschreibende Methoden
- Mensch ist gut im Finden von Mustern
- Daten geeignet darstellen

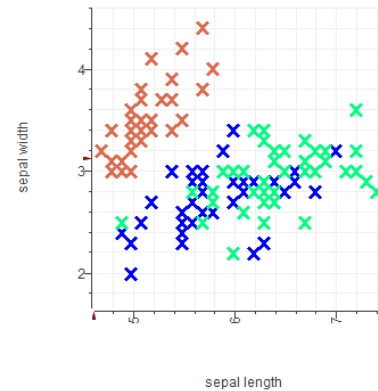
Box Plot



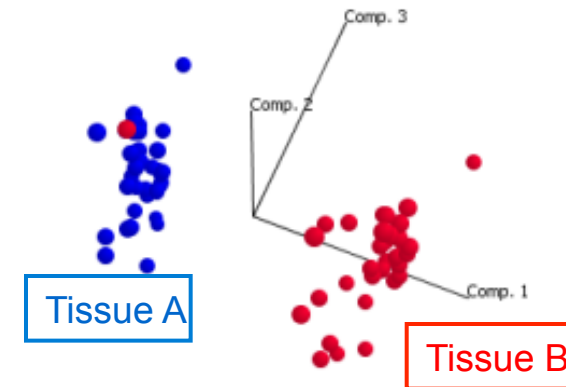
Parallel Coordinate Plot



Scatter plot

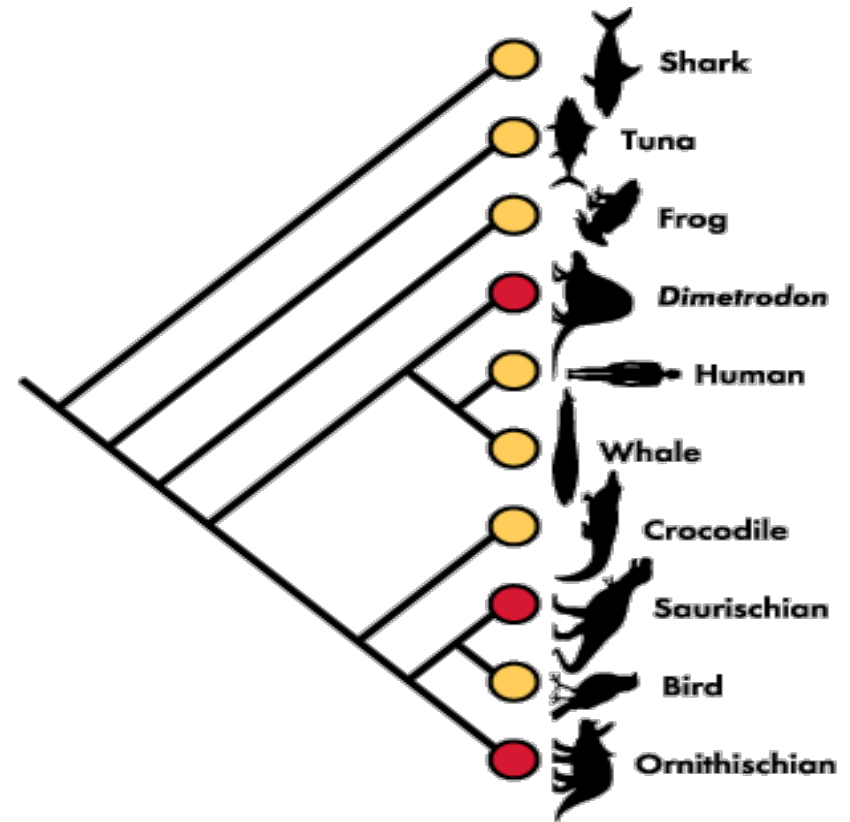


PCA



Methoden des DM: Clustering

- Beschreibende Methoden
- Anordnen so dass ähnliche Objekte beieinander liegen
- Finden von neuen Zusammenhängen
- **Was ist ähnlich?**
- Weitere Anwendung:
 - Automatisches Gruppieren von Daten
 - ...



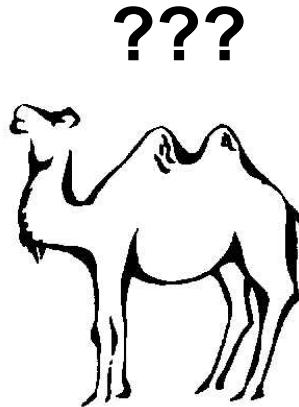
Daten:

Shark: GTATAGAT...TAGC

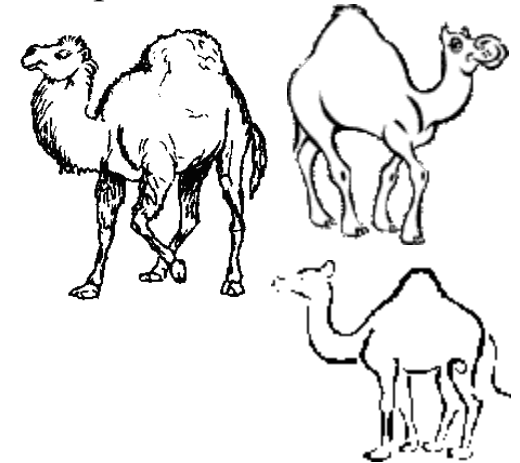
Tuna : TGATAGT...TATAA

Methoden des DM: Klassifizierung

- Vorhersagende Methode
- Idee:
 - Anhand von Beispielen unbekannte Objekt zu Klassifizieren
- Wie beschreibt man die Objekte?



Beispiele fuer Dromedare



Beispiele fuer Kamele



Daten Matrix

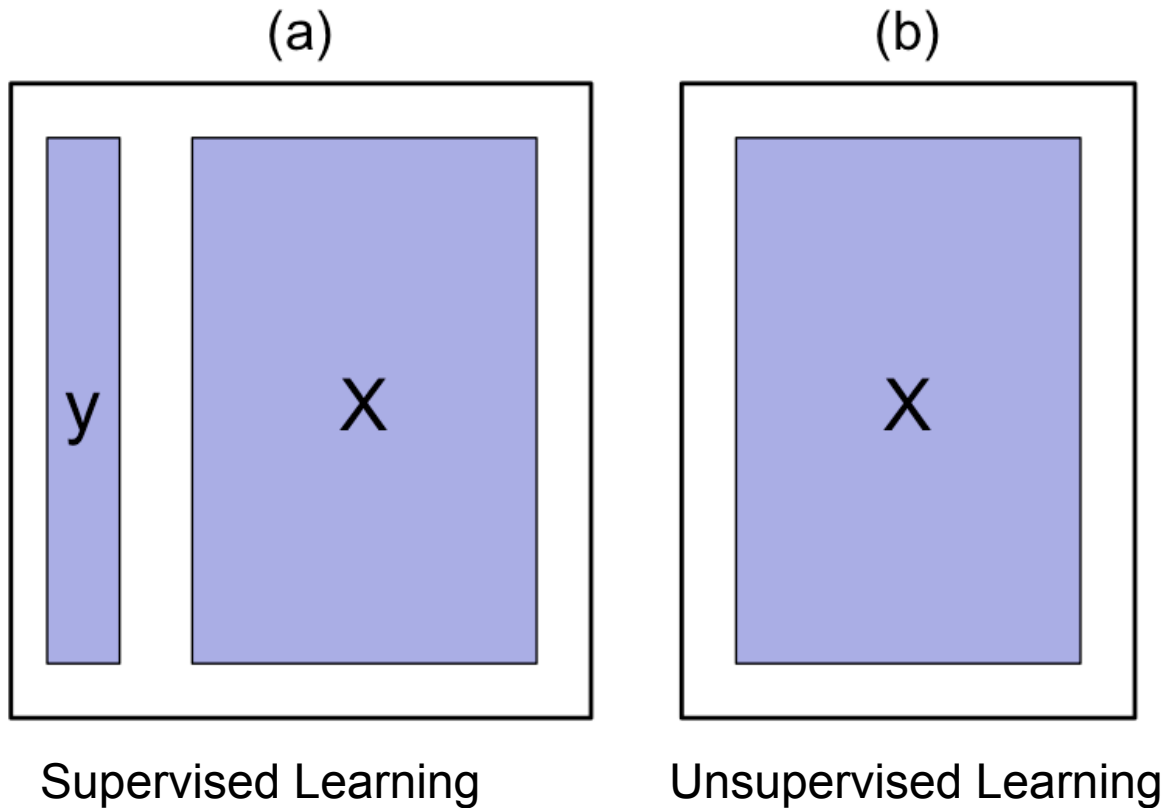


Tier	Art	Anzahl Beine	Anzahl Höcker	Anzahl Augen
1	Dromedar	4	1	2
2	Kamel	3	2	2
...
150	Kamel	4	2	2

- Oft nicht so leicht wie mit den Höckern
- Daten Matrix Supervised (mit grün) und Unsupervised (ohne Grün) Learning

Supervised vs. Unsupervised Learning

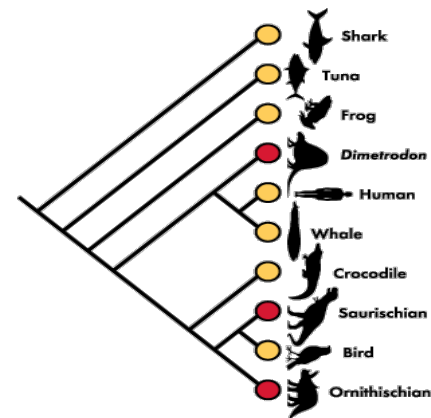
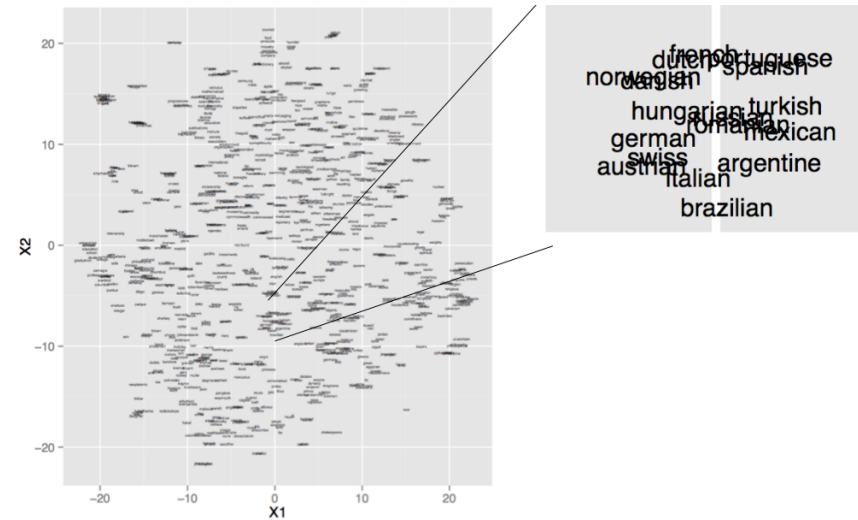
- Supervised Learning: both X and Y are known
- Unsupervised Learning: only X



Overview of the semester

Part I (Unsupervised Learning)

- Dimension Reduction
 - PCA
- Similarities, Distance between objects
 - Euclidian, L-Norms, Gower,...
- Visualizing Similarities (in 2D)
 - MDS, t-SNE
- Clustering
 - K-Means
 - Hierarchical Clustering



Part II (Supervised Learning)

- ...

PCA

Dimensionsreduzierung

- In machen Anwendungen des DM hat man extrem viele Attribute
 - Geneexpression: etwa 30'000 Expressionswerte
 - Dokumente haben 10'000 verschiedene Wörter
- Attribute zu streichen (Feature subset selection)
 - Attribute die nichts (neues) bringen entfernen
 - Ist ein Attribut einfach immer 2 mal ein anderes kann man es gleich weglassen.
- Neue Attribute erzeugen (Dimensionsreduzierung)
 - Man erzeugt ein paar neue Features aus den alten
 - PCA
 - Auch sehr gut geeignet für Visualisierung 2-3 Attribute aus 30'000.

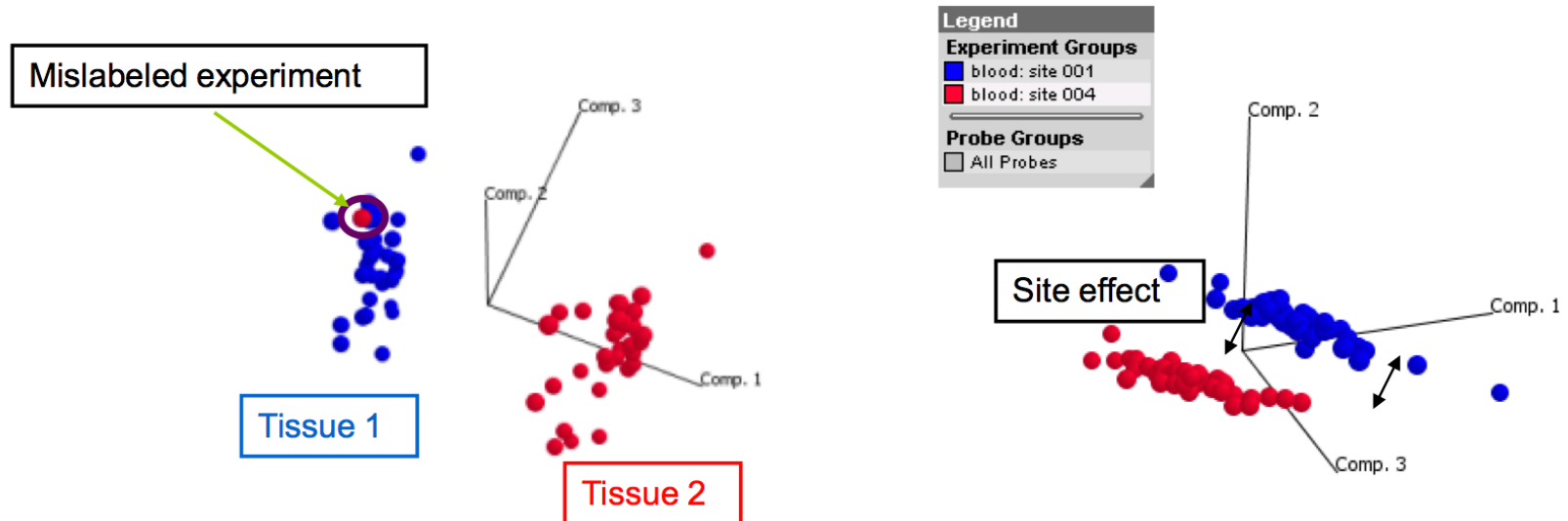
Use cases for dimension reduction

- Example: Gene-expression 30'000 Variables, 45 Patients
- Is all correct?

1		Var1	Var2	Var3	Var4		Var29998	Var29999	Var30000
2	Exp 1	0.72575697	0.90912727	0.34183219	0.05807989		0.89604396	0.82978394	0.67709992
3	Exp 2	0.86738472	0.18892562	0.10061359	0.86472149		0.42230826	0.29717662	0.84183402
4	Exp 3	0.536584	0.1432163	0.48031828	0.54135801		0.35761078	0.93715841	0.09946435
5	Exp 4	0.9750189	0.01024606	0.5091362	0.15267143		0.21621312	0.18110699	0.04165032
6	Exp 5	0.54058652	0.86667423	0.7723371	0.29263193		0.34003667	0.99724025	0.50546296
7	Exp 6	0.51420486	0.68710973	0.53443674	0.39784944		0.09120201	0.02095151	0.7874859
8	Exp 7	0.84066935	0.77631801	0.88487998	0.44679161		0.06190021	0.5841718	0.79347598
9	Exp 8	0.31751833	0.16318624	0.79276299	0.08440983		0.83189181	0.57771774	0.08795787
10	Exp 9	0.25660099	0.26809617	0.87877424	0.41877501		0.23589796	0.84845295	0.12819384
11	Exp 10	0.2133815	0.35005992	0.72679513	0.93096216		0.06484648	0.87936602	0.98808592
12	Exp 11	0.67621687	0.82208455	0.41252689	0.51356375		0.83390677	0.58056355	0.53156471
13	Exp 12	0.83698302	0.39427289	0.64891165	0.38946918		0.86980017	0.02408343	0.1279679
14	Exp 13	0.55158332	0.83563428	0.08601078	0.95342836		0.73062691	0.69545093	0.9684515
15	Exp 14	0.88240173	0.93238978	0.34213598	0.25428121		0.7835426	0.08090319	0.34138029
16	Exp 15	0.35257344	0.04100738	0.08441876	0.54433121		0.39901506	0.75584409	0.90975939
17	Exp 16	0.66411782	0.26894263	0.41570473	0.90789058		0.22048541	0.06748163	0.56058675
18	Exp 17	0.37758333	0.87416083	0.34331478	0.57516207		0.05496016	0.65253258	0.13182802
19	Exp 18	0.16254575	0.45823383	0.41947507	0.20725022		0.76805359	0.29279849	0.17038373
20	Exp 19	0.09717491	0.16687942	0.69490897	0.00982516	...	0.91838419	0.33659043	0.92492301
21	Exp 20	0.18632352	0.26820246	0.43650327	0.87902106		0.24433306	0.63146315	0.32357043
22	Exp 21	0.16356459	0.64175502	0.54539885	0.44821048		0.69554721	0.23790817	0.16114107
23	Exp 22	0.50602018	0.20370984	0.38225406	0.797264		0.14490904	0.44722882	0.31422467
24	Exp 23	0.30294307	0.73866033	0.78865558	0.3532843		0.04564231	0.00854157	0.93346583
25	Exp 24	0.37893552	0.36223255	0.56103558	0.68409418		0.37509117	0.65625123	0.85521739
26	Exp 25	0.48307778	0.70717319	0.73891708	0.56796612		0.21659263	0.89437545	0.3689527
27	Exp 26	0.76200969	0.71219127	0.01349004	0.3164314		0.13000069	0.06598902	0.64582494
28	Exp 27	0.73420576	0.48717756	0.90633582	0.78943633		0.39879527	0.66474155	0.87347295
29	Exp 28	0.81817237	0.96946477	0.10527094	0.34758947		0.41245137	0.05720508	0.64870324
30	Exp 29	0.46189251	0.92516654	0.88555359	0.94335229		0.5822599	0.23685582	0.45742172
31	Exp 30	0.64594505	0.89508066	0.00344762	0.01764184		0.98594893	0.73566371	0.65856274
32	Exp 31	0.29977128	0.57625009	0.42203689	0.53401962		0.73209191	0.78395094	0.79902787
33	Exp 32	0.24302476	0.30748217	0.13336479	0.06307744		0.43565341	0.80502196	0.54948119
34	Exp 33	0.10313471	0.81037692	0.1019815	0.95377739		0.93837378	0.32174496	0.06968822
35	Exp 34	0.57267352	0.90832149	0.01010143	0.79129947		0.99566332	0.60311529	0.94254175
36	Exp 35	0.50726326	0.37511111	0.42905886	0.07225932		0.04399464	0.50687479	0.55638528
37	Exp 36	0.25945485	0.87641759	0.21325063	0.07747172		0.76669305	0.62018391	0.6870877
38	Exp 37	0.74949084	0.4022447	0.67541199	0.63615423		0.10450498	0.70701564	0.5133567
39	Exp 38	0.02851173	0.72048367	0.64104403	0.85594184		0.6012205	0.68162603	0.2020788
40	Exp 39	0.34281377	0.00438227	0.72128967	0.95437483		0.23733739	0.90221594	0.82944182
41	Exp 40	0.59605972	0.47063129	0.0386638	0.01654464		0.75888817	0.71401908	0.68048745
42	Exp 41	0.1990435	0.56673445	0.81536695	0.63442106		0.07796896	0.14863039	0.70065248
43	Exp 42	0.08196709	0.21791967	0.05331609	0.32315459		0.35220877	0.04819191	0.77405895
44	Exp 43	0.40558134	0.79050103	0.27871425	0.24711674		0.66015164	0.00739487	0.64399704
45	Exp 44	0.4931796	0.76353204	0.99650294	0.27611642		0.06225584	0.02230363	0.00332453
46	Exp 45	0.82594597	0.0448367	0.46463296	0.75938489		0.82920136	0.36006835	0.11734431

Use cases for dimension reduction

- Example: Gene-expression 30'000 Variables, 45 Patients
- Is all correct?



Principal Component Analysis

- References
 - ISLR 10.2
 - Skript rkst
- Unsupervised method (we don't need the class label)
- Used to visualize high dimensional data
- Dimension reduction technique

Basic idea of PCA

- This is Lui from the great cartoon La Linea



- Lui has a problem he wants to analyze **two variables** but can only **visualize 1-dimensional data**.
- Lets see how we can help Lui to just use only 1-dimension to visualize his 2-dimensional expression data.
- By helping Lui we hope to learn how we can use only 2-3 dimensions to visualize our 30000 dimensional data.

Basic idea of PCA

- Suppose we have 2-dimensional data but could only visualize 1-D data.
- Example data 50 examples 2 variables

	Var1	Var2
Exp 1	42.3	321
Exp 2	32	32.3

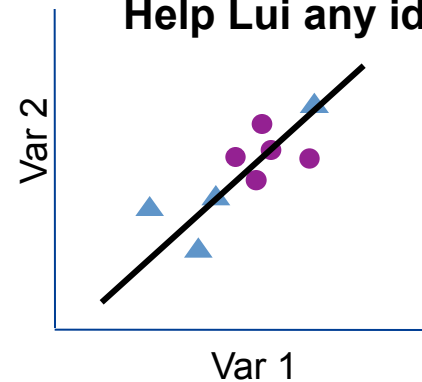
Exp 50	42	13

1-Dimensional View (as Lui would see it)



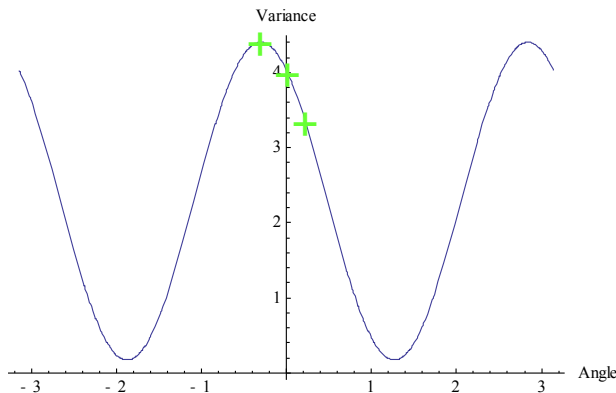
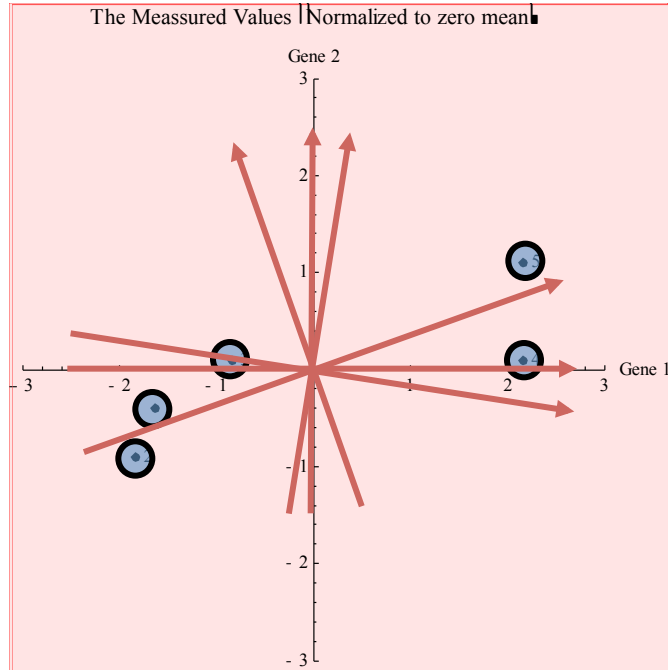
- After the reduction still many features of the data are visible and Lui is happy.

2 Dimensional View
(as we would see it).
Help Lui any idea?



General idea of PCA: Find axis with explains the data best. And use just a few (1 for Lui) (2 or 3 for humans).

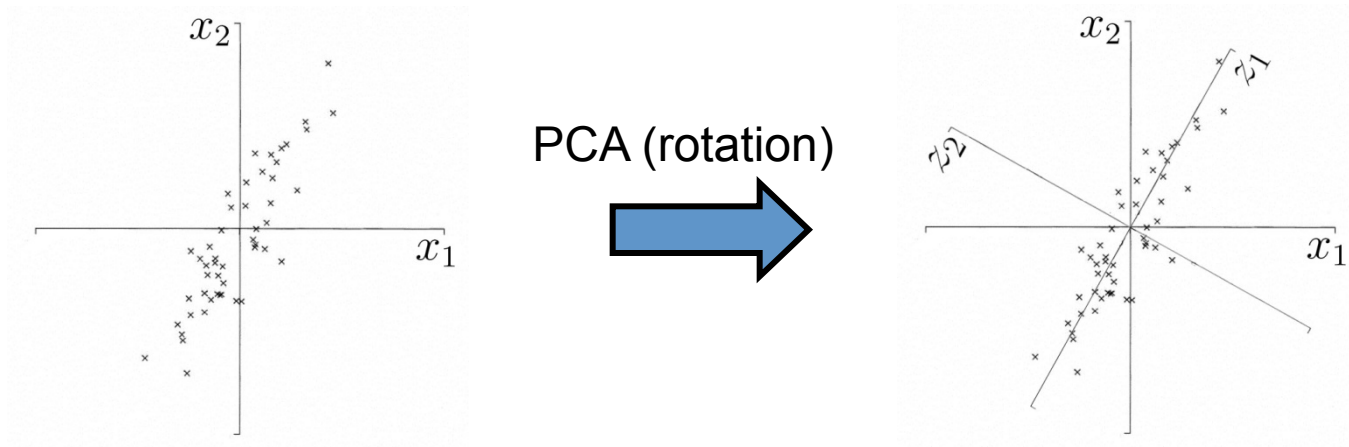
Intuitive Definition of the PCA as a rotation



- **Definition:**
Rotate your coordinate system so that the data has the **highest variance** in the **first component**, the **second highest** in the **second component** ...
- **Example** Here the variance for the first component is maximal for a rotation of -0.30 rad or about -18 degree. Since we are in 2d the problem is already defined.
- Q: Is this always possible?
- A: Yes it is (linear algebra next slide)
- Q: It must be quite complicated to find the rotation.
- A: No! Linear algebra is so beautiful-- one just has to diagonalize the covariance matrix.

PCA for dimension reduction:

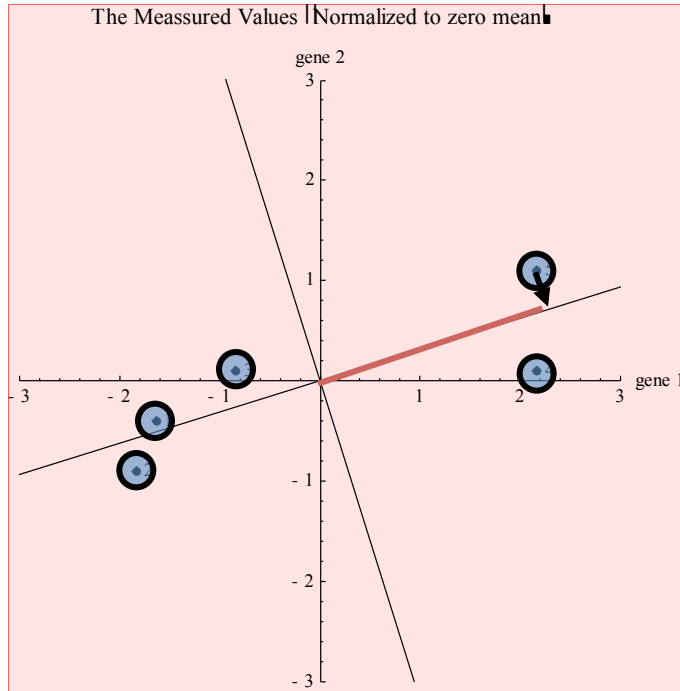
- Situation: We now have a **rotation** into a new coordinate system.



- **No data is lost in that transformation.** The first component explains most of the variance, the second the second most,.... All components explain the whole variance.
- Now take only a few components and hope that data is explained by them.
- How good is the approximation? A measure is the **explained variance**.

Explained Variance definition and Example

Example Data Set 1



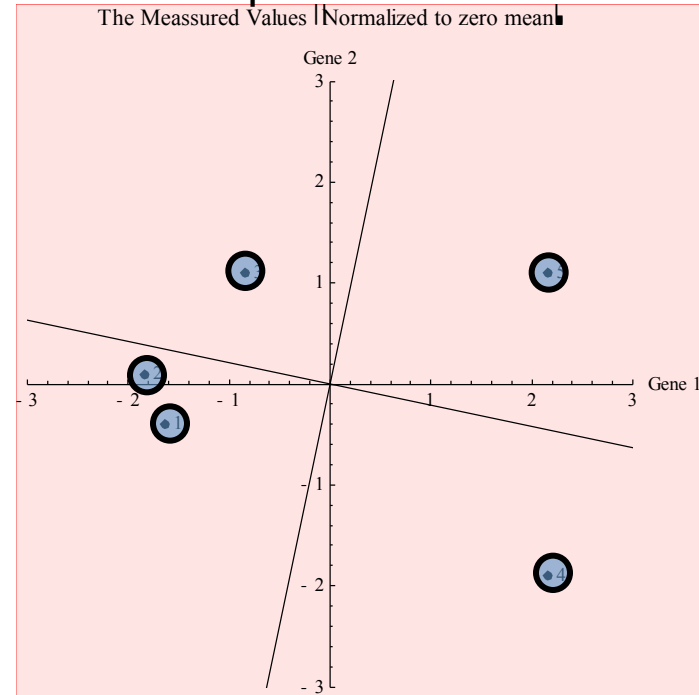
Variance: Sum of squares of all “ “: —

$\sqrt{4.40469, 0.17331}$

Explained variance percentage of total
96% = $4.40 / (4.40 + 0.17)$, **4%**.

First component already
 explains data to a great deal.

Example Data Set 2



Variance:

$4.14257, 1.43543$

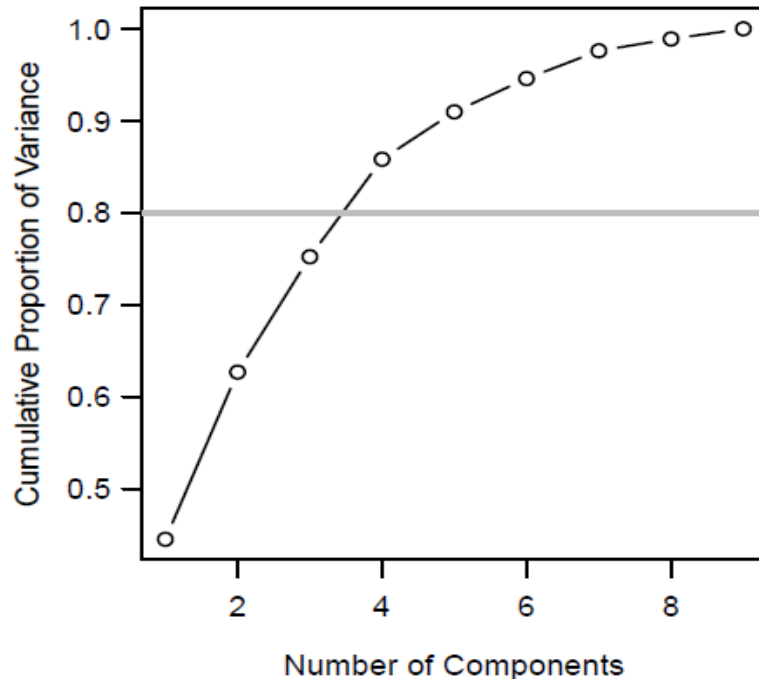
Explained variance:
74%, 26%.

First component alone might
 not be sufficient to explain the data.

How many PCs do we need? First criterion

The total variance can be calculated as: $V_{total} = \sum_{j=1}^p \text{var}(X_j) = \sum_{j=1}^p \text{var}(Y_j) = \sum_{j=1}^p \lambda_j$
(the total variance is preserved under rotation)

Rule of thumb: ~80% of Var_{total} should be explained by the first k PCs

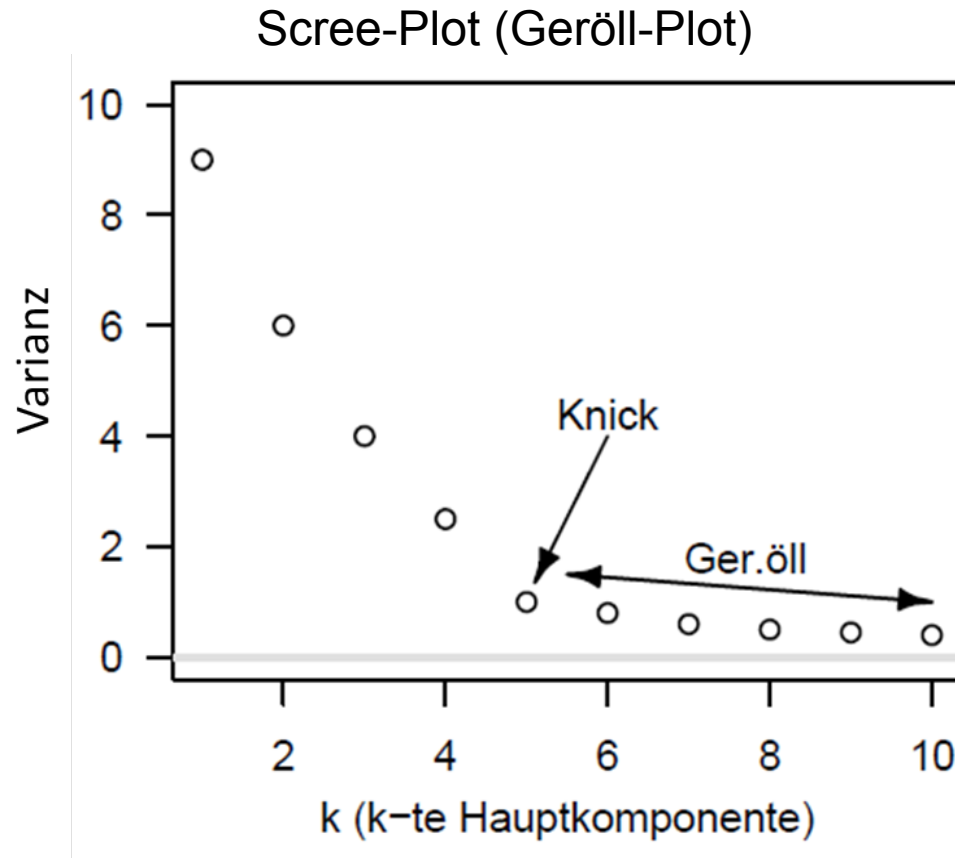


Gütekriterium der Approximation
durch die ersten k Hauptkomponenten:

$$P_k = \frac{\sum_{j=1}^k \text{var}(Y_j)}{V_{total}} \in [0,1]$$

How many PCs do we need? Second criterion

The position of the bend in the scree-plot indicates how many PCs are needed. After the bend in the scree-plot we do not gain much when adding more PCs.



Aufgabe 2

PCA von Hand ¶

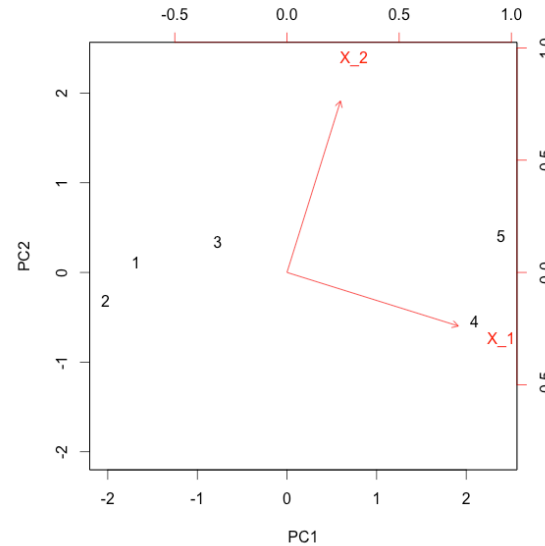
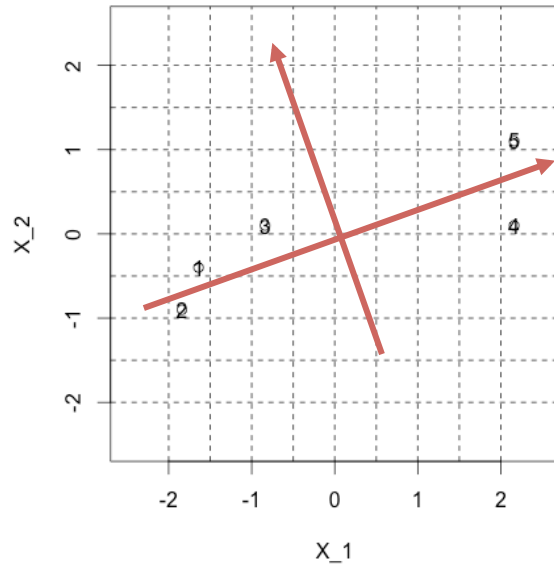
Wir wollen nun eine PCA der Datenpunkte mit Lineal und Papier durchführen. Die Genauigkeit ist hier nicht so wichtig. ¶

```
> X
      [,1] [,2]
[1,] -1.64 -0.4
[2,] -1.84 -0.9
[3,] -0.84  0.1
[4,]  2.16  0.1
[5,]  2.16  1.1 ¶
```

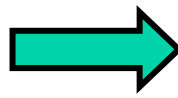
- a) → Zeichnen Sie die Punkte in das Millimeterpapier ein. ¶
- b) → Zeichnen Sie dann in das Diagramm ein gedrehtes Koordinatensystem so ein, dass die erste Achse PC1 die grösste Varianz hat. Die Achse PC2 ist dazu senkrecht. ¶
- c) → Welche Koordinaten haben die Punkte in dem neuen Koordinatensystem (etwa). ¶
- d) → Welche Koordinaten haben die Einheitsvektoren in dem neuen Koordinatensystem (etwa) ¶
- e) → Zeichnen Sie die Daten in ein neues Diagramm mit den Achsen PC1, PC2. Zeichnen Sie dort auch die Einheitsvektoren des ursprünglichen Koordinatensystems ein. ¶
- f) → Vergleichen Sie Ihre Ergebnisse mit dem R output ¶

```
> res = prcomp(X, scale. = FALSE)
> res$x          # The PCA-Transformed values
      PC1      PC2
[1,] -1.6846468  0.1075419
[2,] -2.0247206 -0.3100102
[3,] -0.7719022  0.3460737
[4,]  2.0914512 -0.5490283
[5,]  2.3898185  0.4054229
> res$rotation   # The Rotation Matrix
      PC1      PC2
[1,]  0.9544511 -0.2983673
[2,]  0.2983673  0.9544511
> biplot(res, scale=FALSE) ¶
```


Results



n: # observations
p: # feature



PCA
rotation

$$Z = X U$$

```
> X
      [,1] [,2]
[1,] -1.64 -0.4
[2,] -1.84 -0.9
[3,] -0.84  0.1
[4,]  2.16  0.1
[5,]  2.16  1.1
```

Coordinates with
respect to original
variables

$$z_1 = u_{11}x_1 + u_{21}x_2$$

$$z_2 = u_{12}x_1 + u_{22}x_2$$

U contains loadings

```
      PC1      PC2
[1,] -1.6846468  0.1075419
[2,] -2.0247206 -0.3100102
[3,] -0.7719022  0.3460737
[4,]  2.0914512 -0.5490283
[5,]  2.3898185  0.4054229
```

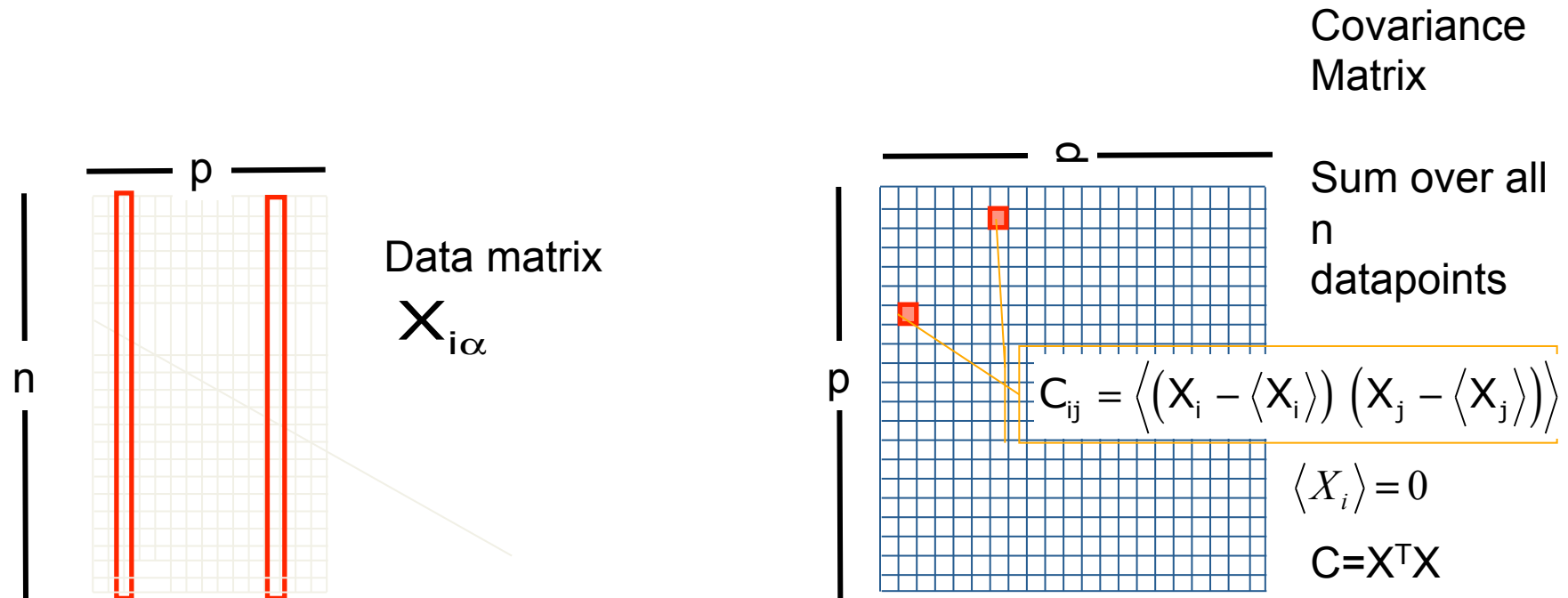
Coordinates with
respect to principal
components

U is an orthogonal
Matrix

Mathematical Definition

- We have defined the PCA as **a rotation (or reflection)** so that the first PC has the highest variance, the second is orthogonal to the first and has the second highest variance, ...
- How can one calculate this?
- It turns out that one can solve either
 - the Eigenvalue Problem of $X^T X$
 - or the Singular Value Decomposition of X

Covariance Matrix as $X^T X$



It turns out that the optimization corresponds to a diagonalization of the covariance matrix C .

$$C = U D U^T$$

PCA as eigenvalue problem of $X^T X$

Diagonalize Covariance Matrix

$$\mathbf{C} = \mathbf{U} \mathbf{D} \mathbf{U}^T \quad \mathbf{U}^T = \mathbf{U}^{-1}$$

$$\mathbf{D} = \begin{bmatrix} \lambda_1 & & & & \\ & \lambda_2 & & & \\ & & 0 & & \\ & & & \lambda_n & \\ 0 & & & & 0 & & \\ & 0 & & & & 0 & \\ & & & & & & 0 \end{bmatrix} \quad \lambda_i \text{ are the sorted Eigenvalues of } \mathbf{C}$$

From linear algebra:

Since \mathbf{C} is positive and symmetrical \mathbf{U} is composed of the Eigenvectors of \mathbf{C} .

\mathbf{U} is a rotation which transforms the experiments into the new coordinate system. In this new coordinate system the covariance matrix is diagonal with the Eigenvalues λ_i on the diagonal and (since sorted) the first axis carries the most variance. The λ_i are the variances in the new coordinate system.

It can be shown that this definition is equivalent to the definition rotate so that highest variance is in first component second highest in second.

PCA as Eigenvalues

```
> co = t(X) %*% X
```

```
> co
```

```
      X_1  X_2  
X_1 16.112 4.82  
X_2  4.820 2.20
```

```
> eig = eigen(co)
```

```
> U = eig$vectors
```

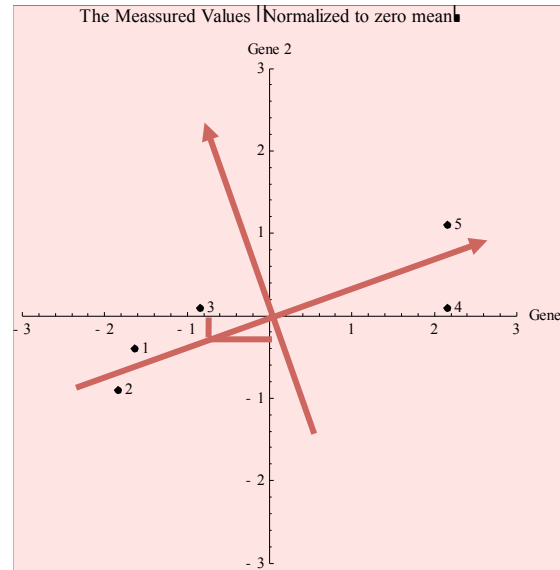
```
> X %*% U
```

```
      [,1]      [,2]  
[1,]  1.6846468 -0.1075419  
[2,]  2.0247206  0.3100102  
[3,]  0.7719022 -0.3460737  
[4,] -2.0914512  0.5490283  
[5,] -2.3898185 -0.4054229
```

```
> res = prcomp(X, scale. = FALSE)
```

```
> res$x
```

```
      PC1      PC2  
[1,] -1.6846468  0.1075419  
[2,] -2.0247206 -0.3100102  
[3,] -0.7719022  0.3460737  
[4,]  2.0914512 -0.5490283  
[5,]  2.3898185  0.4054229
```



Without calculation XTX

```
> U1 = svd(X)$v
```

```
> X %*% U
```

```
      [,1]      [,2]  
[1,]  1.6846468 -0.1075419  
[2,]  2.0247206  0.3100102  
[3,]  0.7719022 -0.3460737  
[4,] -2.0914512  0.5490283  
[5,] -2.3898185 -0.4054229
```

PCA as Eigenvalues

```
> res = prcomp(X, scale. = FALSE)
> Z = res$x          # The PCA-Transformed values
> res$sdev^2
[1] 4.4046904 0.1733096
> cov(X)
      X_1  X_2
X_1 4.028 1.205
X_2 1.205 0.550
> cov(Z)
      PC1          PC2
PC1 4.404690e+00 6.911247e-16
PC2 6.911247e-16 1.733096e-01
> cov(X)[1,1] + cov(X)[2,2]
[1] 4.578
> cov(Z)[1,1] + cov(Z)[2,2]
[1] 4.578
```

- After the PCA the covariance matrix is diagonal
- In the squared diagonal are the explained variances

PCA in R

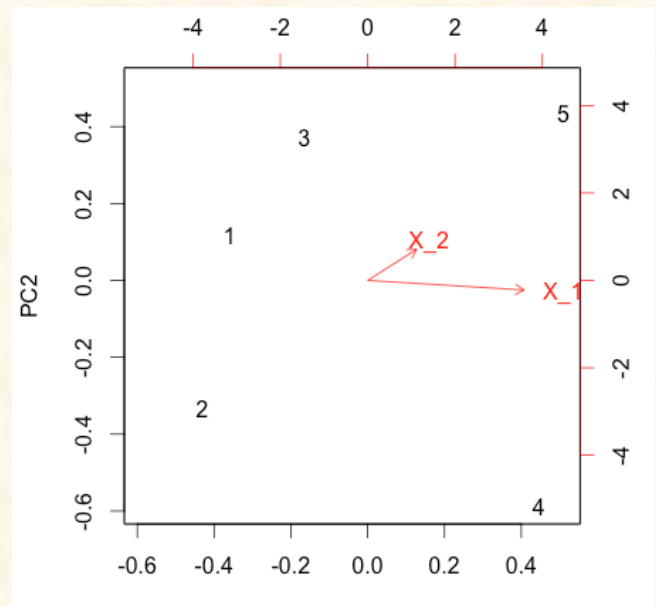
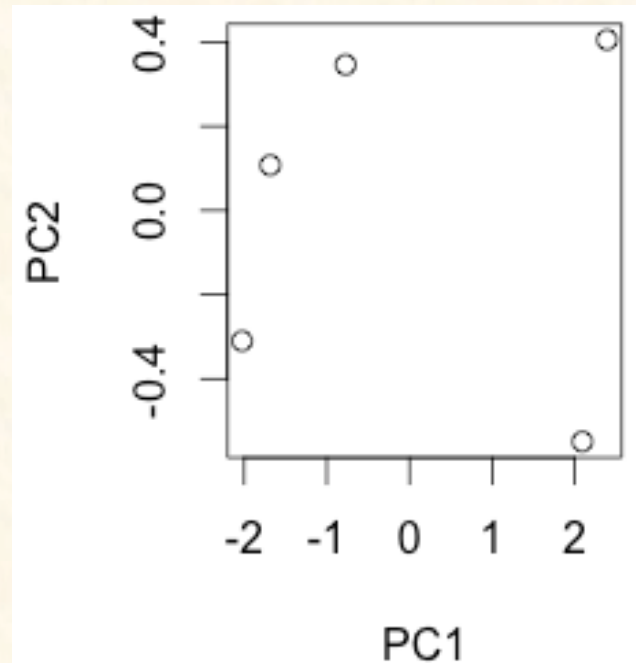
```
> res = prcomp(X, scale. = FALSE)
> plot(res$x)
> biplot(res, scale = TRUE)
> res$rotation #Loadings
```

	PC1	PC2
X_1	0.9544511	-0.2983673
X_2	0.2983673	0.9544511

```
> summary(res)
```

Importance of components:

	PC1	PC2
Standard deviation	2.0987	0.41630
Proportion of Variance	0.9621	0.03786
Cumulative Proportion	0.9621	1.00000



Standardisieren, wenn die beobachteten Messgrößen in verschiedenen Einheiten (cm, m, kg...) vorliegen. Auch bei sehr grossen Unterschieden in den Varianzen der Variablen sollte man über deren Ursache nachdenken und ggf. auch dann standardisieren, wenn alle Variablen dieselbe Einheit haben.

Nicht standardisieren, wenn die gemessenen Variablen vergleichbar sind bezüglich ihrer Einheiten und ihrer Variabilität.

Example

```
> head(X)
```

	Murder	Assault	UrbanPop	Rape
Alabama	13.2	236	58	21.2
Alaska	10.0	263	48	44.5
Arizona	8.1	294	80	31.0
Arkansas	8.8	190	50	19.5
California	9.0	276	91	40.6
Colorado	7.9	204	78	38.7

A data frame with 50 observations on 4 variables.

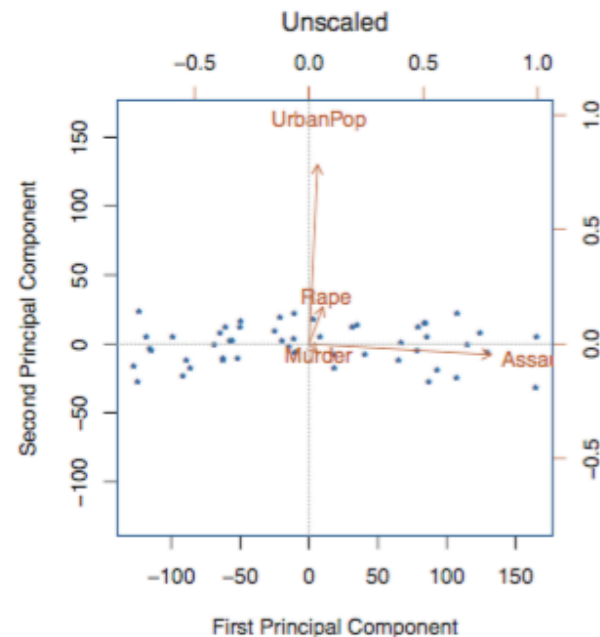
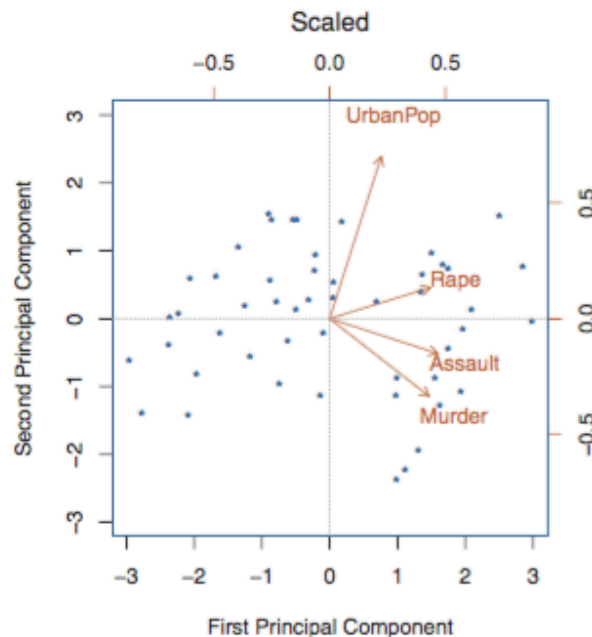
[,1] Murder numeric Murder arrests (per 100,000)

[,2] Assault numeric Assault arrests (per 100,000)

[,3] UrbanPop numeric Percent urban population

[,4] Rape numeric Rape arrests (per 100,000)

Would you scale?



Second Interpretation of the PCA

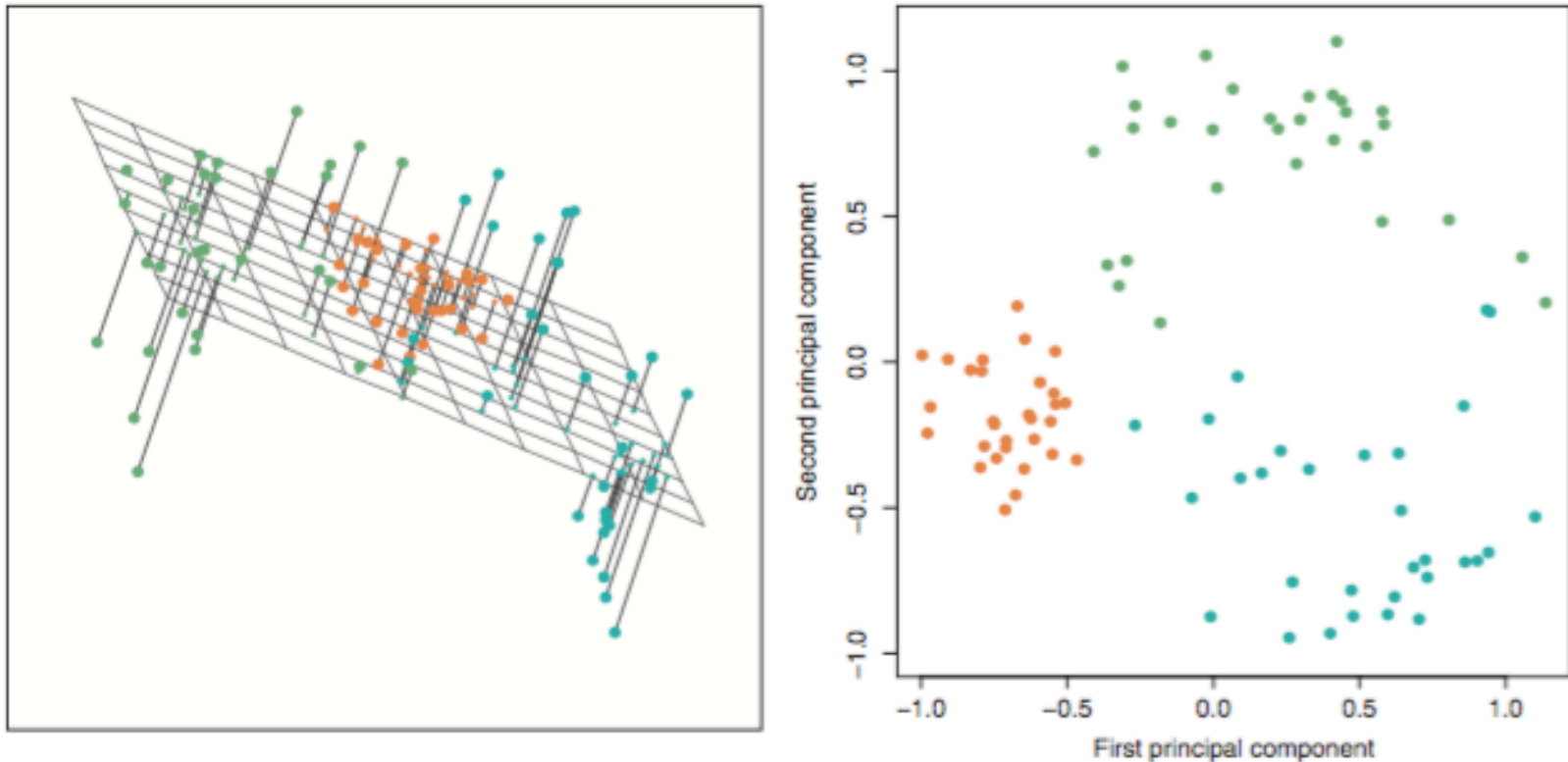


FIGURE 10.2. *Ninety observations simulated in three dimensions. Left: the first two principal component directions span the plane that best fits the data. It minimizes the sum of squared distances from each point to the plane. Right: the first two principal component score vectors give the coordinates of the projection of the 90 observations onto the plane. The variance in the plane is maximized.*