

Elements of Machine Learning & Data Science

Time Series, Data Quality, and Preprocessing

Lecture 11

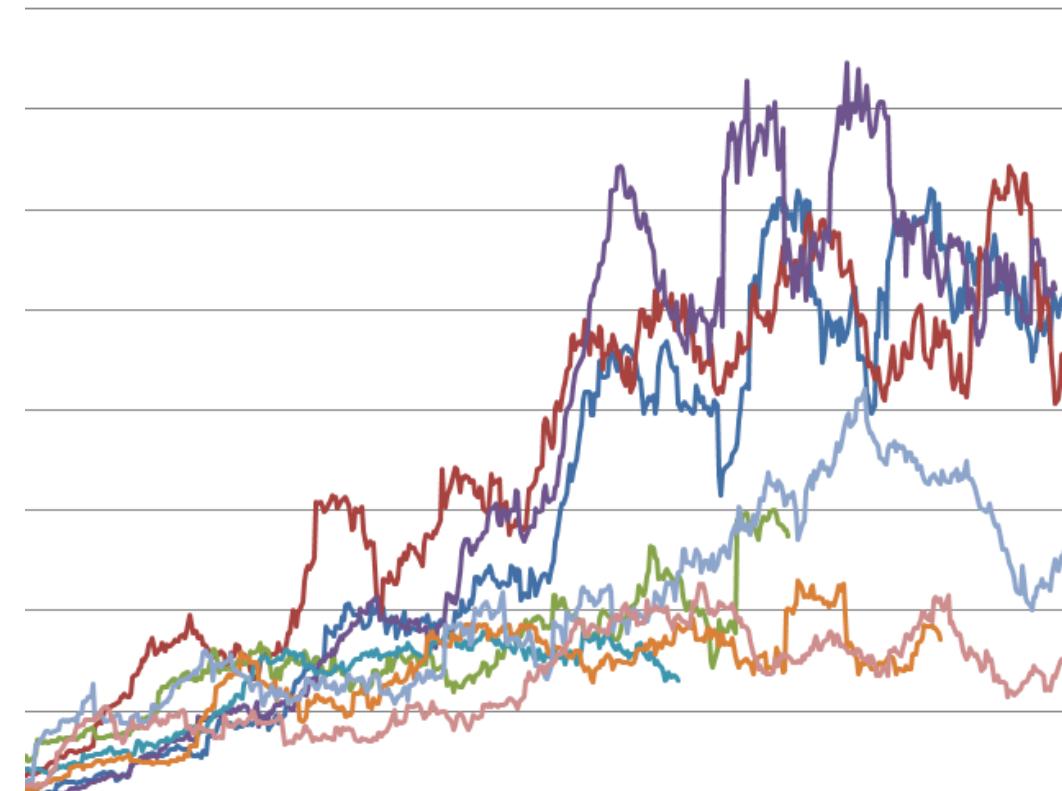
Prof. Wil van der Aalst

Marco Pegoraro, M.Sc.

Tsunghao Huang, M.Sc.

This lecture: Two Main Topics

Time series analysis



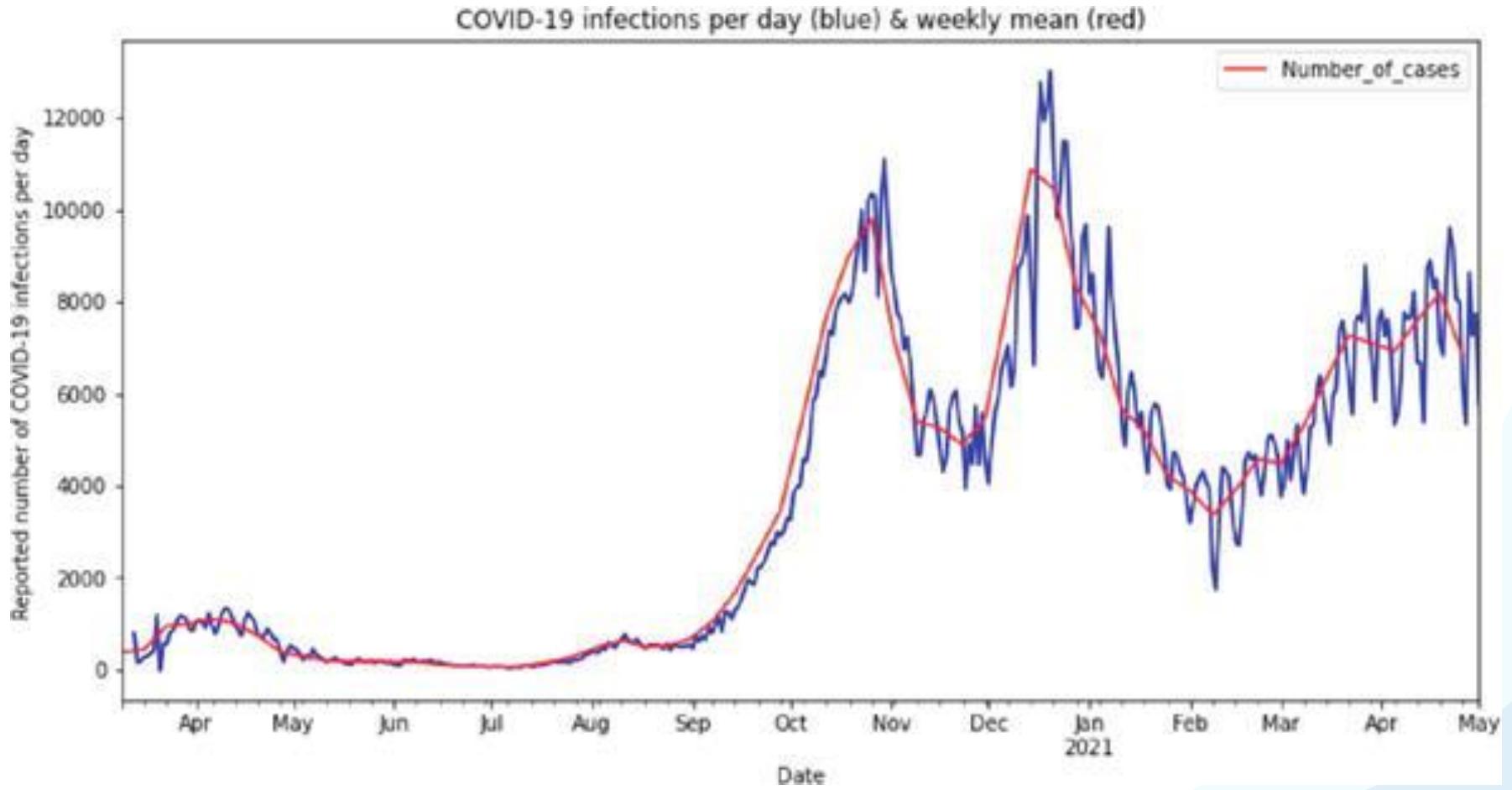
Data Quality, and Preprocessing



Generated using DALL E3

Time Series

1. Introduction
2. Analysis
3. Forecasting



Taken from Time Series Forecasting on COVID-19 Data 10.5772/intechopen.104920

Temporal Data – Discrete Timestamped Events

Time-stamp	f_1	f_2	f_3	f_4	...	f_D
t_1						
t_2						
t_3						
t_4						
t_5						
...						

Every instance happened at a specific time



- **Sequence mining**
(seen before)
- **Process mining**
(later lecture)
- **Time series**
(this lecture)

Temporal Data – Discrete Timestamped Events

Time series data

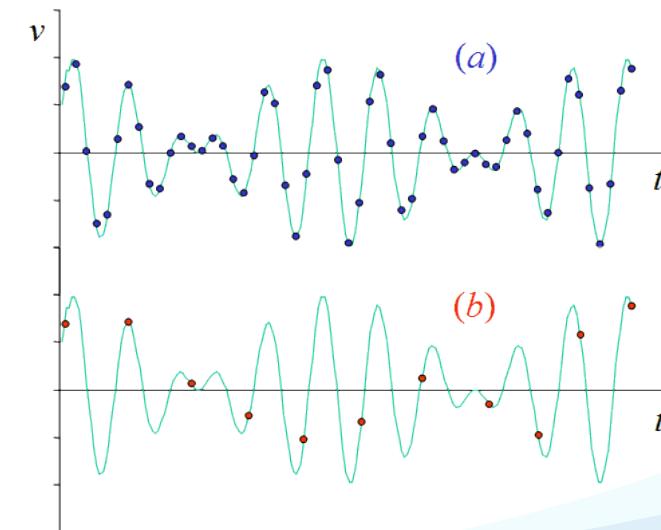
Time-stamp	f_1	f_2	f_3	f_4	...	f_D
t_1						
t_2						
t_3						
t_4						
t_5						
...						

Intervals are typically equal

numerical

Times series analysis

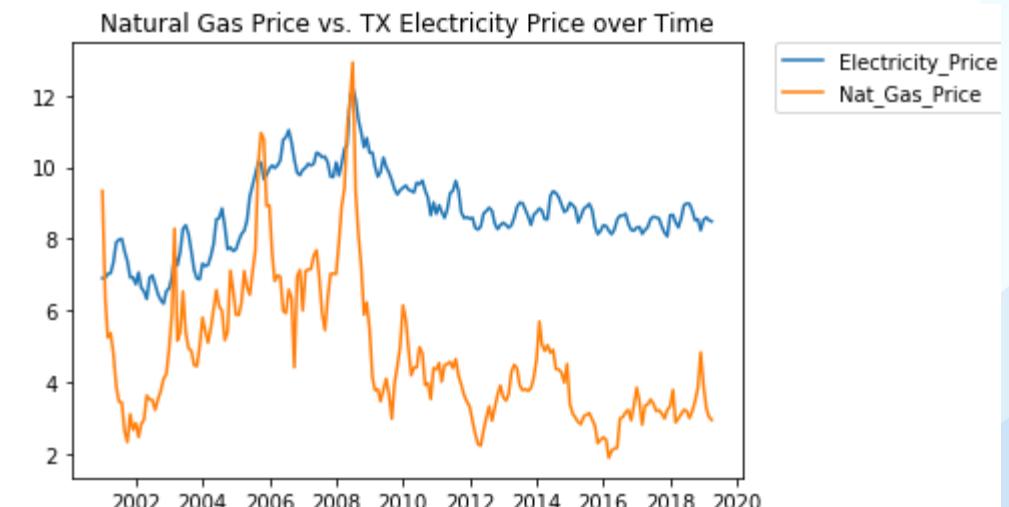
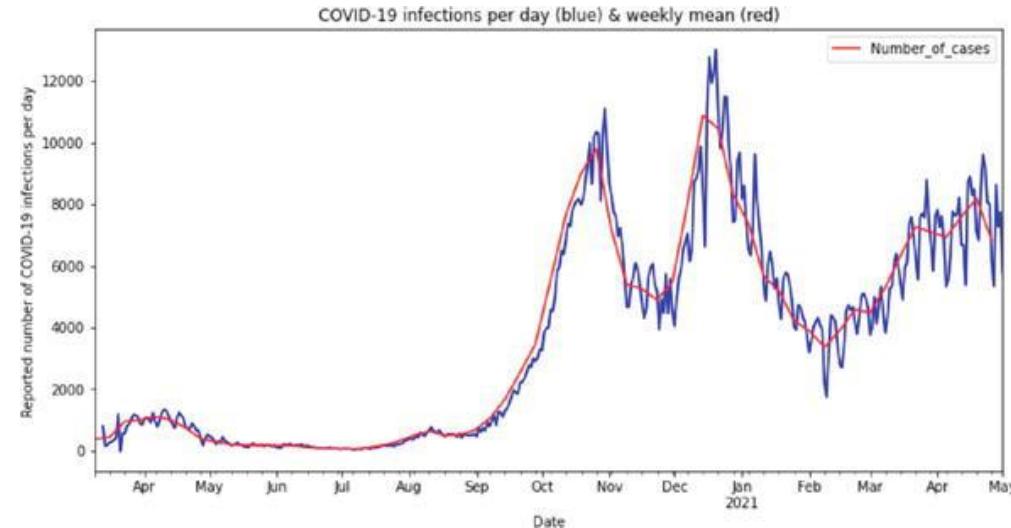
- Equidistant timestamps (determined by sampling rate)
- Features are numerical



Sampling rate
millisecond,
second, hour,
day, week, etc.

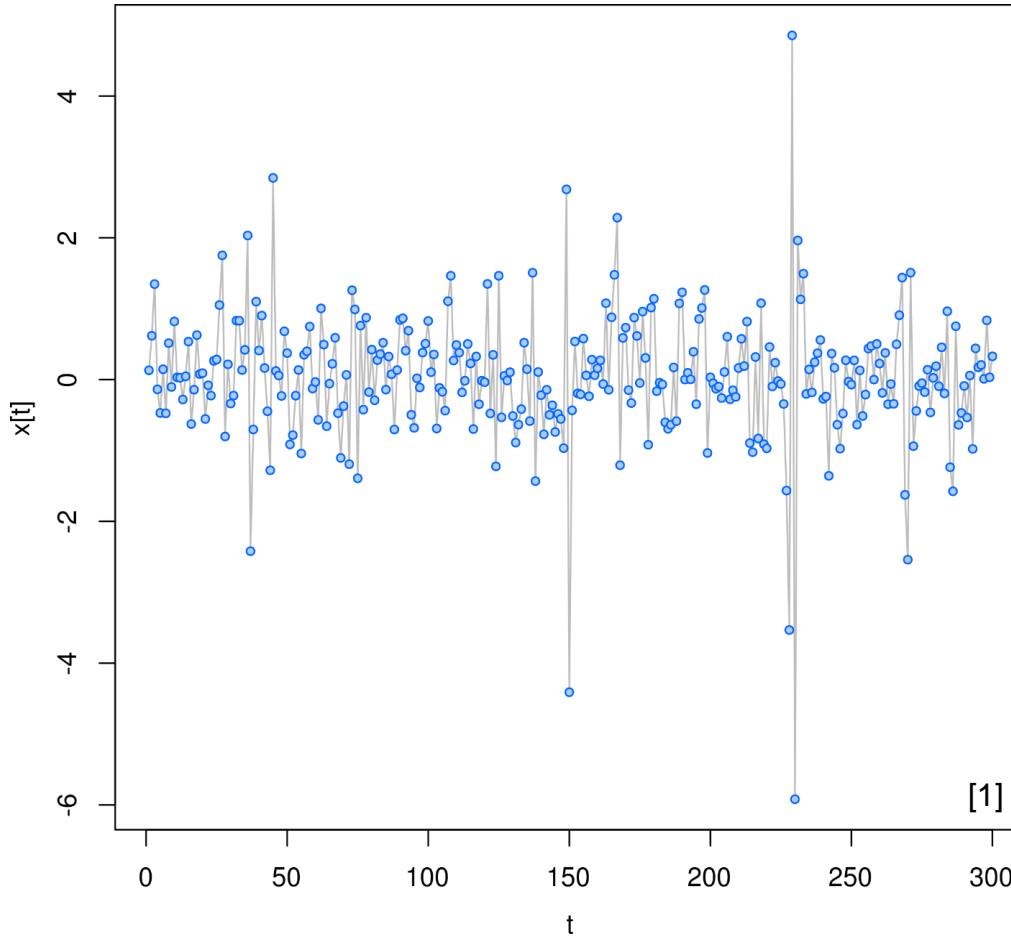
Time Series Analysis Has Numerous Applications

- Health (e.g., Covid)
- Finance (e.g., stock market)
- Inventory management
- Marketing
- Energy systems
- Climate change
- Political polls
- Etc.



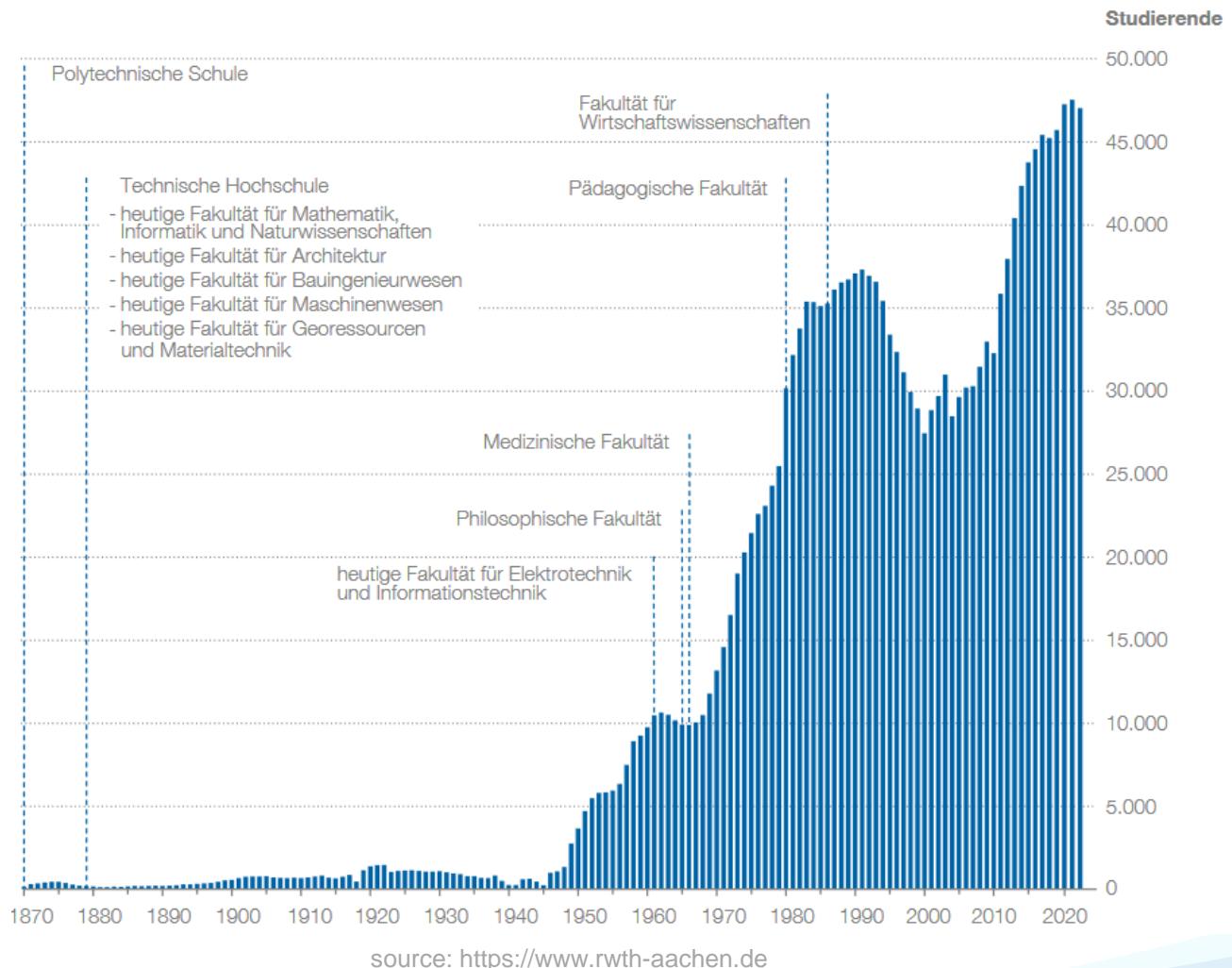
Let's Start With Univariate Time Series

Time-stamp	x
00:01	0.9
00:02	0.5
00:03	1.6
00:04	-0.8
00:05	-0.4
00:06	0.1
00:07	-0.3
00:08	-0.4
...	...



Time Series – Some More Examples

- Number of RWTH students



Time Series - Some More Examples

- Finance: stock price

Tesla, Inc.



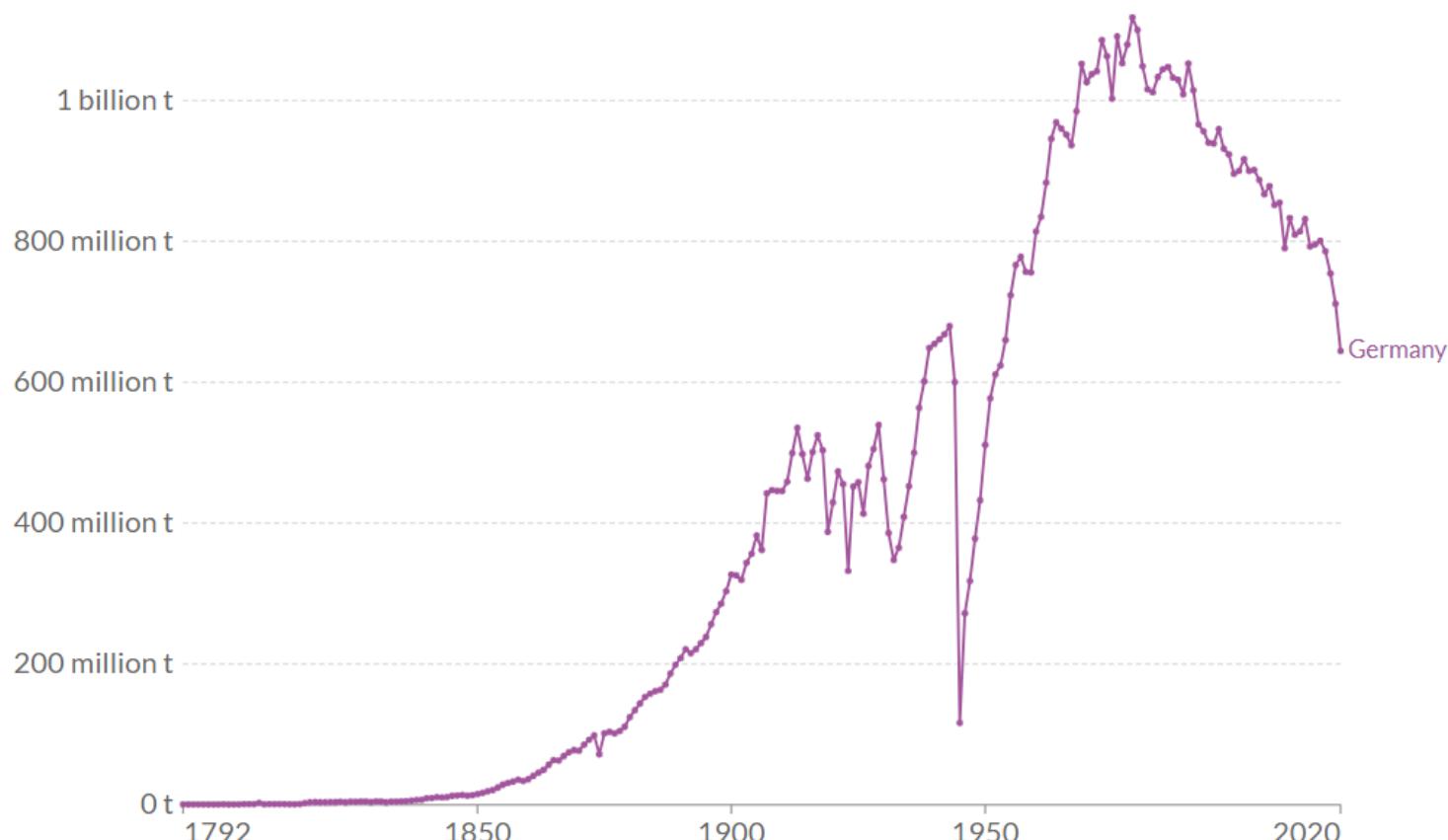
Time Series - Some More Examples

- Climate: CO₂

Annual CO₂ emissions

Carbon dioxide (CO₂) emissions from the burning of fossil fuels for energy and cement production. Land use change is not included.

Our World
in Data



Source: Global Carbon Project

OurWorldInData.org/co2-and-other-greenhouse-gas-emissions/ • CC BY

Time Series

1. Introduction
2. **Analysis**
3. Forecasting

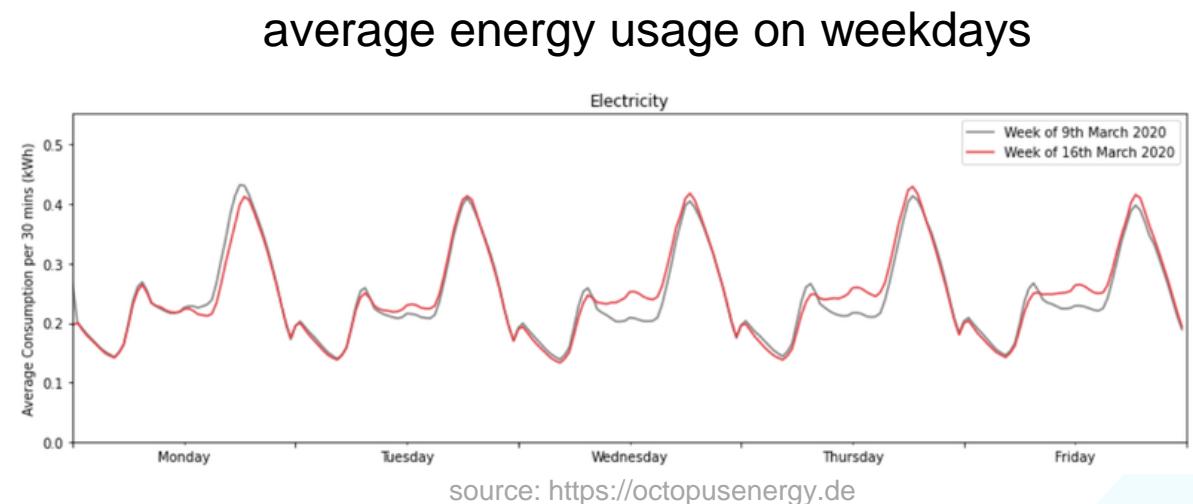


Time Series Patterns

- **Trend:** a time series exhibits a long-term increase or decrease in the data.
- **Seasonal:** a times series is affected by a fixed and known frequency (e.g., month, weekday).
- **Cyclic:** a time series exhibits rise and falls that are not of a fixed frequency (e.g., economic fluctuations).



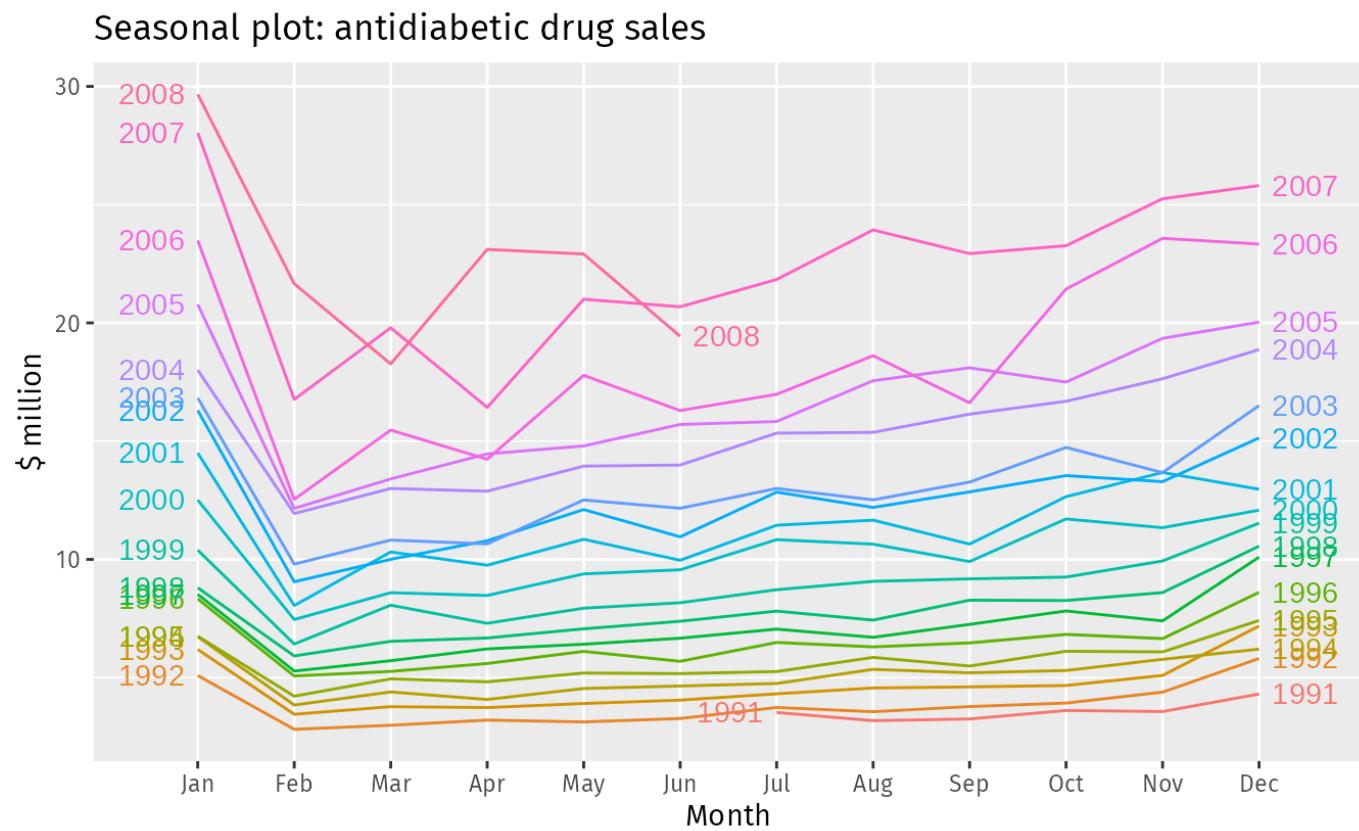
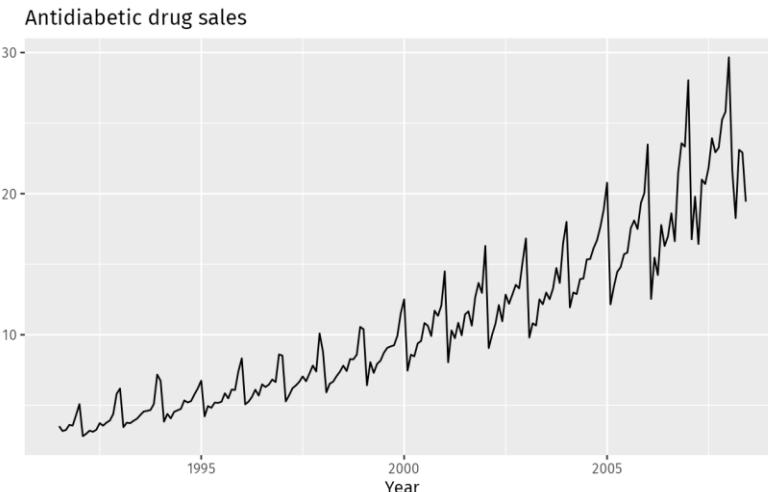
trend and cyclic patterns



seasonal patterns

Seasonal Plot

- Similar to a time plot, but the observations are plotted against a season.
- Easier to observe the underlying seasonal patterns.



Autocorrelation

- The linear relationship between lagged values of a time series.

$$r_k = \frac{\sum_{t=k+1}^T (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^T (y_t - \bar{y})^2}$$

The length of the time series

The coefficient between y_t and y_{t-k}

Lag

Average of observation y

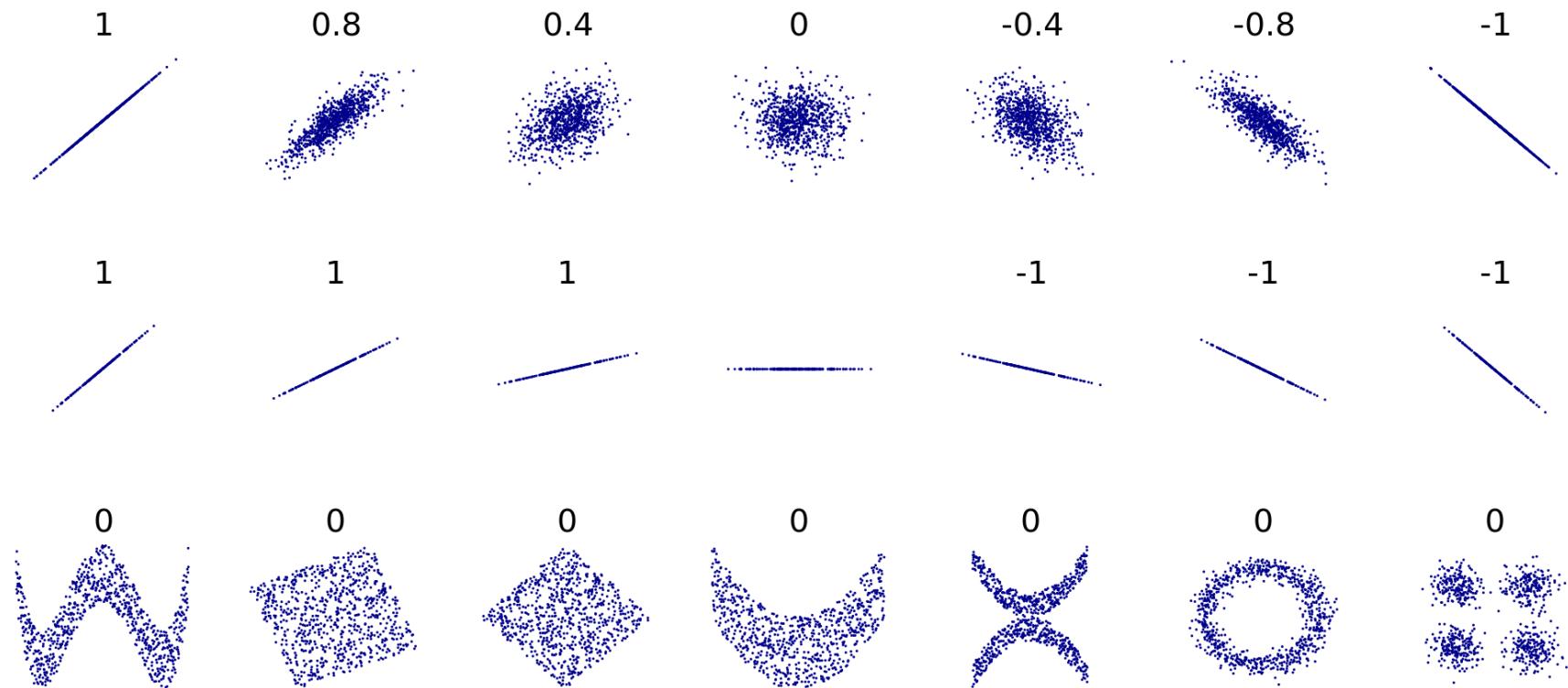
Observation at time t

Understanding Autocorrelation (1/2)

Sample correlation coefficient

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

Recall: the value of a correlation coefficient ranges between -1 and $+1$.



Understanding Autocorrelation (2/2)

Sample correlation coefficient

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

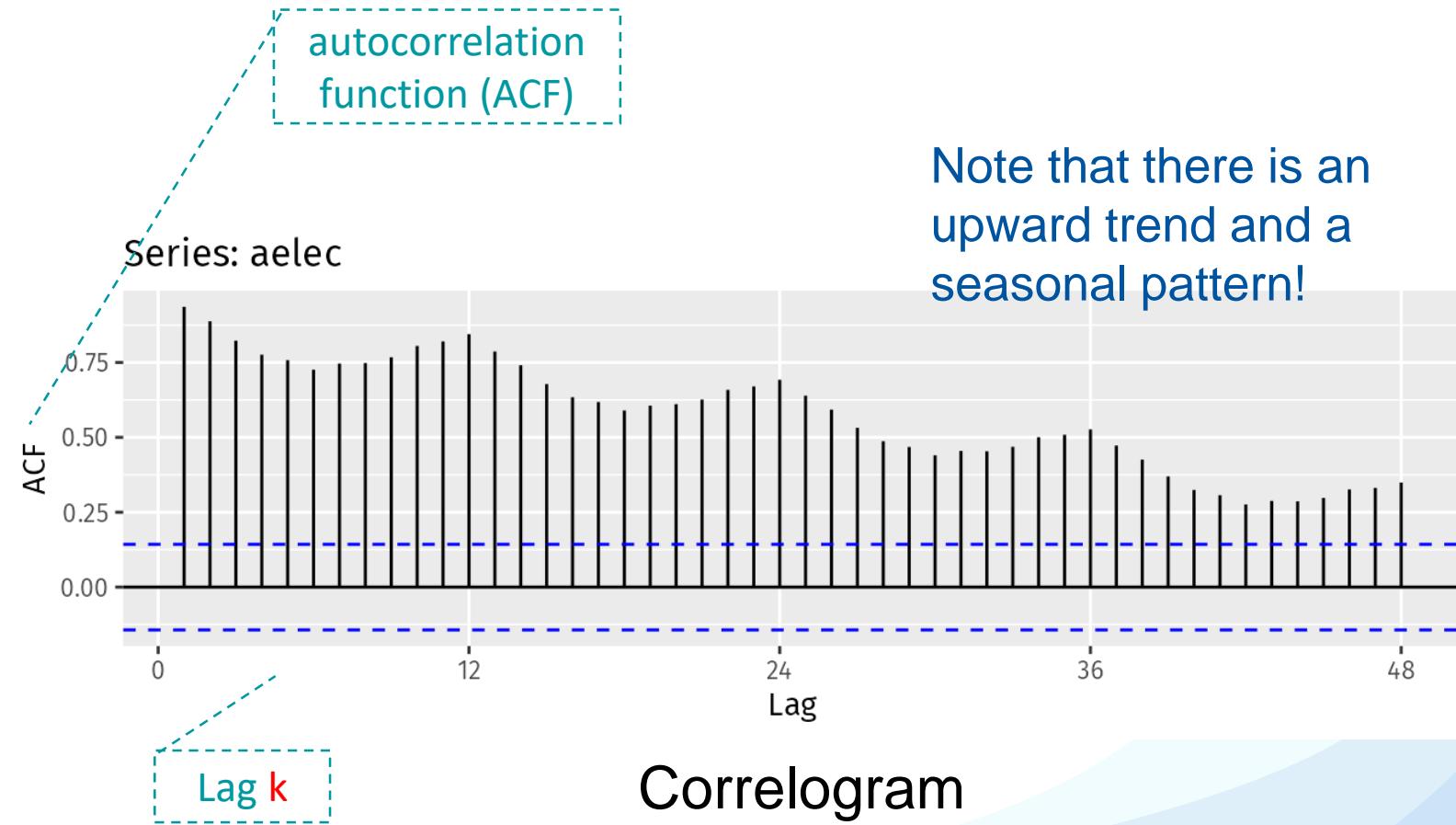
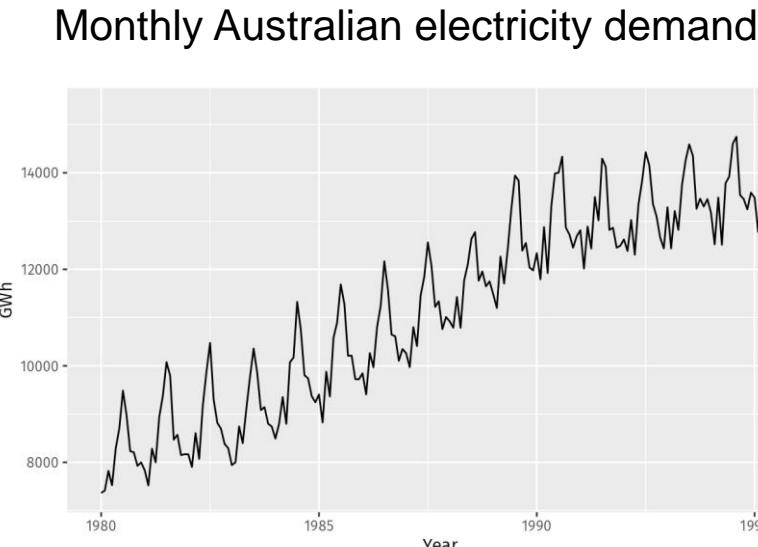
Autocorrelation

$$r_k = \frac{\sum_{t=k+1}^T (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^T (y_t - \bar{y})^2}$$

There is just one sample mean \bar{y}
 $(\bar{x} = \bar{y})$ and typically $T \gg k$.

Correlogram

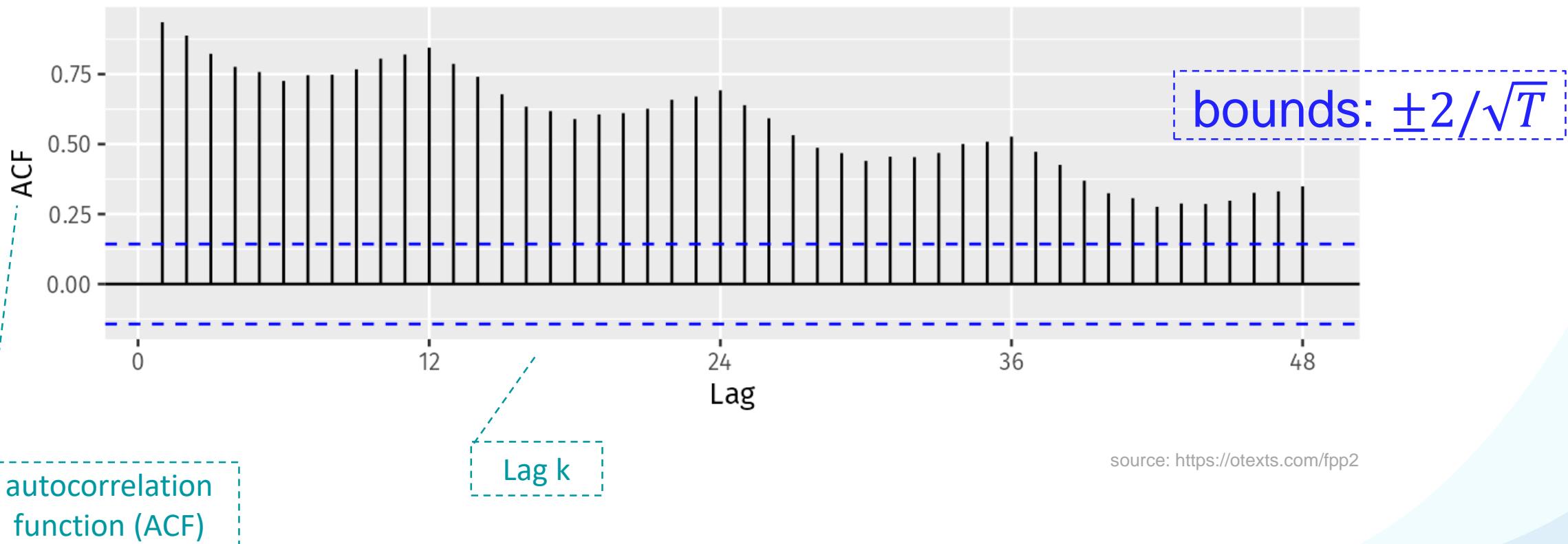
- A plot showing the autocorrelation coefficients between lagged values.



Correlogram

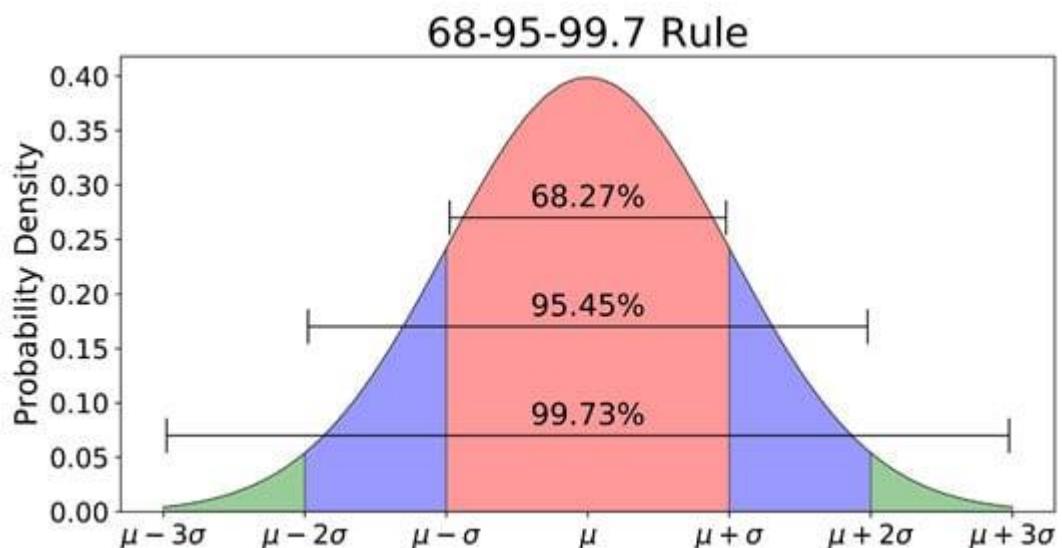
- Bounds: $\pm 2/\sqrt{T}$, where T is the number observations in the time series.

Series: aelec



Correlated or Not?

- Assume independence of subsequent observations.
- Under this assumption, the autocorrelation coefficient r_k follows (approximately!) a **normal distribution with a mean of 0 and a variance of $1/T$** , i.e., $\mu = 0$ and $\sigma = 1/\sqrt{T}$.
- This means that approximately 68% of the autocorrelation coefficients fall within $[-\sigma, \sigma]$, 95% of the autocorrelation coefficients fall within $[-2\sigma, 2\sigma]$, and 99.5% of the autocorrelation coefficients fall within $[-3\sigma, 3\sigma]$,



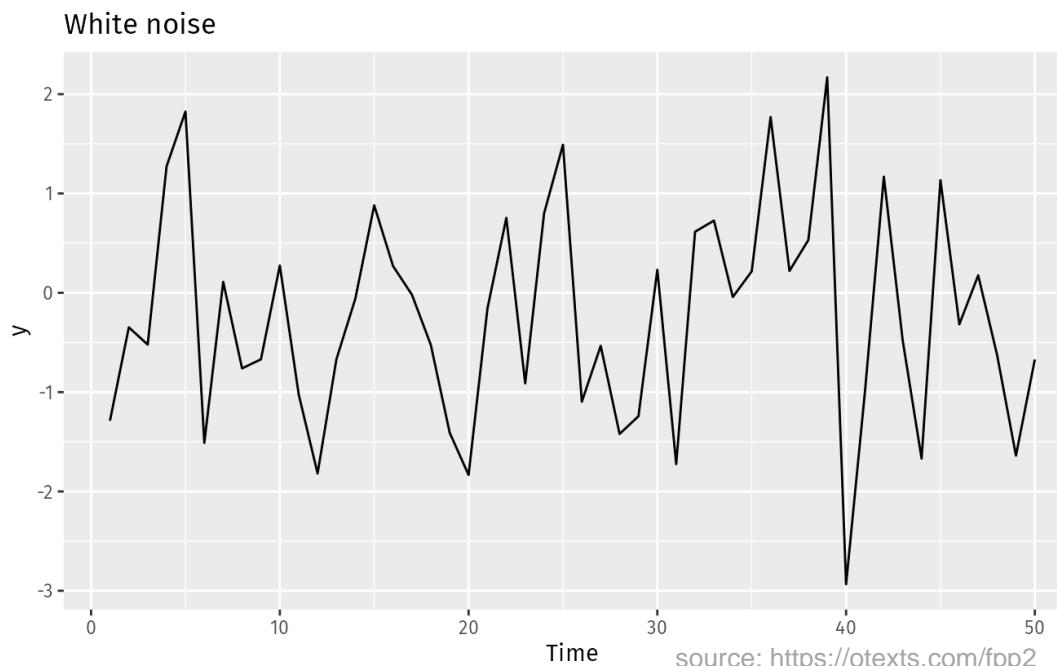
95% of the autocorrelation coefficients fall within $[-2/\sqrt{T}, 2/\sqrt{T}]$

Therefore, values outside of this interval are unlikely (less than 5%) under the independence assumption.

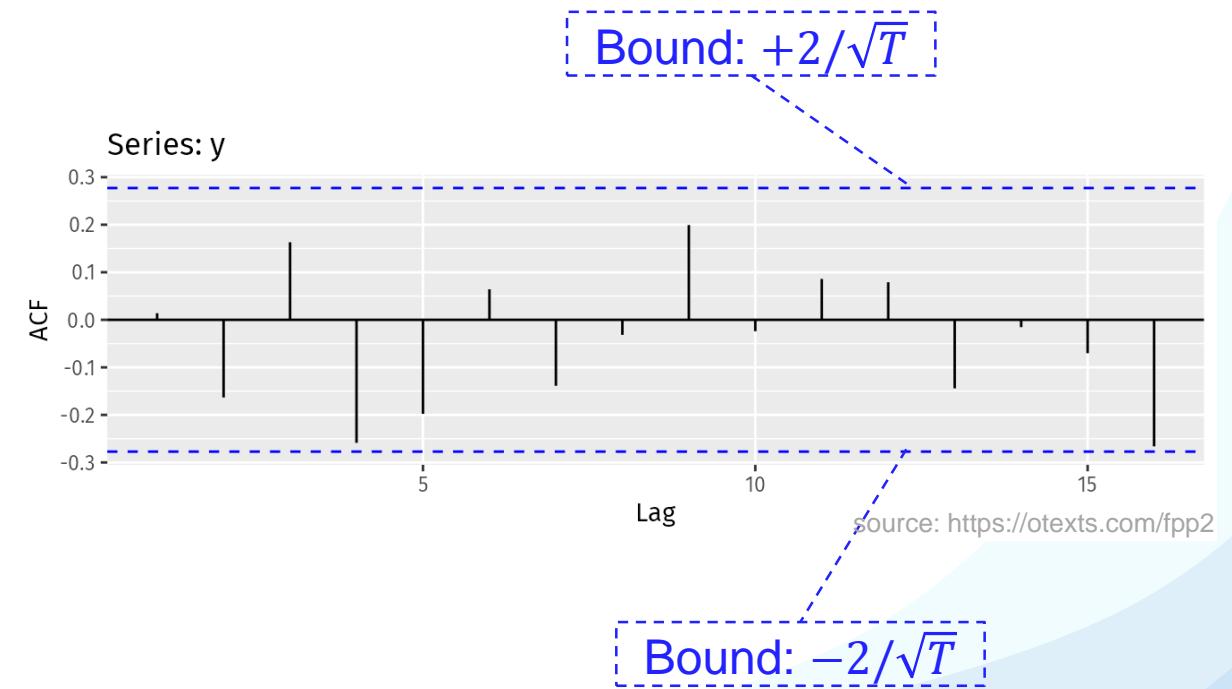
$$r_k = \frac{\sum_{t=k+1}^T (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^T (y_t - \bar{y})^2}$$

White Noise

- Time series consisting of **independent** observations are called **white noise**.
- Under this assumption there is just a 5% chance that the autocorrelation coefficient will fall outside of the bounds $[-2/\sqrt{T}, 2/\sqrt{T}]$ by chance.
- Spikes outside these bounds, suggest that there may be a correlation.



white noise randomly generated



Time Series Decomposition

- Various patterns coexist
- May be helpful to split a time series into several components
- Additive decomposition

$$y_t = S_t + T_t + R_t$$

The diagram illustrates the additive decomposition of a time series. A horizontal dashed line represents the total time series y_t . Three vertical dashed lines extend downwards from this line to the components: the top dashed line is labeled "trend-cycle component", the bottom-left dashed line is labeled "seasonal component", and the bottom-right dashed line is labeled "remainder component".

- Alternative: multiplicative decomposition

$$y_t = S_t \times T_t \times R_t$$

Time Series Decomposition

Estimating the trend-cycle component T_t : moving average

- The estimate of the T_t is obtained by averaging values of the time series within k periods of t .
- A moving average of order $m = 2k + 1$ is called **m -MA**.

$$\hat{T}_t = \frac{1}{m} \sum_{j=-k}^k y_{t+j}$$

$$m = 2k + 1$$

Year	Sales (GWh)	5-MA
1989		2354.34
1990		2379.71
1991		2318.52
1992		2468.99
1993		2386.09
1994	2569.47	2424.56
1995	2575.72	2463.76
1996	2762.72	2552.60
1997	2844.50	2627.70
1998	3000.70	2750.62
1999	3108.10	2858.35
2000	3357.50	3014.70
2001	3075.70	3077.30
2002	3180.60	3144.52
2003	3221.60	3188.70
2004	3176.20	3202.32
2005	3430.60	3216.94
2006	3527.48	3307.30
2007	3637.89	3398.75
2008	3655.00	3485.43

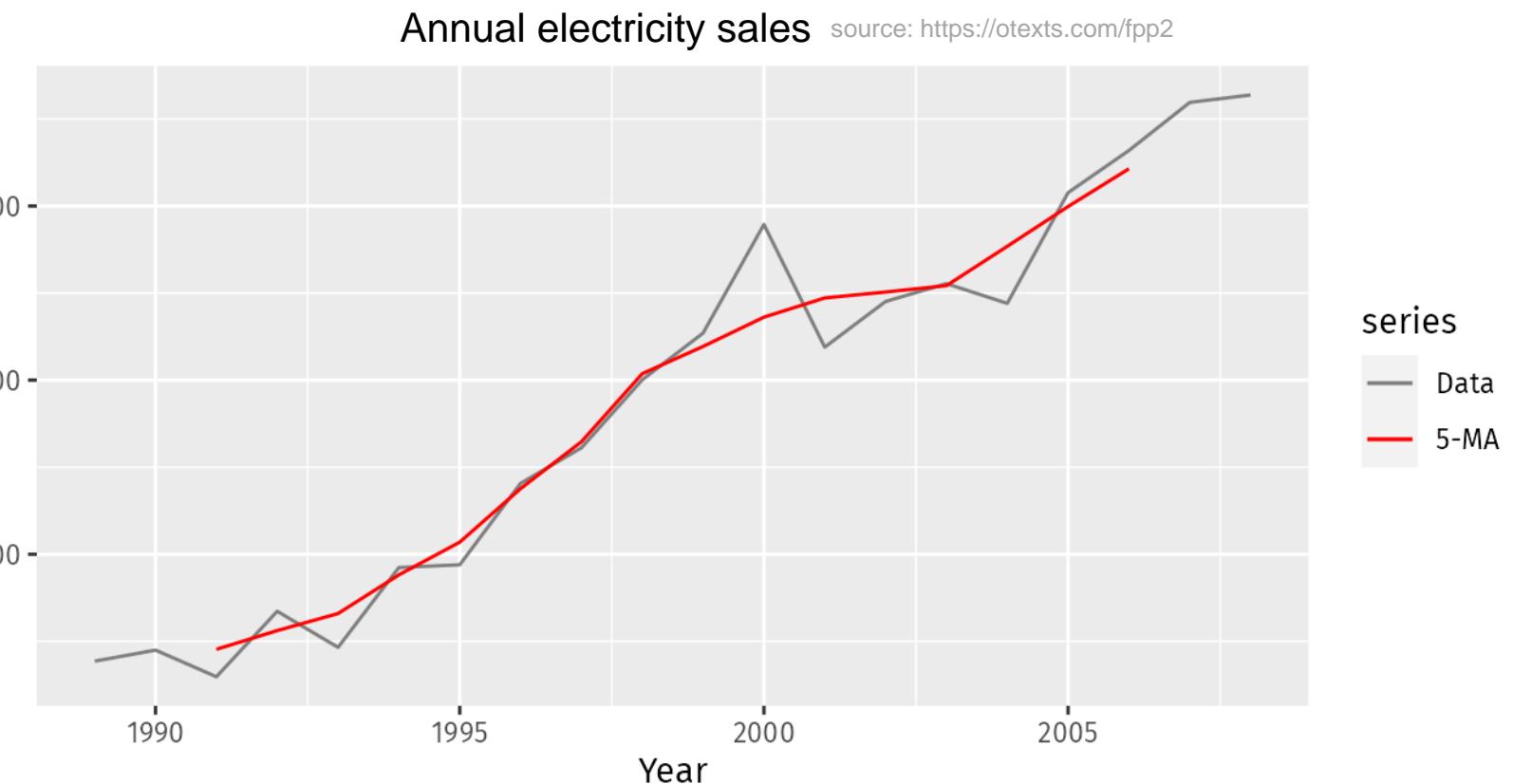
$$\begin{aligned} k &= 2 \\ m &= 5 \end{aligned}$$

Time Series Decomposition

Estimating the trend-cycle component T_t : moving average

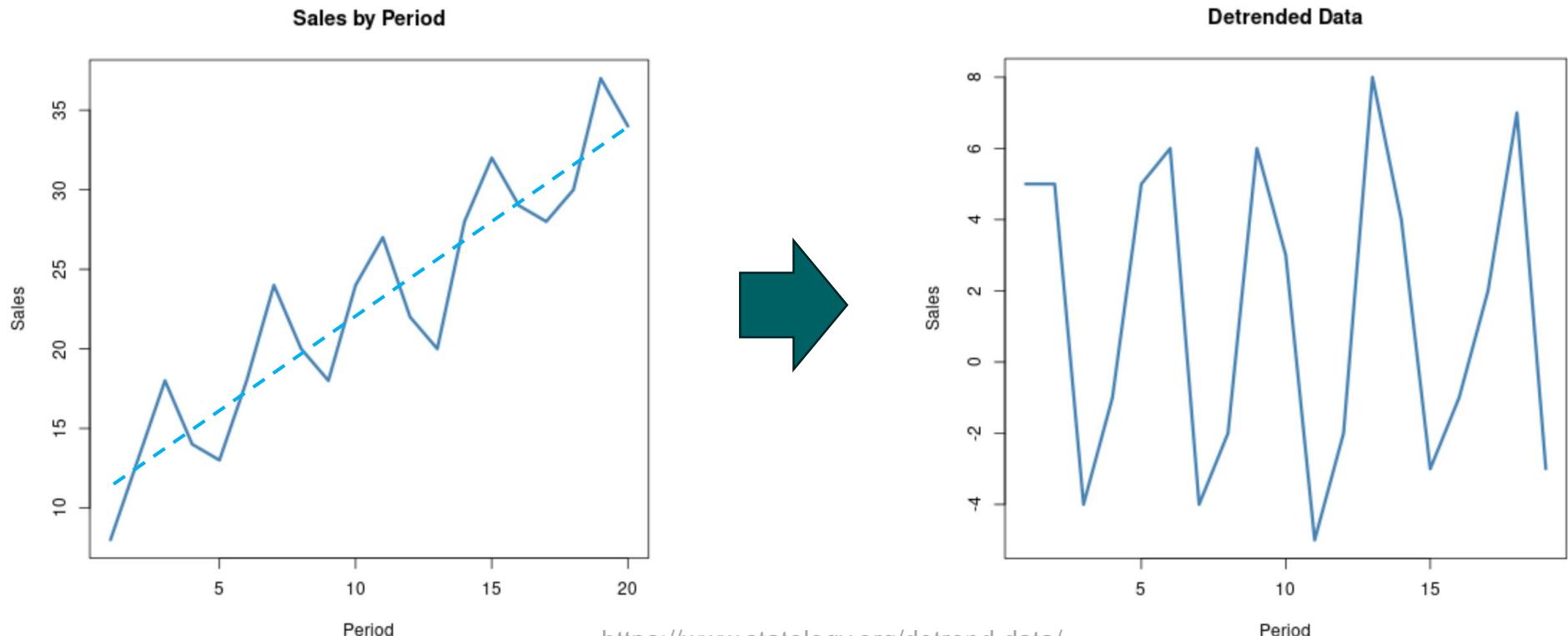
- The estimate of the T_t is obtained by averaging values of the time series within k periods of t .
- A moving average of order m is called **m -MA**

$$\hat{T}_t = \frac{1}{m} \sum_{j=-k}^k y_{t+j}$$



Time Series Decomposition

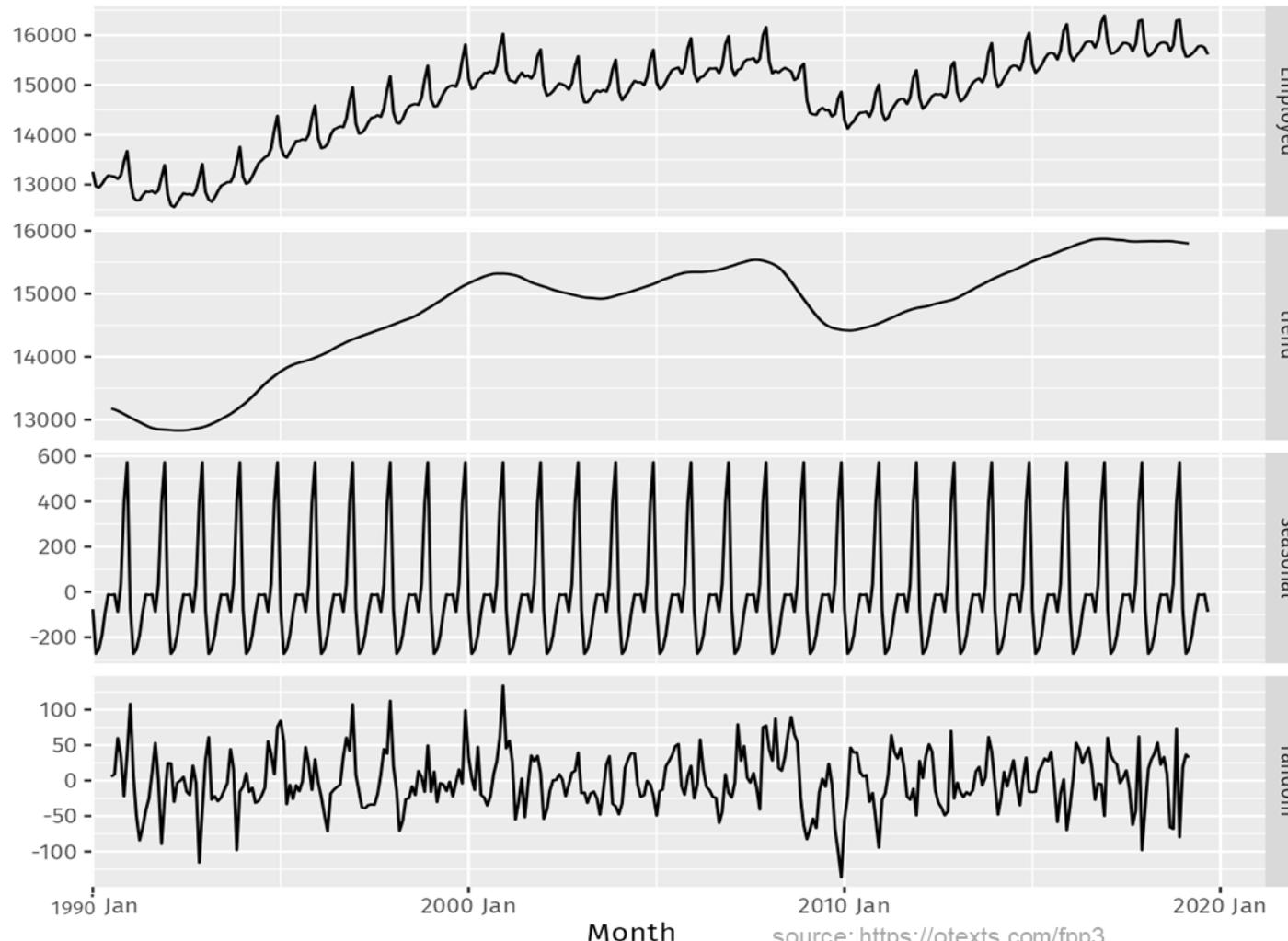
- To **detrend** time series data means to remove an underlying trend in the data.
- This way one can see subtrends in the data that are seasonal or cyclical.
- Calculate the **detrend series**: $y_t - \hat{T}_t$.
- Similarly, one can detrend for seasonal effects.



Time Series Decomposition

Classical additive decomposition of total US retail employment

Employed = trend + seasonal + random



$$y_t = S_t + T_t + R_t$$

trend-cycle

$$T_t$$

seasonal

$$S_t$$

remainder

$$R_t$$

Time Series Decomposition: Example



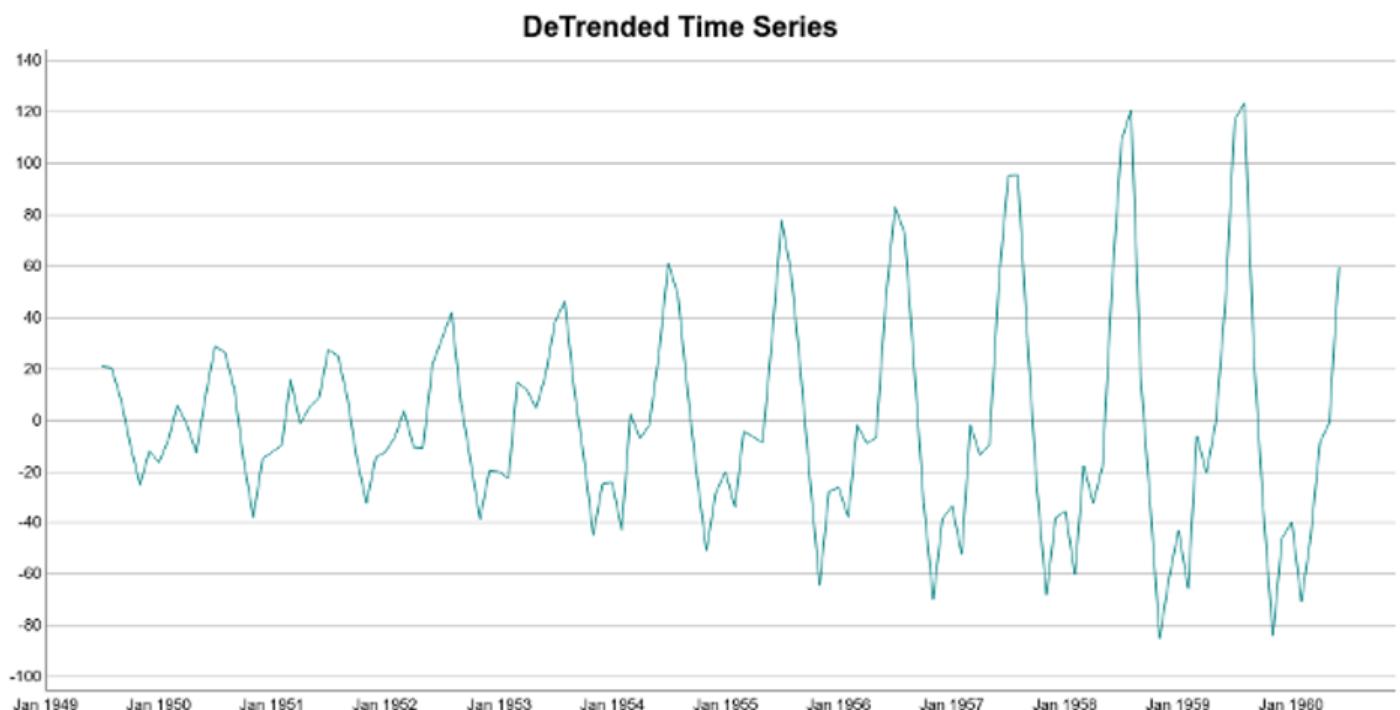
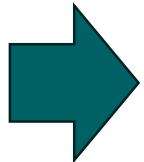
$$y_t = S_t + T_t + R_t$$

Example taken from <https://www.encora.com/insights/a-visual-guide-to-time-series-decomposition-analysis>

Time Series Decomposition: Example

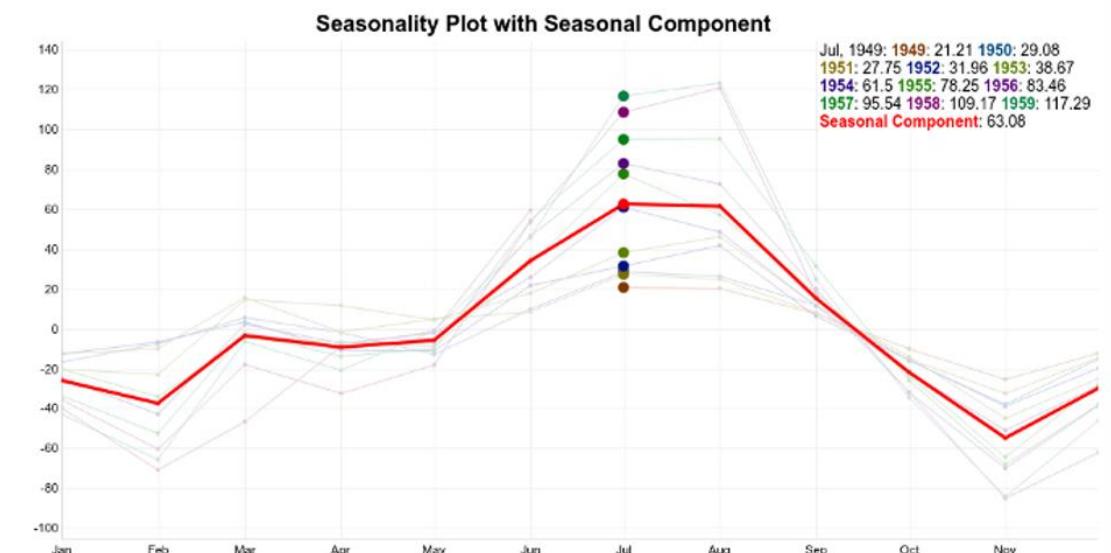
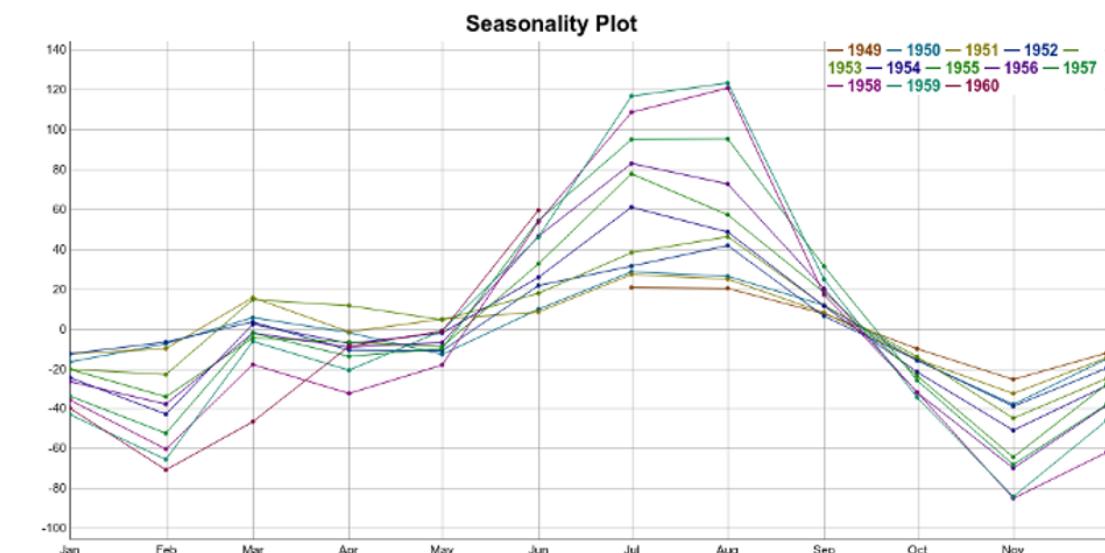


$$y_t = S_t + T_t + R_t$$

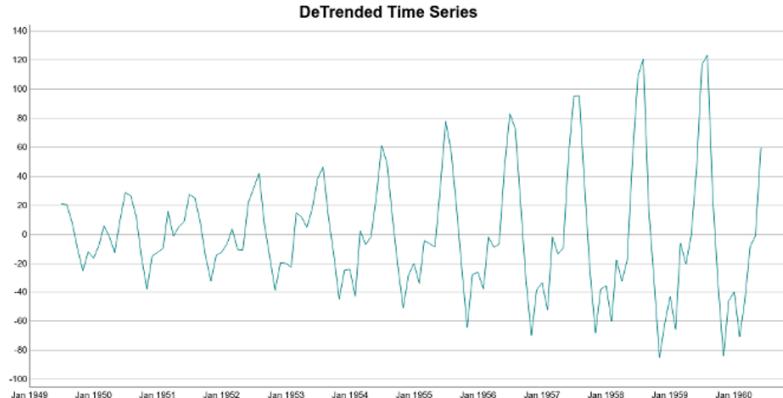


$$y'_t = S_t + R_t = y_t - T_t$$

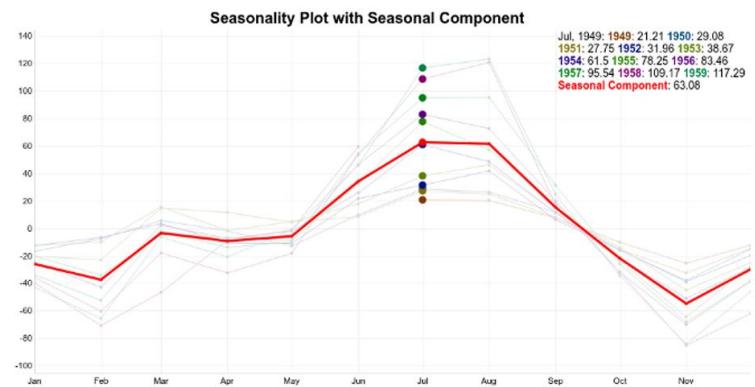
Time Series Decomposition: Example

 S_t

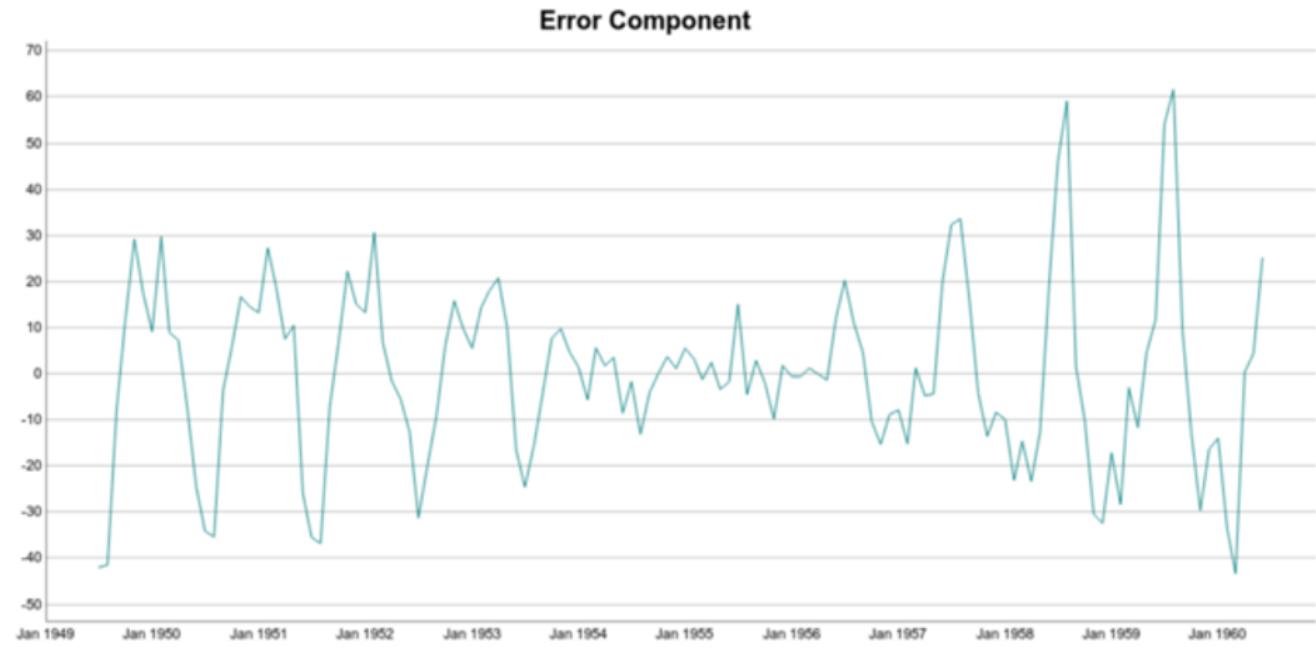
Time Series Decomposition: Example



$$y_t' = S_t + R_t$$

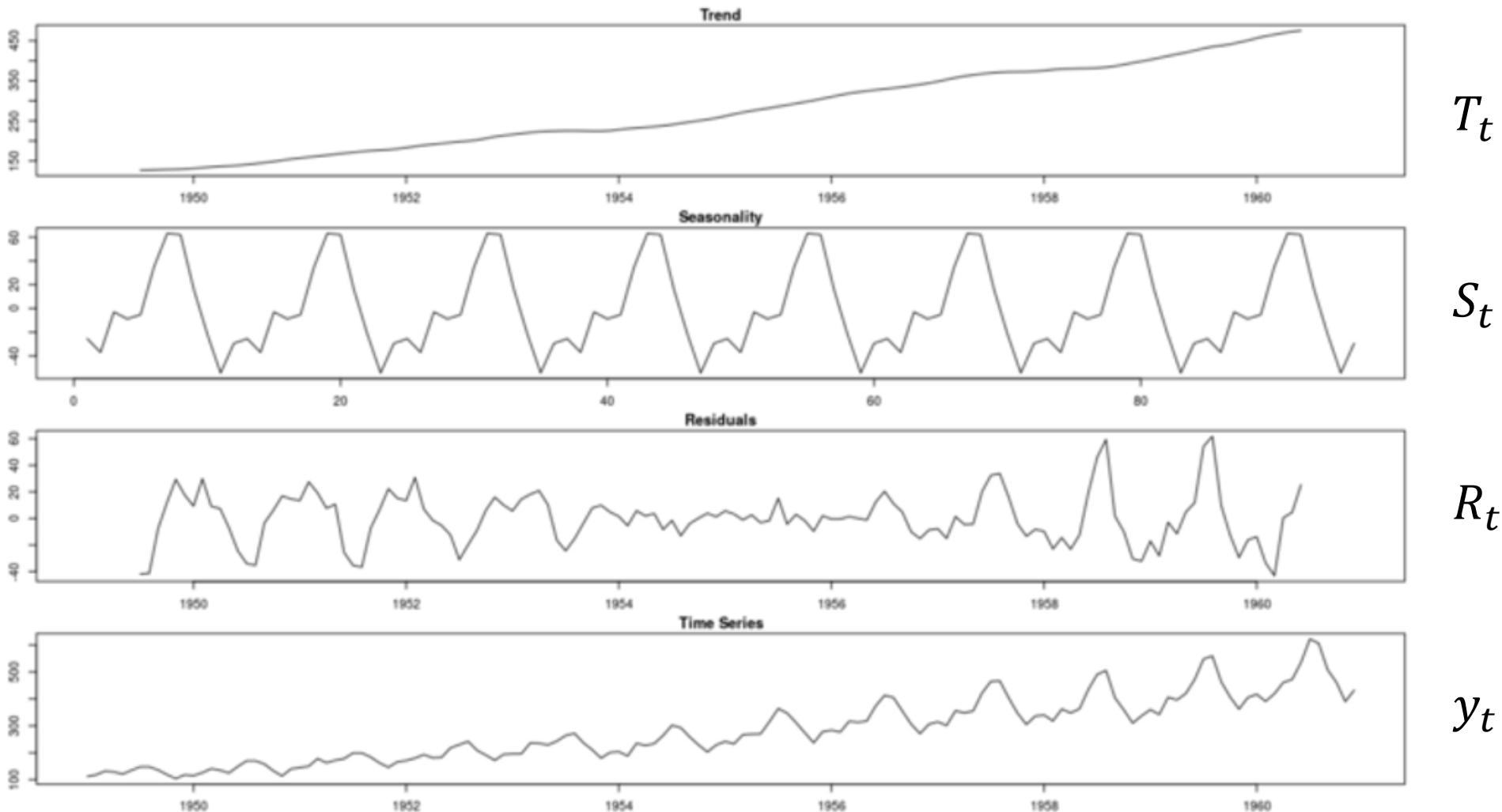


S_t



$$y_t'' = R_t = y_t - (S_t + T_t)$$

Time Series Decomposition: Example



Time Series

1. Introduction
2. Analysis
3. **Forecasting**



Autoregressive (AR) Models

- An Autoregressive (AR) model is a regression of the variable against itself.
- The variable of interest is forecasted using a linear combination of past values of the variable.

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t$$

Diagram illustrating the components of the AR(p) model:

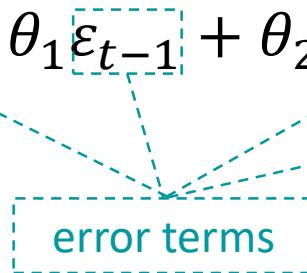
- coefficient**: Labels the term ϕ_1 .
- past value of y at time $t - 1$** : Labels the term y_{t-1} .
- error term**: Labels the term ε_t .

- It is referred to as a $AR(p)$ model, an Autoregressive (AR) model of order p .

ϕ is “phi”

Moving Average (MA) Models

- A Moving Average (MA) model is a regression of the past errors.
- The variable of interest is forecasted using a linear combination of past forecast errors.

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}$$


- Note that given $c, \theta_1, \theta_2, \dots, \theta_q$, it is possible to compute $\epsilon_1, \epsilon_2, \dots, \epsilon_T$.
- It is referred to as a MA(q) model, a Moving Average (MA) model of order q .

θ is “theta”

Moving Average (MA) Models: Compute Errors

An example of an MA(1) Model

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

10

0.5

assumed to be given, e.g., through EM

Time	Forecasted Values (\hat{y}_t)	Error at time t (ε_t)	Actual Values (y_t)
1	10		9
2	$9.5 = 10 + 0.5(-1)$		11.5
3	$11 = 10 + 0.5(2)$		10
4	$9.5 = 10 + 0.5(-1)$		10.5
5	$11 = 10 + 0.5(2)$		10
6	$9 = 10 + 0.5(-1)$		9

can be derived

Moving Average (MA) Models: Compute Errors

An example of an MA(1) Model

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

10

0.5

assumed to be given, e.g., through EM

Time	Forecasted Values (\hat{y}_t)	Error at time t (ε_t)	Actual Values (y_t)
1	10	-1	9
2	$9.5 = 10 + 0.5(-1)$	2	11.5
3	$11 = 10 + 0.5(2)$	-1	10
4	$9.5 = 10 + 0.5(-1)$	2	10.5
5	$11 = 10 + 0.5(2)$	-1	10
6	$9 = 10 + 0.5(-1)$	0	9

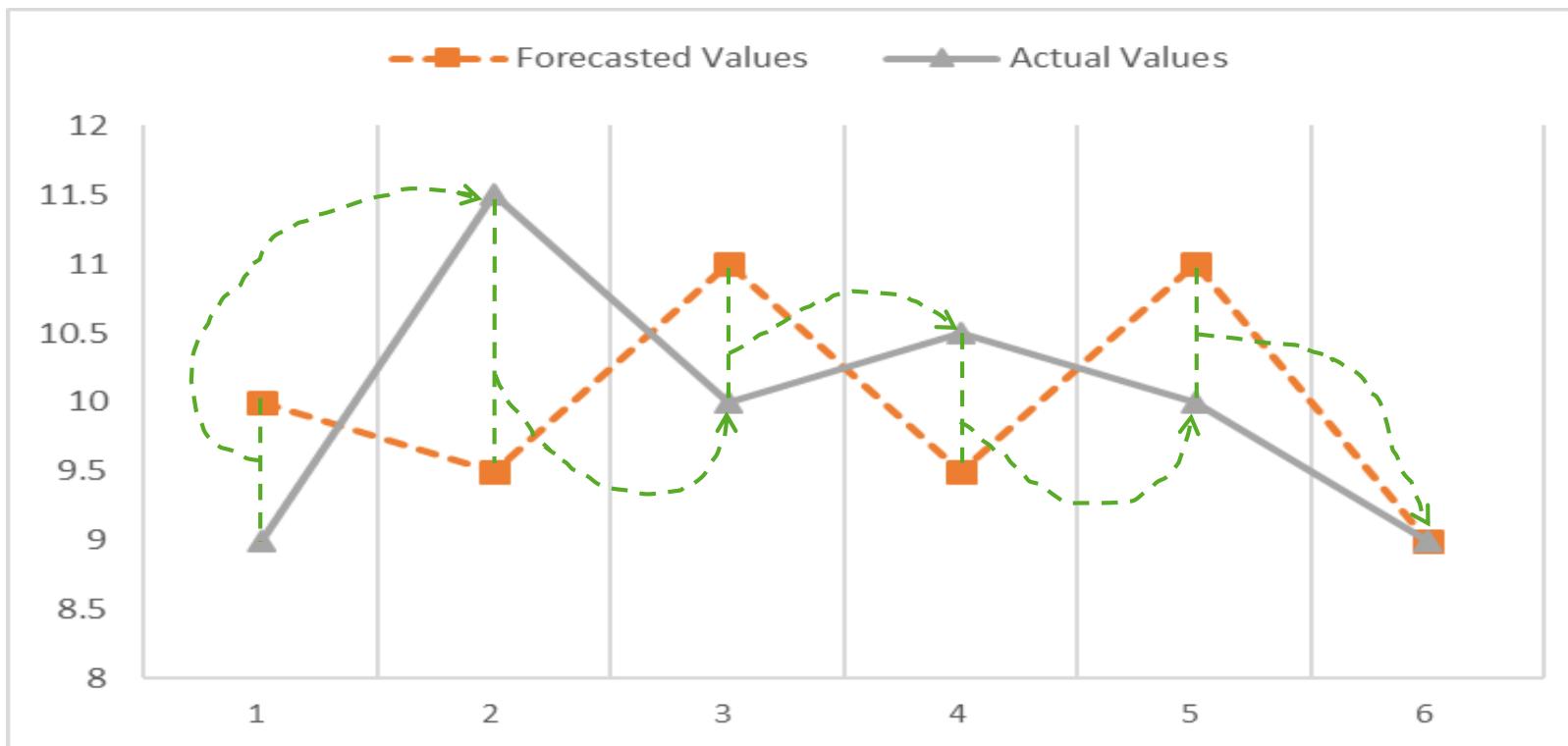
can be derived

Moving Average (MA) Models: Interpret in Reverse

An example of an MA(1) Model

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

10 0.5



ARMA: Combine Autoregressive (AR) and Moving Average (MA) Models

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

Autoregressive (AR) **Moving Average (MA)**

- Note that given $c, \phi_1, \phi_2, \dots, \phi_p, \theta_1, \theta_2, \dots, \theta_q$, it is possible to compute $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_T$.
- Use for example Expectation-Maximization (EM) algorithms

ϕ is “phi”

θ is “theta”

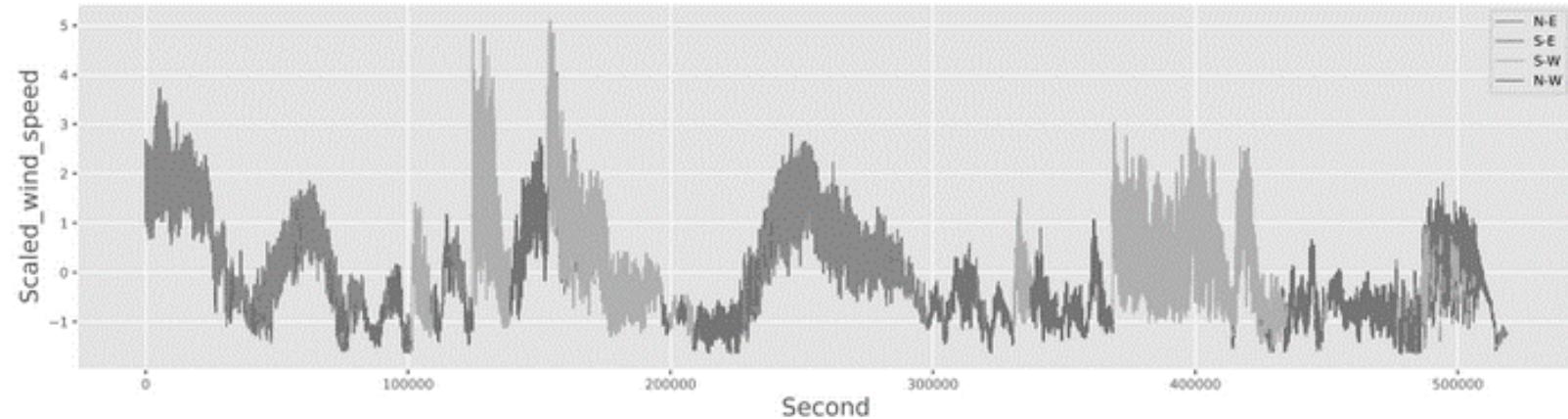
Stationarity & Differencing

The observation y_t of a **stationary time series** does not depend on the time t .

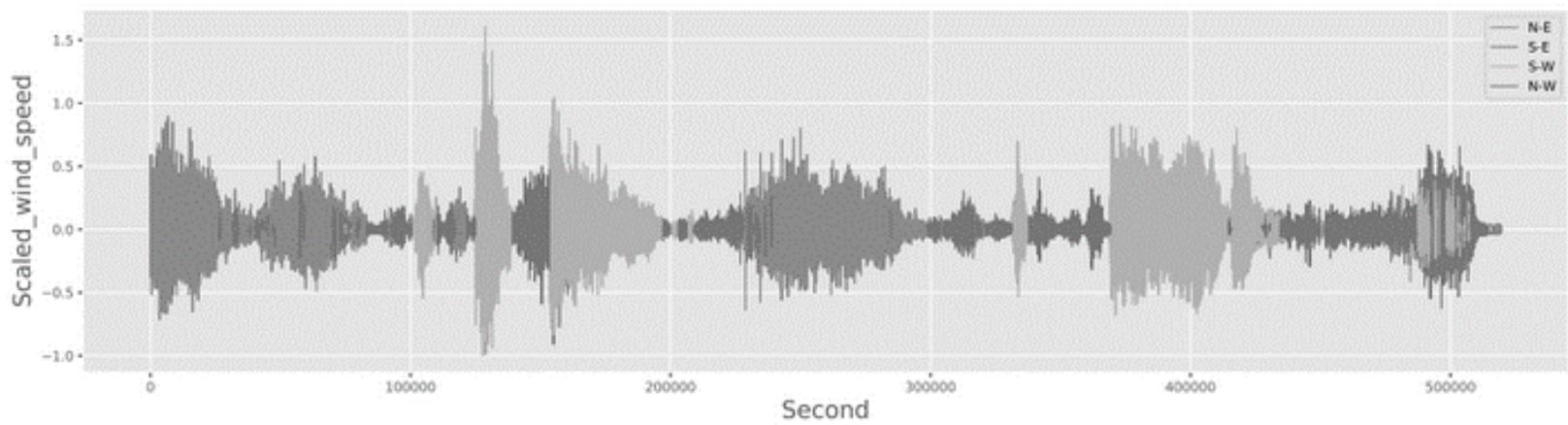
- Obviously, time series with trends or seasonality are not stationary.
- Therefore, one may use **differencing** to focus on differences rather than absolute values (**first-order differencing**)
- The new series represents the change between consecutive observations: $y'_t = y_t - y_{t-1}$. The result is a new time series and then it is “business as usual”
- Sometimes, it may be necessary to difference the series a second time (**second-order differencing**) to make it stationary $y''_t = y'_t - y'_{t-1} = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) = y_t - 2y_{t-1} + y_{t-2}$.

Stationarity & Differencing

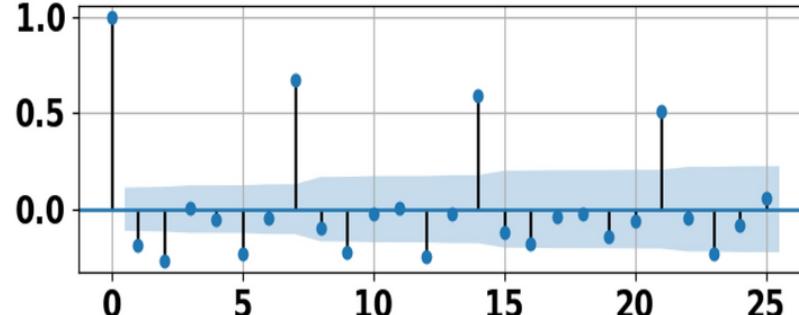
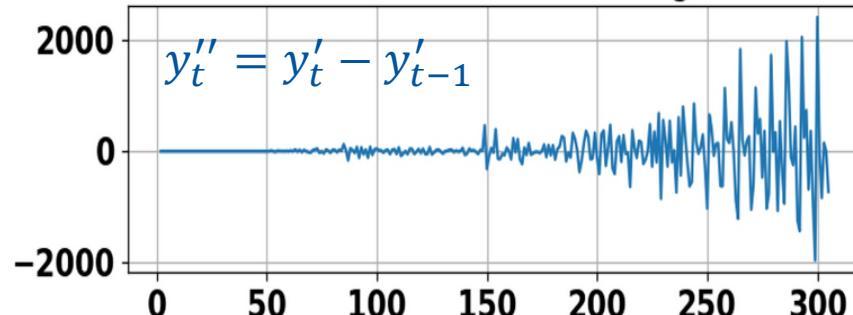
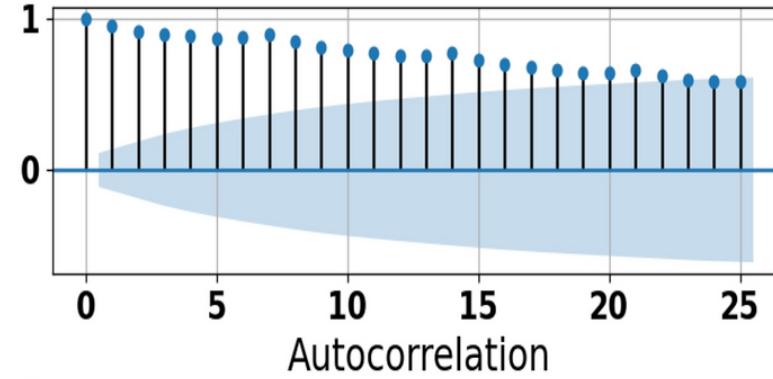
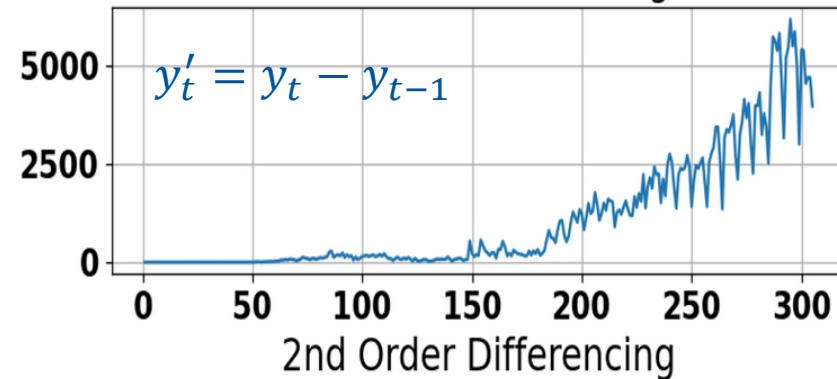
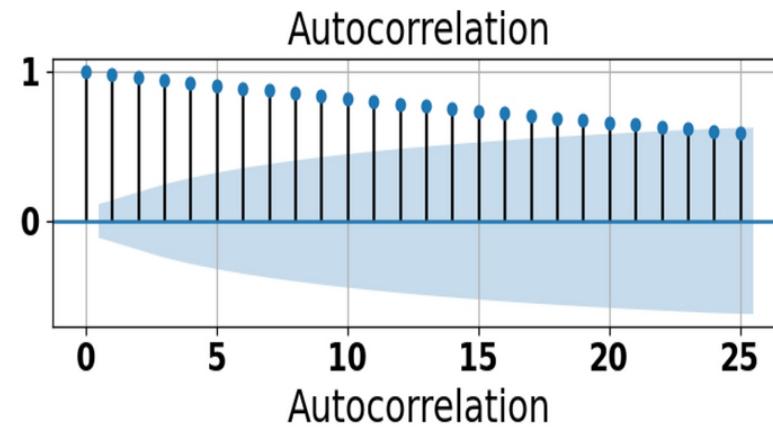
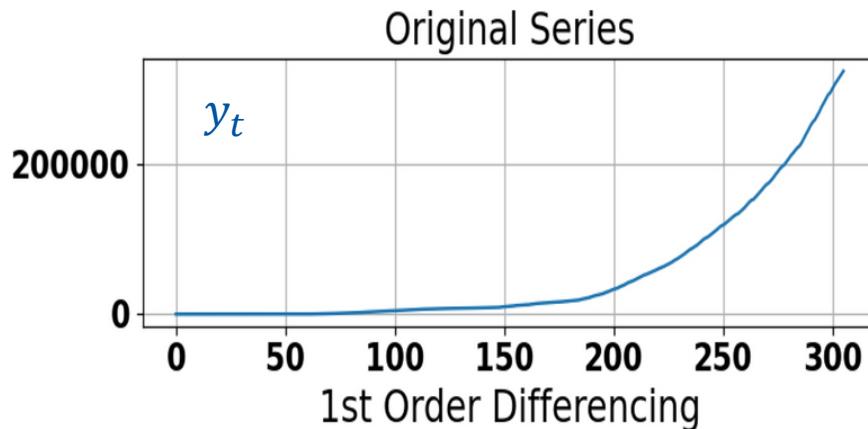
y_t



$y'_t = y_t - y_{t-1}$



Stationarity & Differencing

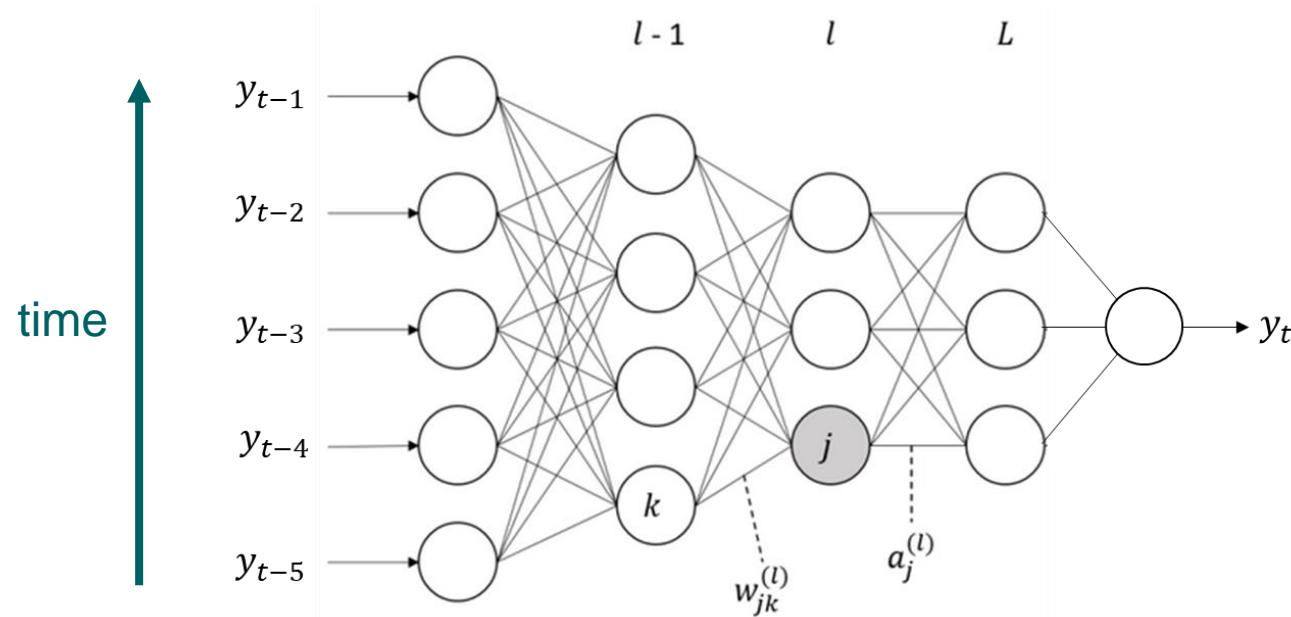


ARIMA Models

- Combination of the elements present before
 - **AR**: Autoregressive (lagged values)
 - **I**: Integrated (differencing to make series stationary)
 - **MA**: Moving Average (lagged errors)
- Hence **ARIMA** is **ARMA** with differencing
- Parameters: $ARIMA(p, d, q)$
 - p = order of the autoregressive part
 - d = number of times for differencing
 - q = order of the moving average part

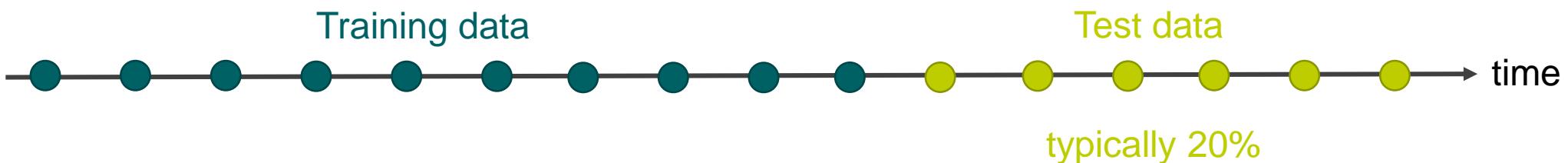
Machine Learning Models

- Feed-forward Neural Networks
- Recurrent Neural Networks, e.g, long short-term memory
- ...



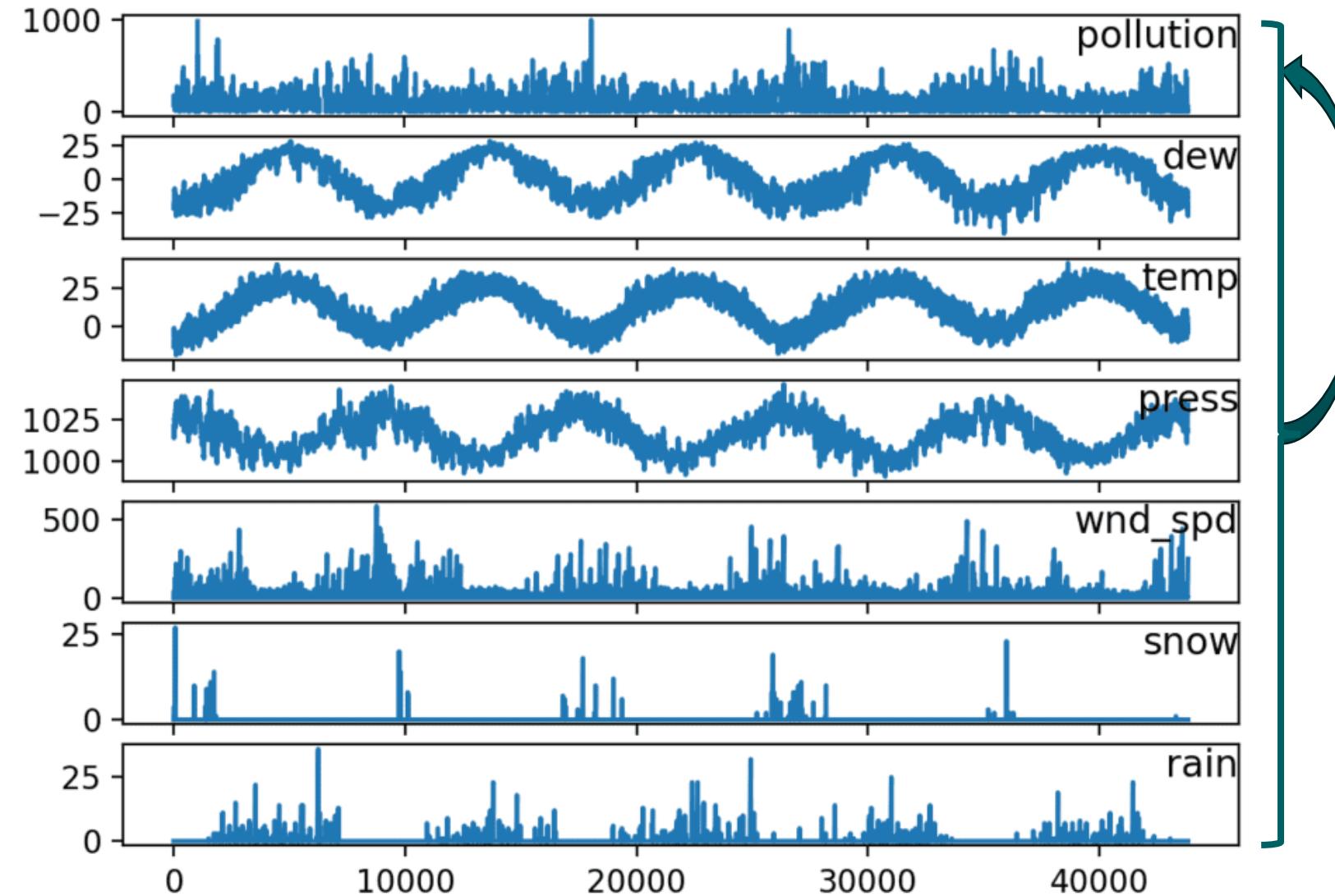
Evaluation

- A model that fits the training data well will not necessarily forecast well.
- We should split the data into training and test set.



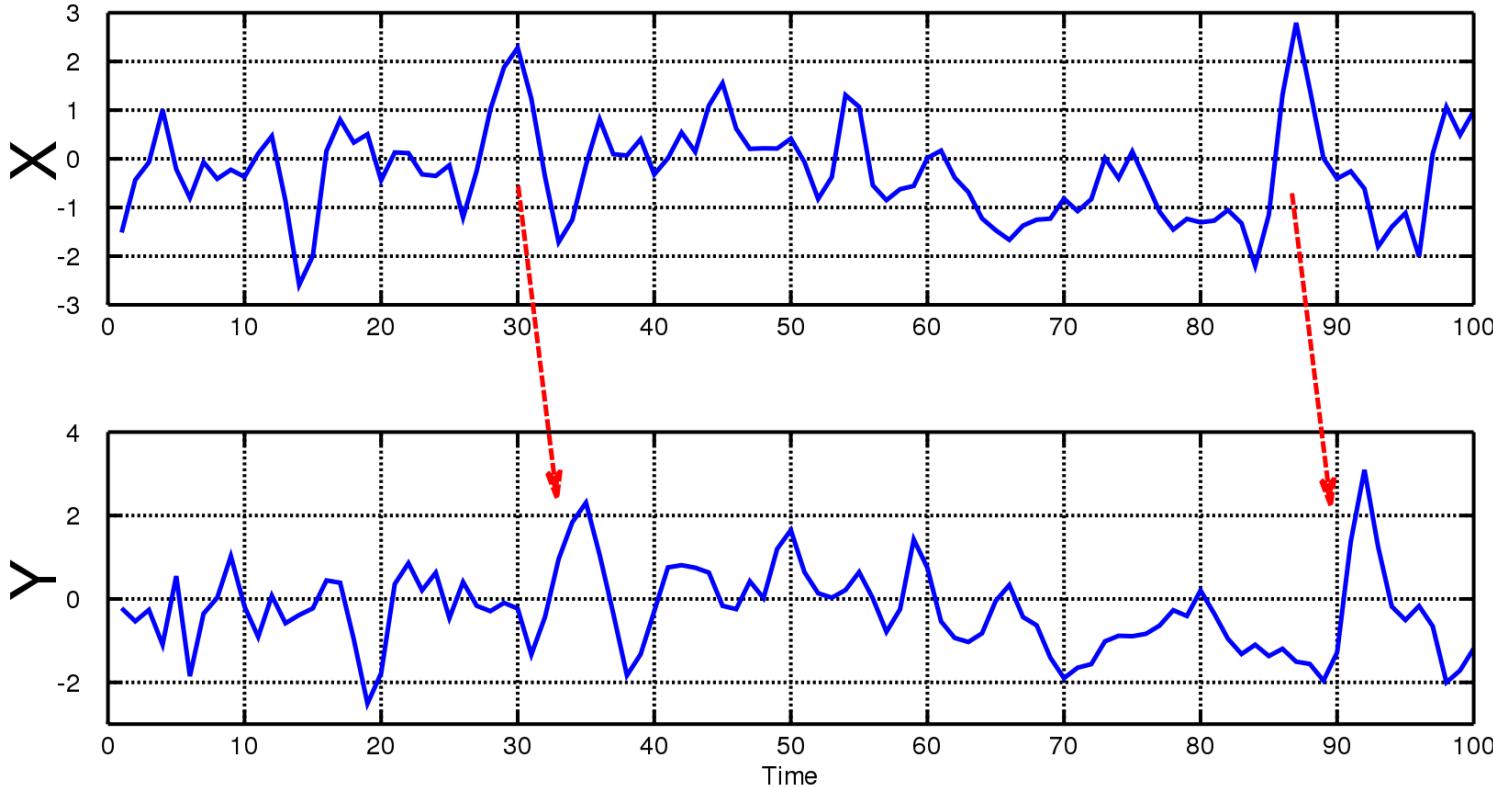
- Forecast errors: the difference between an observed value and its forecast
- Examples:
 - Mean Absolute Error (MAE): $\frac{\sum_{i=1}^n |\epsilon_i|}{n}$
 - Mean Squared Error (MSE): $\frac{\sum_{i=1}^n \epsilon_i^2}{n}$

From Univariate To Multivariate Time Series



- Next to time as a feature, multiple numerical features.
- One feature may depend on itself and other features.
- Assumption: future values do not cause current values.

Granger Causality



- The Granger causality test is a statistical hypothesis test for determining whether one time series is useful in forecasting another.
- Are predictions of the value of Y based on its own past values and on the past values of X better than predictions of Y based only on Y's own past values?
- Not really causality ...

https://en.wikipedia.org/wiki/Granger_causality

Let's Take A Step Back: How to Get the Data?



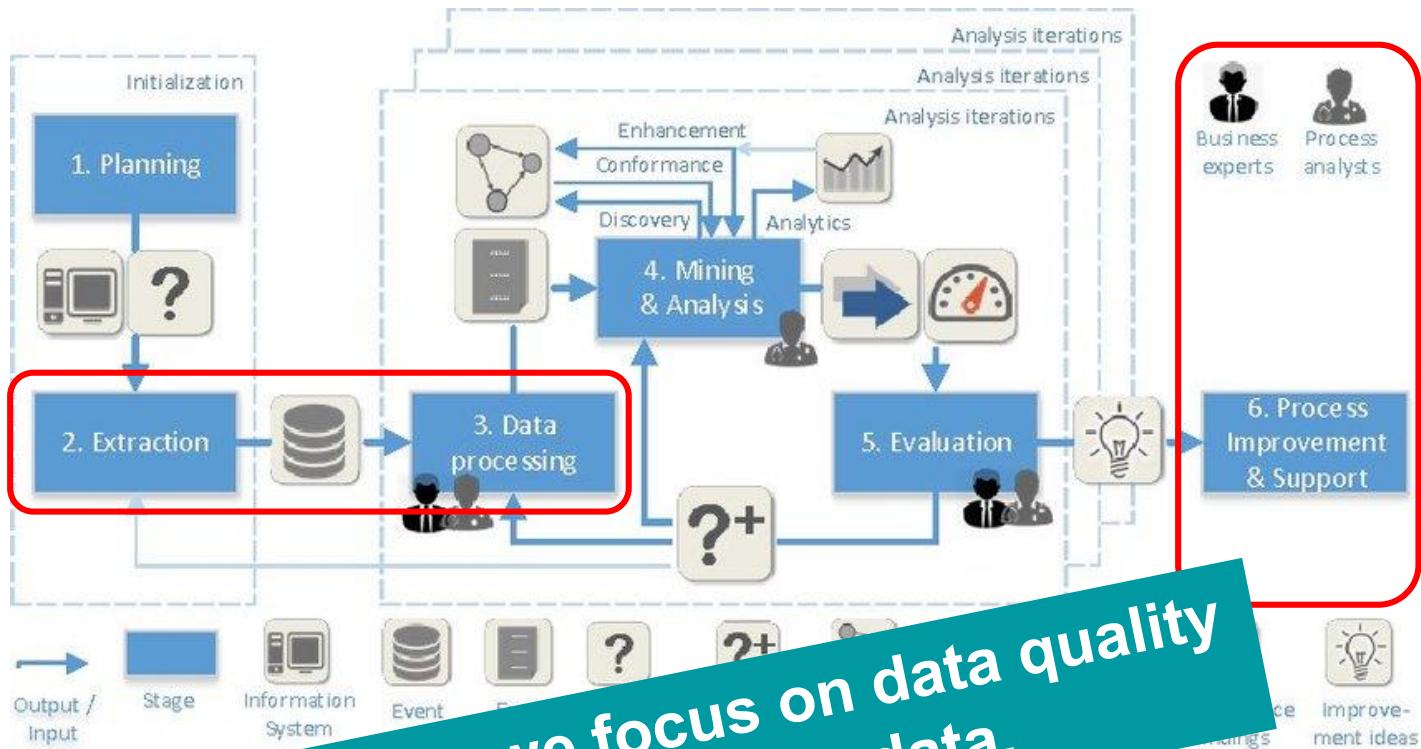
80/20

It is not uncommon that 80% of the effort/time in a data science project is devoted to finding, extracting, cleaning, and transforming the data. Only 20% is concerned with analysis.

The Two Biggest Hurdles in Practice: Getting the Data and Implementing Changes



Example: Process Mining



In the remainder, we focus on data quality and preprocessing of tabular data.

EKPO – Purchasing Document Item
#1 MANDT – Client
#2 EBELN – Purchasing Document Number
#3 EBELP – Item Number of Purchasing Document
#4 LOEKZ – Deletion indicator in purchasing document
#5 STATU – RFQ status
...
#299 POL_ID – Order List Item Number
#300 CONS_ORDER – Purchase Order for Consignment

CIO of a US bank: “We reduced the number of applications from 12.000 to 8.000” : -)

An SAP installation has hundreds of thousands of tables.

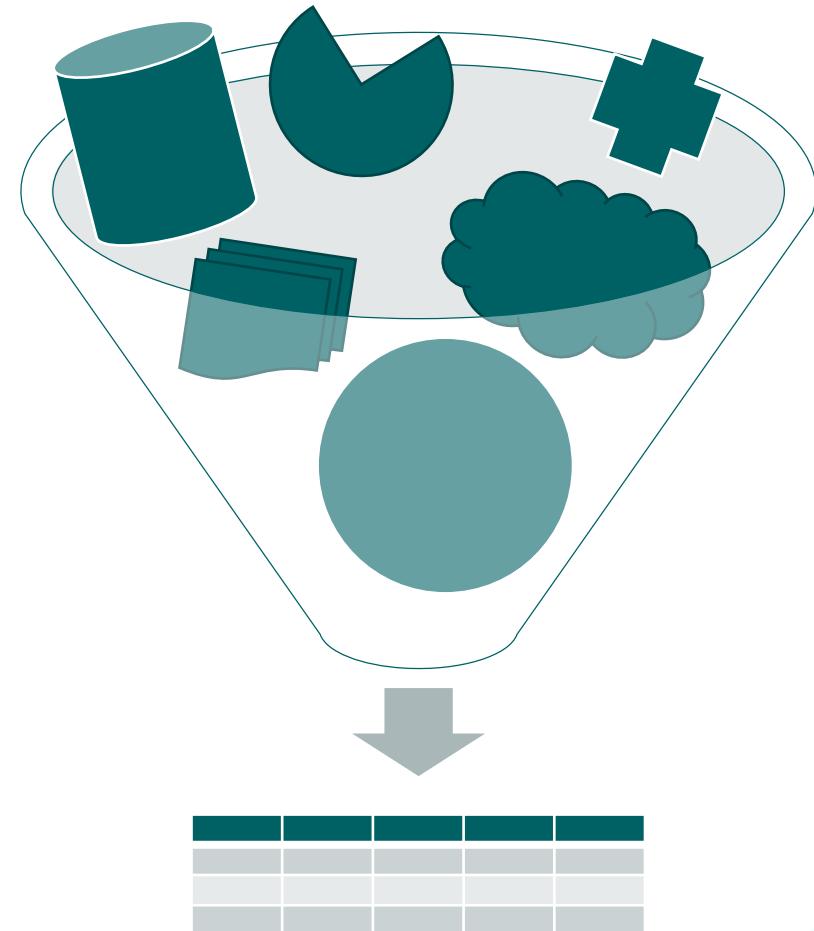
Tables may have hundreds of columns (e.g. EKPO has > 300 fields).

Organizations such as Siemens have 70 SAP installations.



Data Quality & Preprocessing

1. Introduction
2. Missing Values
3. Outliers
4. Semantic Problems
5. Transformation & Normalization
6. Data Reduction
7. Conclusion



Data Science Pipeline

- Garbage in, garbage out
- Possible **problems** (big data, security), **errors** (data quality), **biases** (e.g., survivorship bias) everywhere
- Problems, errors and biases **propagate**

Goal: increase data quality and modify the data to suit the analysis question and applied techniques



Data Quality Aspects

- Accuracy
- Completeness
- Consistency
- Timeliness
- Believability
- Interpretability

- Consider when setting up databases etc.
- Use to gain overview of quality of provided data

Name	Age	Siblings	Date of Admission
Sara Johnson	55	0	30.09.2022
NAME	17		23-11-22
Smith	28	2	8/24/22
Emma Miller	2	56	May 10 th , 22
Jones	87	3	220701
...	

Example Data

Data Quality Aspects

- Accuracy
- Completeness
- Consistency
- Timeliness
- Believability
- Interpretability

Are the values correct? Is it possible to identify errors in the data?

Name	Age	Siblings	Date of Admission
Sara Johnson	55	0	30.09.2022
NAME	17		23-11-22
Smith	28	2	8/24/22
Emma Miller	2	56	May 10 th , 22
Jones	87	3	220701
...	

Emma seems to have an improbable number of siblings. Was the value entered incorrectly?

Data Quality Aspects

- Accuracy
- **Completeness**
- Consistency
- Timeliness
- Believability
- Interpretability

Are values missing?
Is there 'disguised' missing data?
(e.g., default or pre-selected values)

Name	Age	Siblings	Date of Admission
Sara Johnson	55	0	30.09.2022
NAME	17		23-11-22
Smith	28	2	8/24/22
Emma Miller	2	56	May 10 th , 22
Jones	87	3	220701
...	

NAME is the default placeholder where the name should have been entered.

Data Quality Aspects

- Accuracy
- Completeness
- **Consistency**
- Timeliness
- Believability
- Interpretability

Does the data adhere to common naming conventions and formats?

Are these conventions and formats used consistently throughout the data?

Name	Age	Siblings	Date of Admission
Sara Johnson	55	0	30.09.2022
NAME	17		23-11-22
Smith	28	2	8/24/22
Emma Miller	2	56	May 10 th , 22
Jones	87	3	220701
...	

The first name is not always included. The admission date format varies.

Data Quality Aspects

- Accuracy
- Completeness
- Consistency
- **Timeliness**
- Believability
- Interpretability

Data may be missing for some time periods
(aging, lost updates, etc.).

Some values may be up-to-date while others are outdated.

Name	Age	Siblings	Date of Admission
Sara Johnson	55	0	30.09.2022
NAME	17		23-11-22
Smith	28	2	8/24/22
Emma Miller	2	56	May 10 th , 22
Jones	87	3	220701
...	

People gain years (i.e., celebrate birthdays) all year, but age is updated only occasionally.

Data Quality Aspects

- Accuracy
- Completeness
- Consistency
- Timeliness
- **Believability**
- Interpretability

Does the user trust the data, i.e., does the user believe the data to be true, real, credible?

Depends on the data source and processing history.

Name	Age	Siblings	Date of Admission
Sara Johnson	55	0	30.09.2022
NAME	17		23-11-22
Smith	28	2	8/24/22
Emma Miller	2	56	May 10 th , 22
Jones	87	3	220701
...	

Previous errors and inconsistencies have decreased the trust in the system.

Data Quality Aspects

- Accuracy
- Completeness
- Consistency
- Timeliness
- Believability
- **Interpretability**

Are the data understandable without much explanation?

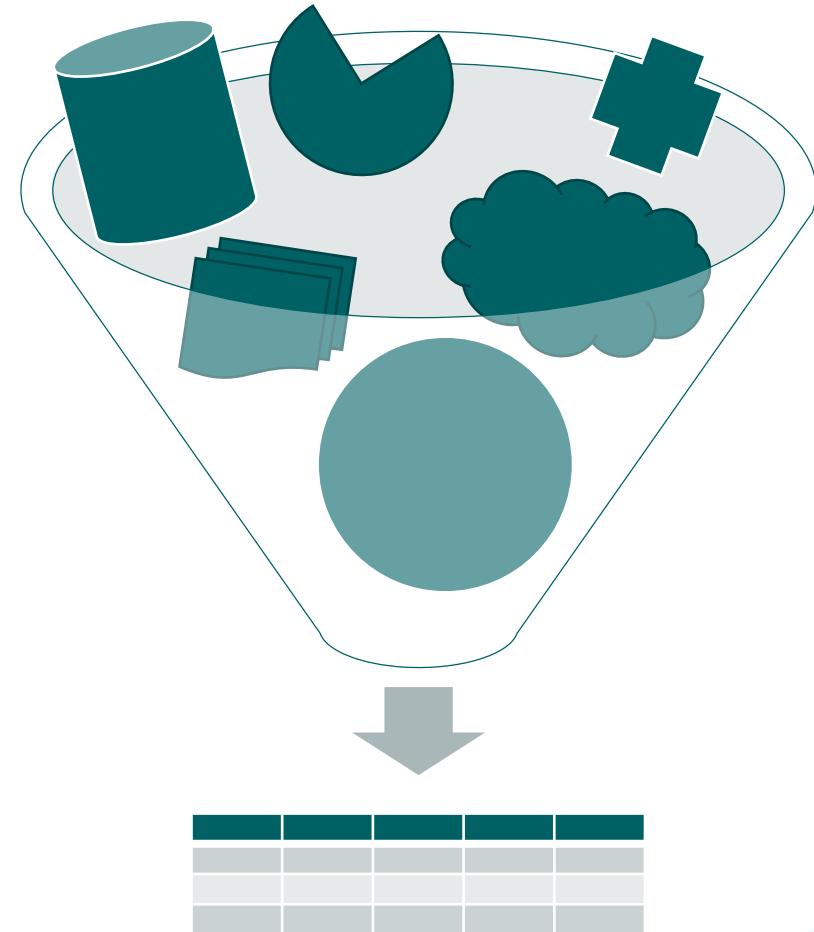
Does it leave room for ambiguity?

Name	Age	Siblings	Date of Admission
Sara Johnson	55	0	30.09.2022
NAME	17		23-11-22
Smith	28	2	8/24/22
Emma Miller	2	56	May 10 th , 22
Jones	87	3	220701
...	

Is the age given in years or in months? Does the number of siblings include half-siblings?

Data Quality & Preprocessing

1. Introduction
2. **Missing Values**
3. Outliers
4. Semantic Problems
5. Transformation & Normalization
6. Data Reduction
7. Conclusion



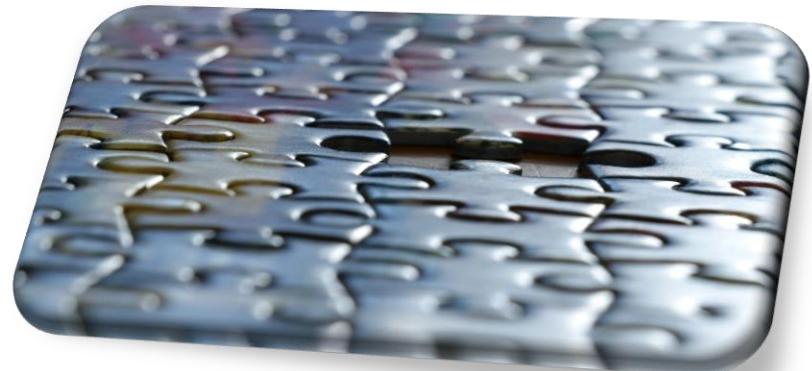
Detecting Missing Values

Missing values may be obvious...

- Empty value
- NaN / NA

... or may be disguised!

- Default value
- Invalid value



Handling Missing Values

- 1) Fill in manually
- 2) Ignore
- 3) Fill in a derived value



1



3



2

Handling Missing Values: Ignore



Discard the feature

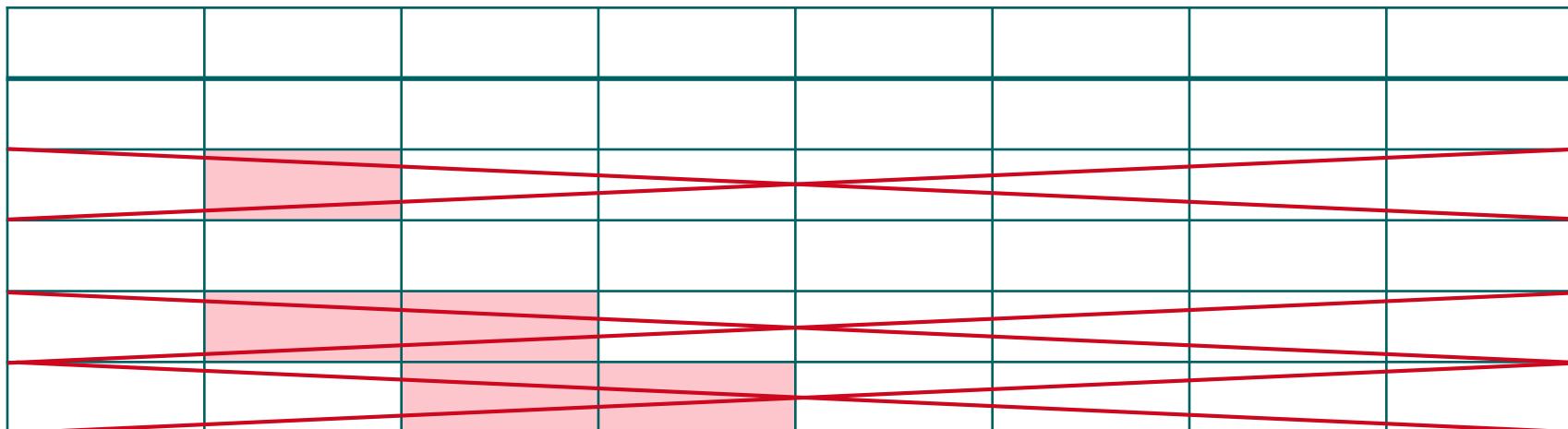
- The whole feature is removed from the data
 - Usually done if the number of missing values is too large to allow meaningful analysis
(as a rule of thumb, if more than 60 % of the feature values are missing)

A grid diagram consisting of 12 horizontal rows and 6 vertical columns. A large red 'X' is drawn across the grid. Four specific cells are shaded light red: the intersection of the second row from the bottom and the fourth column from the left, the intersection of the third row from the bottom and the fourth column from the left, the intersection of the second row from the bottom and the fifth column from the left, and the intersection of the third row from the bottom and the fifth column from the left.

Handling Missing Values : Ignore

Discard the instance

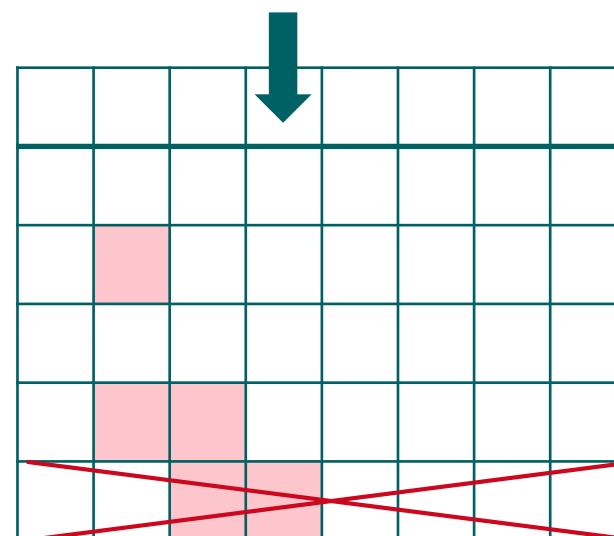
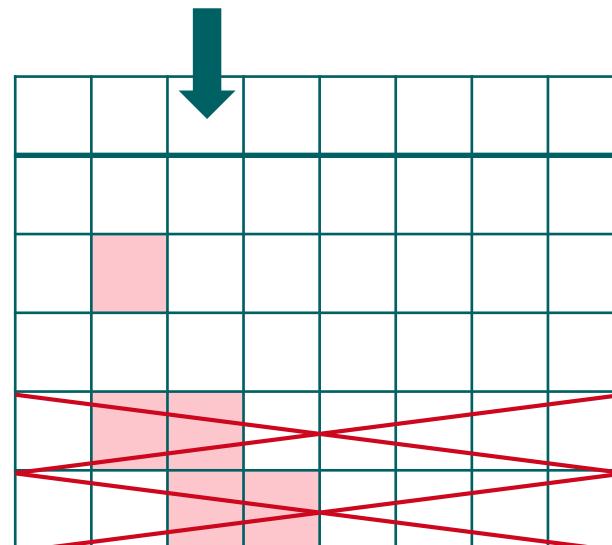
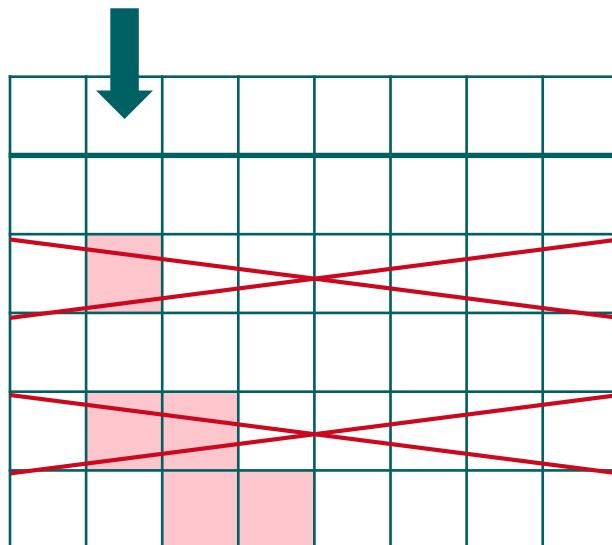
- The entire instance is simply **discarded**
- Usually done when the whole instance becomes unusable (e.g., labeling attribute for classification is missing)
- If the data set misses a lot of values, this technique may make the whole data set **unusable** or introduce a **bias**



Handling Missing Values: Ignore

Ignore the instance only for features where the value is missing

- The instance is ignored when analyzing features where it misses a value
- Information for other features remains usable



Handling Missing Values: Create



Mean/median/mode of the whole feature

- Compute mean/median/mode and fill the gaps accordingly
- May introduce values far away from the real value
- Example: compute yearly income

Handling Missing Values: Create

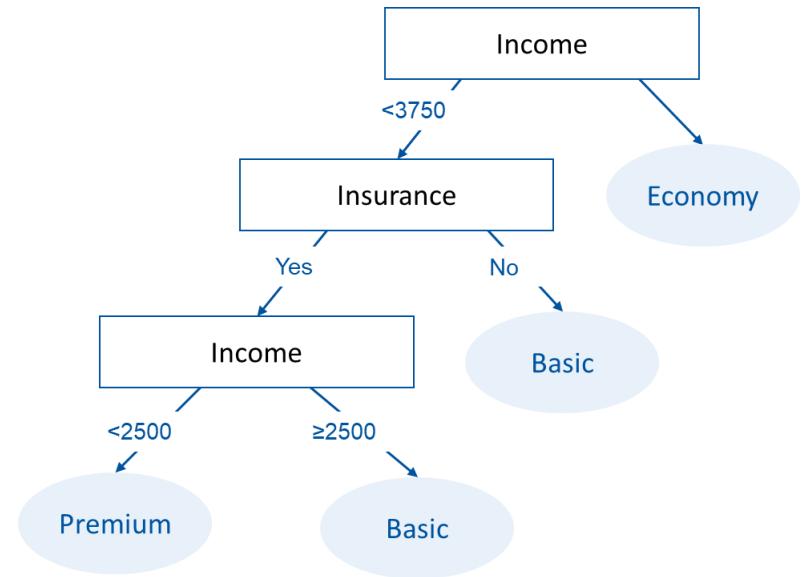
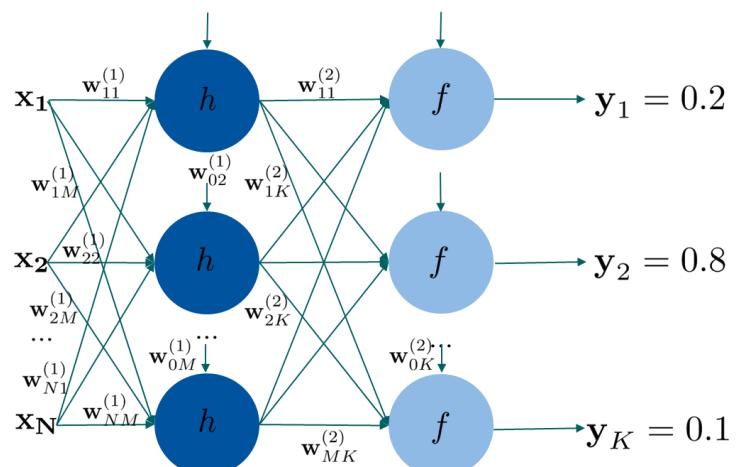
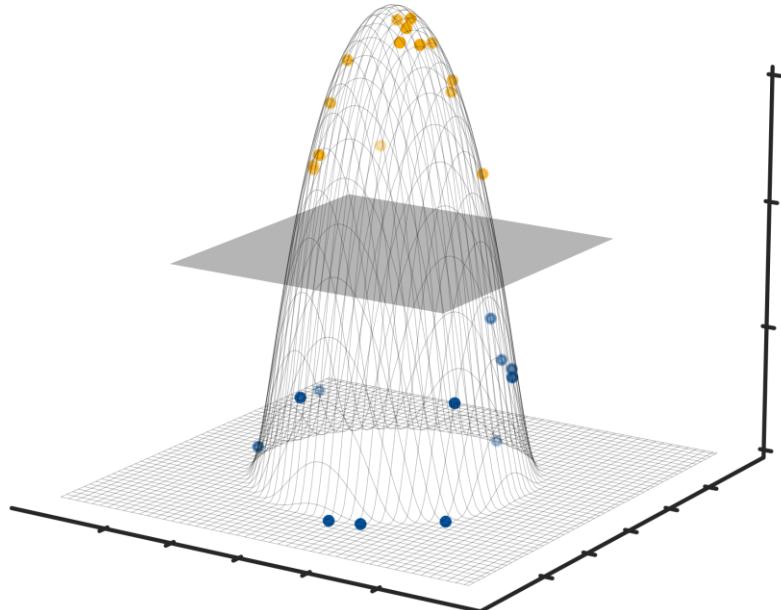
Mean/median/mode of all instance belonging to the same class

- Compute mean/median/mode only based on instances with the same class label
 - Higher chances to be accurate compared to the overall mean/median/mode
 - Only valuable if we have meaningful groups in the data
-
- Example: compute income for a 20-year-old Student living in Aachen, Germany

Handling Missing Values: Create

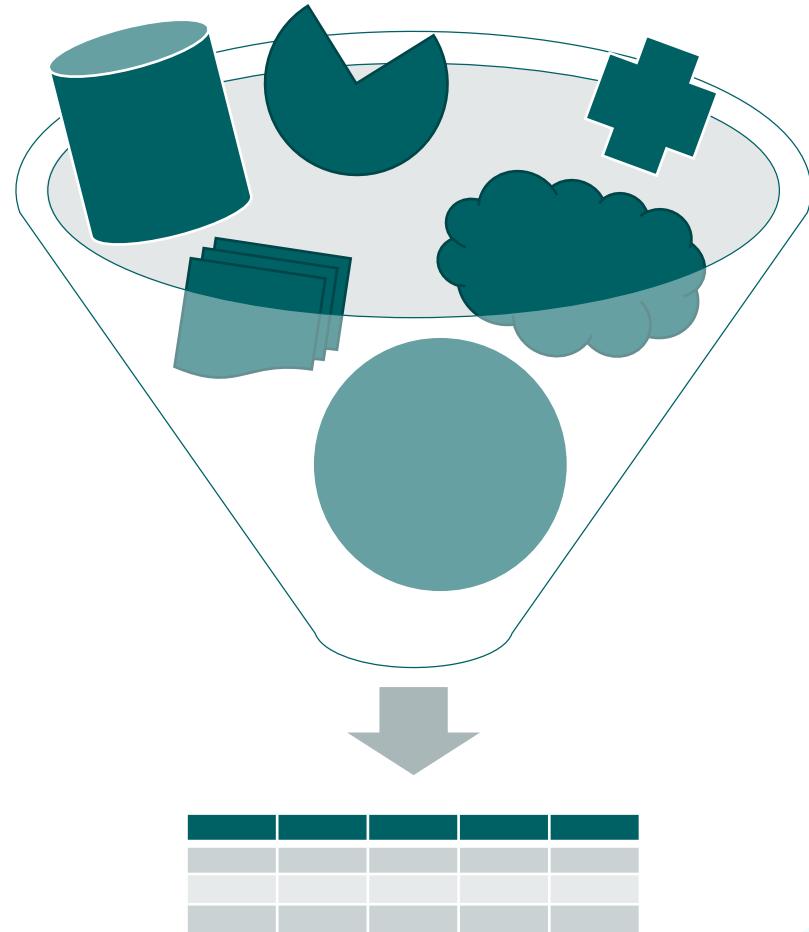
Complex derived value (use a predictor model)

- Fill in the value given by a suitable prediction model
- E.g., decision trees, regression, NNs, SVMs...

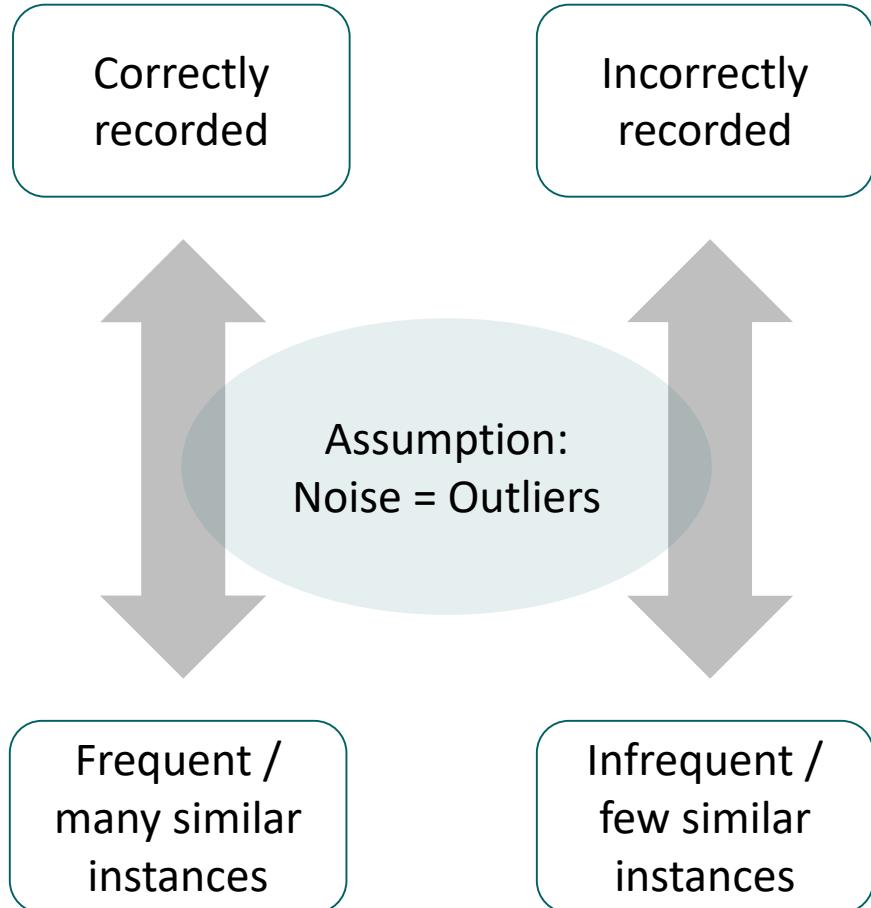


Data Quality & Preprocessing

1. Introduction
2. Missing Values
- 3. Outliers**
4. Semantic Problems
5. Transformation & Normalization
6. Data Reduction
7. Conclusion



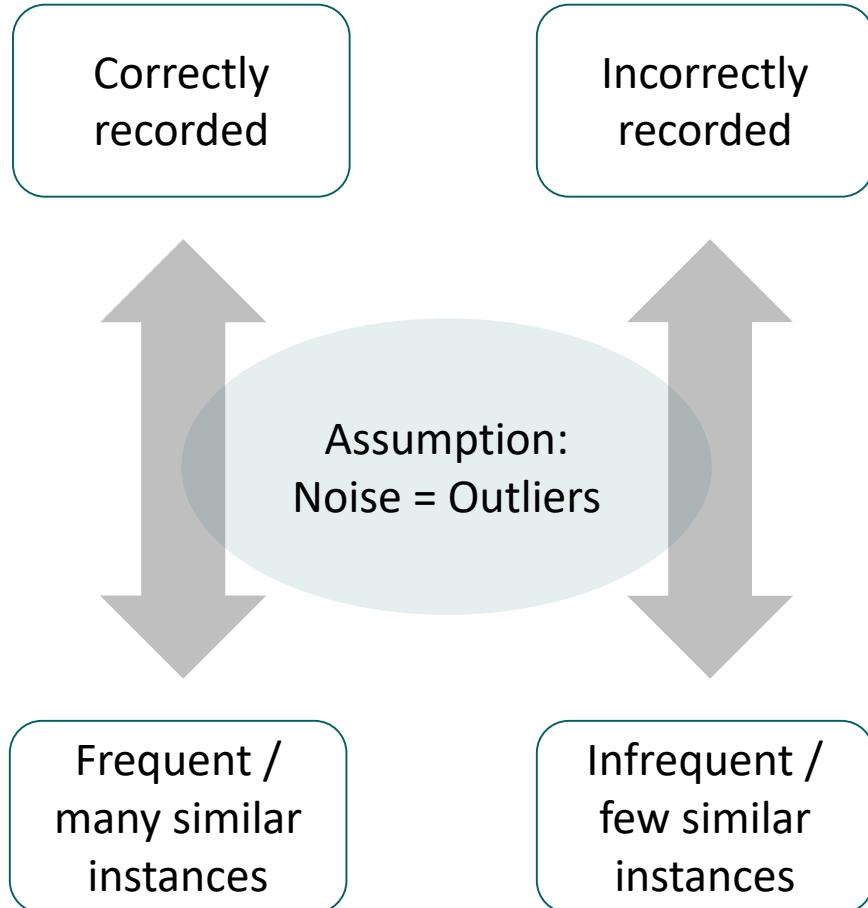
Introduction



What is noise?

- We assume that noise causes outliers
- Thus, **outliers** indicate noise

Outlier Detection

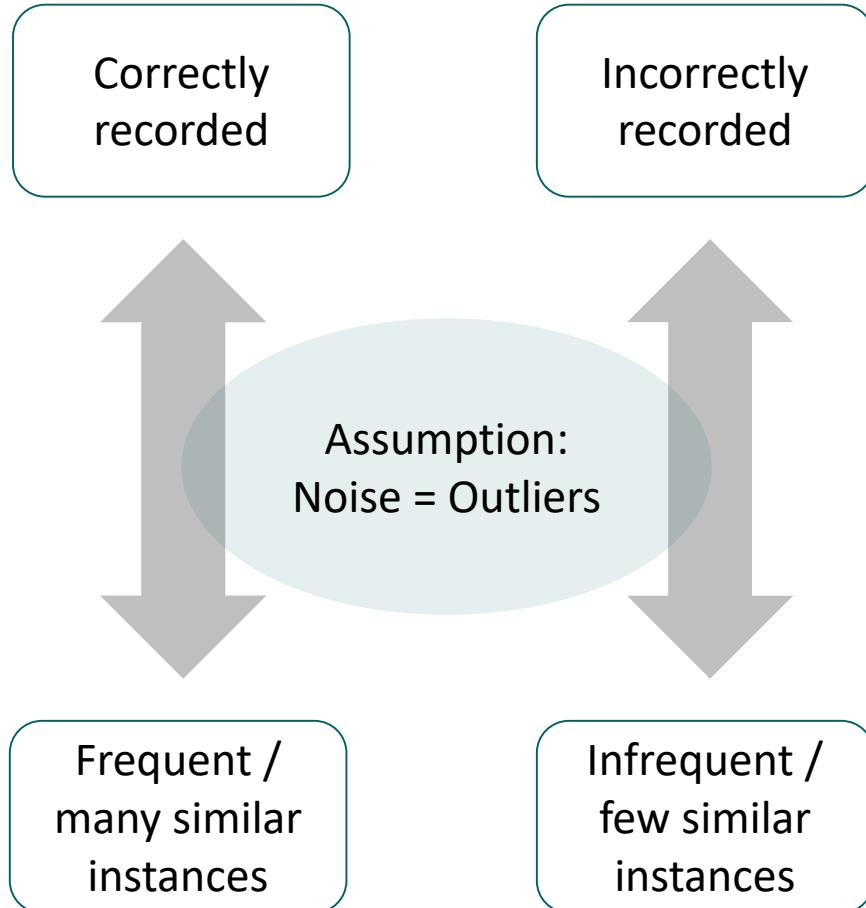


How to detect outliers?

- Boxplots
- Decision trees
- Regression
- SVMs
- Clustering
- ...

→ Predictor models can be used to **define** outliers

Outlier Handling



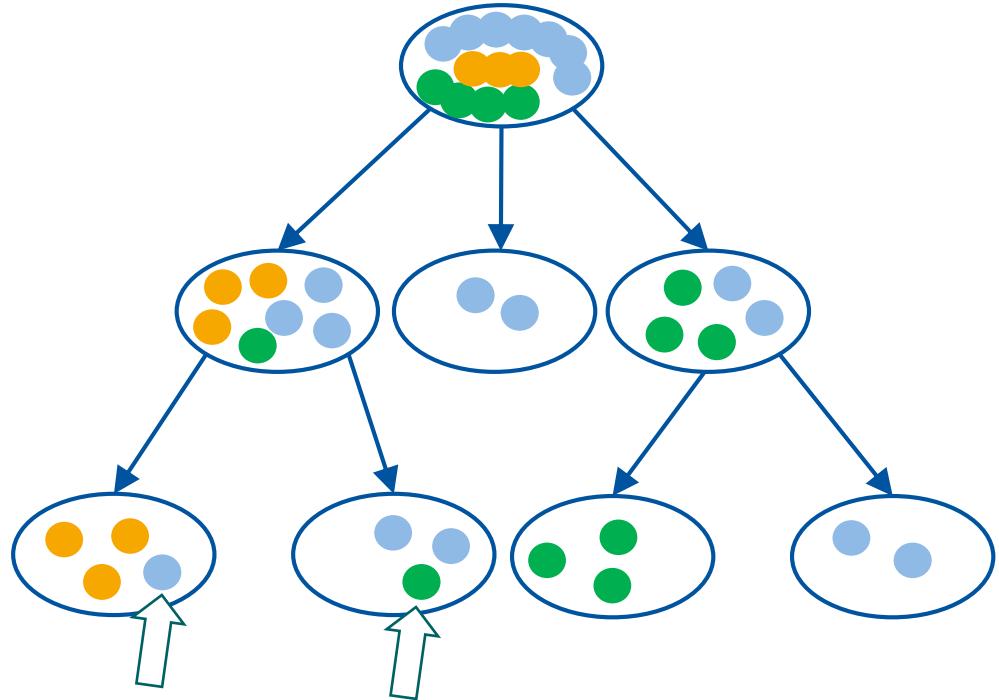
How to handle outliers?

Outliers can be [handled as missing values](#):

- Fill in a correct value manually
- Ignore the feature/instance
- Replace with a derived value

→ Predictor models can be used to [replace outliers](#)

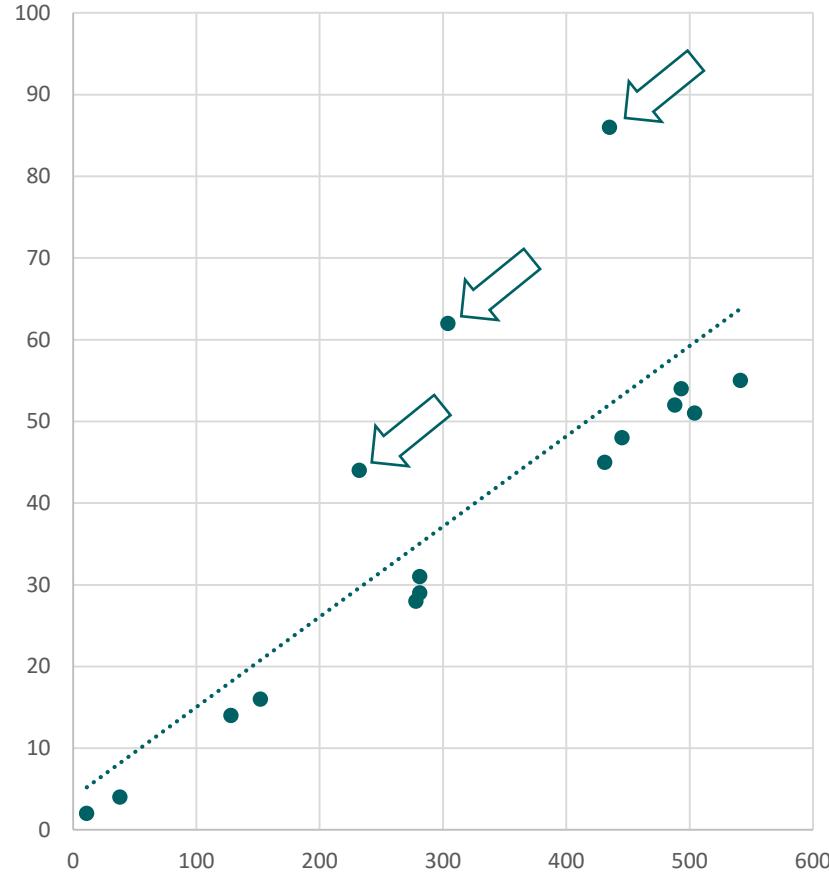
Outlier Detection - Decision Trees



How to detect outliers?

- Every leaf node is assigned a class label
- Instances in that leaf node with a **non-matching class label** can be considered outliers

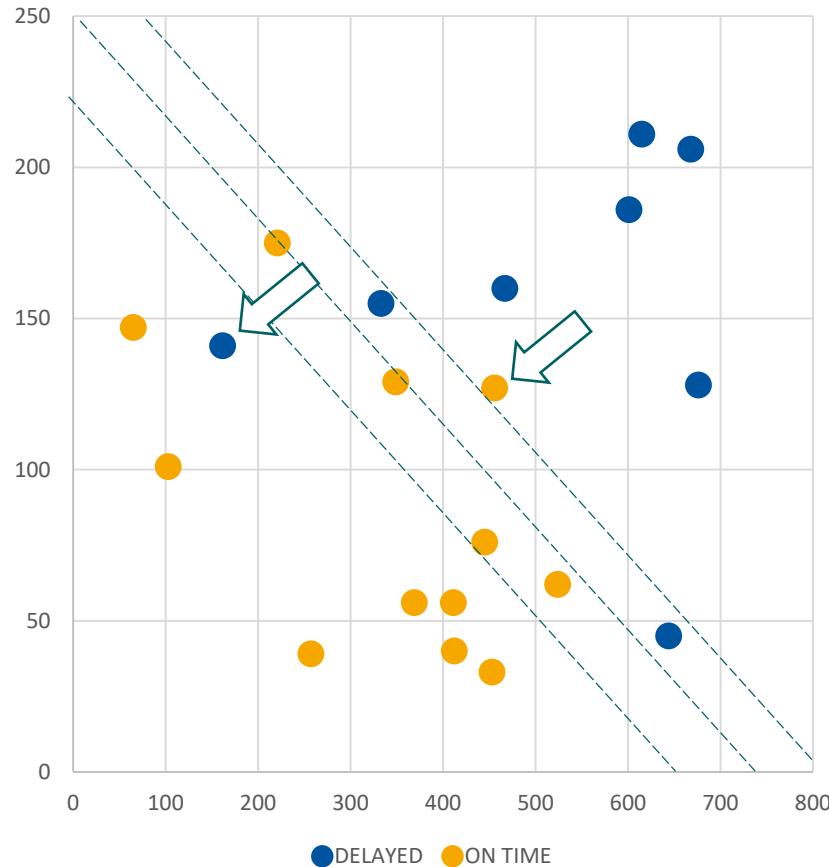
Outlier Detection - Regression



How to detect outliers?

- Instances which are **far away** from the predicted value are considered outliers
- The definition of 'far away' depends on an **error function** and **threshold**

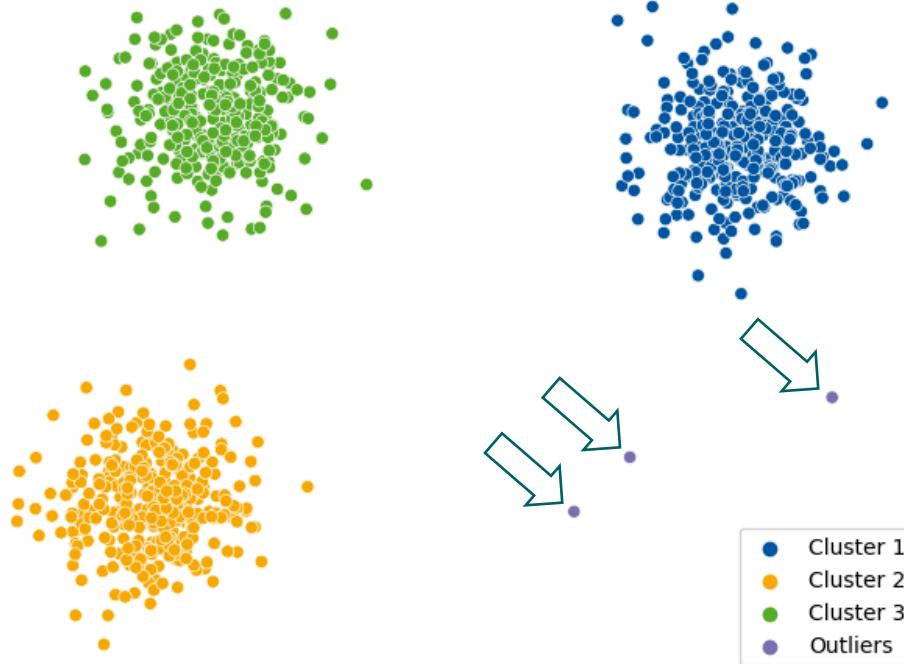
Outlier Detection – SVM



How to detect outliers?

- Instances which are (too far) on the [wrong side](#) of the hyperplane are considered outliers
- Soft margin may be used to define how far

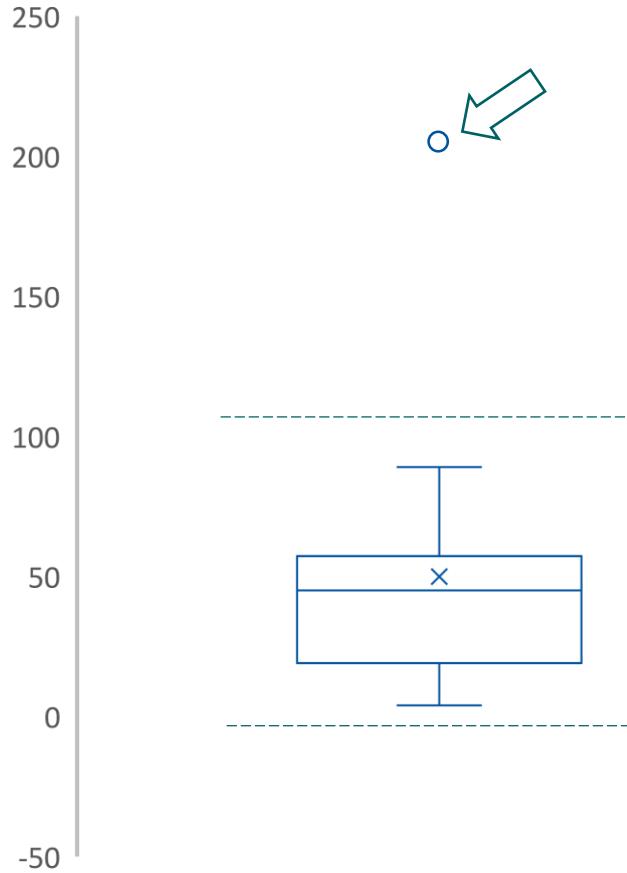
Outlier Detection - Clustering



How to detect outliers?

- Instances **outside of any cluster** can be considered outliers

Outlier Detection - Boxplots



How to detect outliers?

- Instances **above** the upper fence
- Instances **below** the lower fence

→ Outlier handling option:
Clamp values to the nearest fence

Outlier Handling



[1]

How to handle outliers?

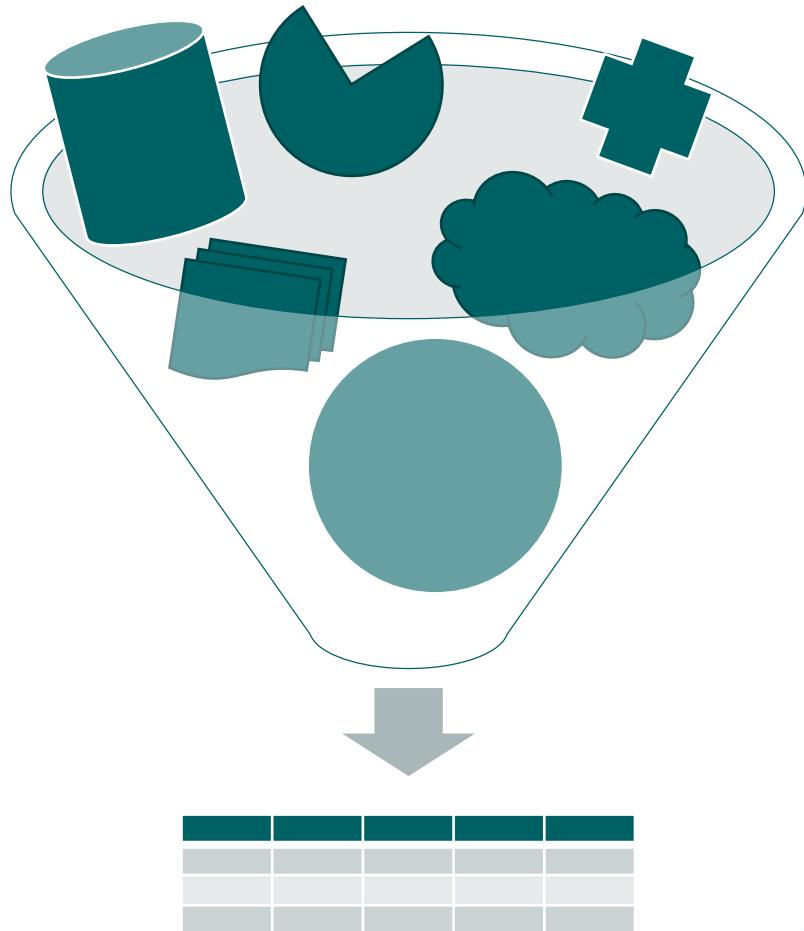
Outliers can be [handled as missing values](#):

- 1) Fill in a correct value manually
- 2) Ignore the feature/instance
- 3) Replace with a derived value

Again: the appropriate method depends on the data and purpose

Data Quality & Preprocessing

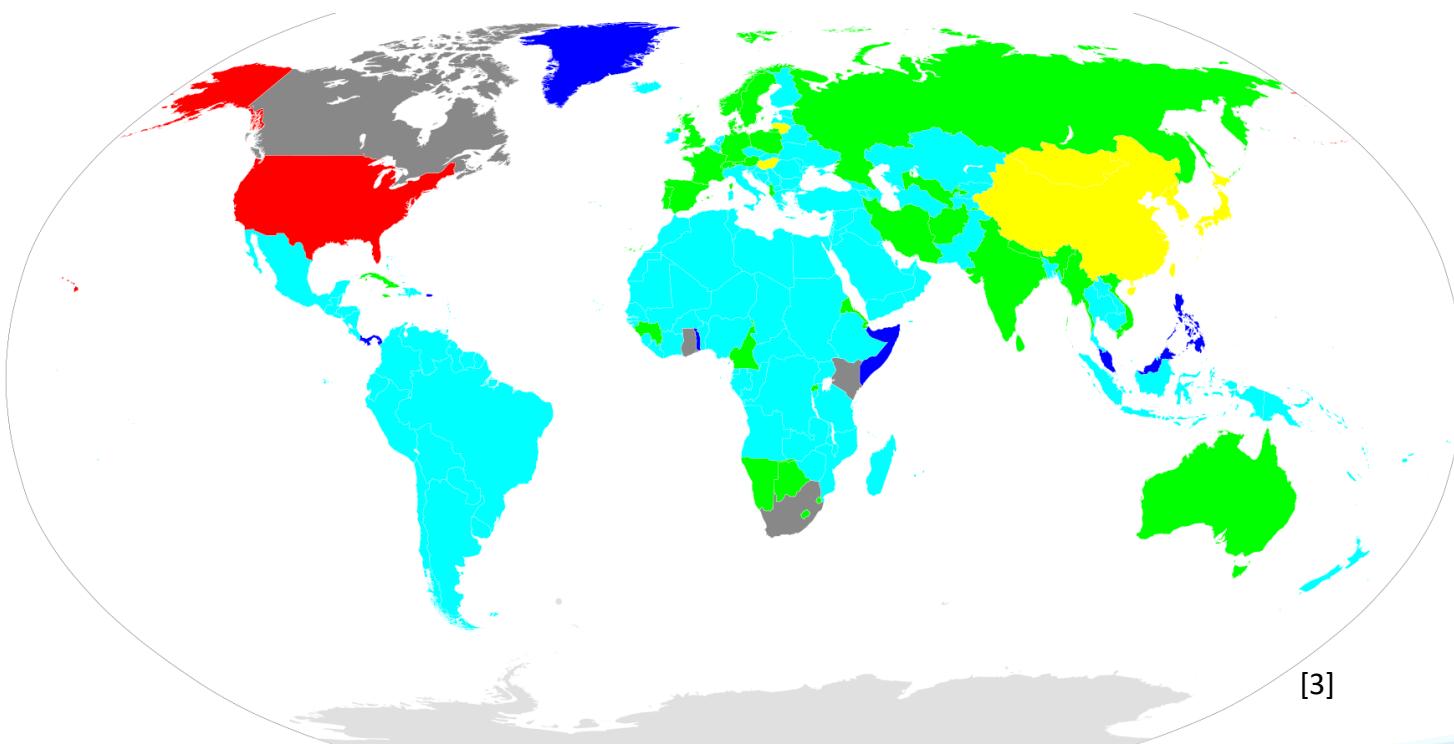
1. Introduction
2. Missing Values
3. Outliers
- 4. Semantic Problems**
5. Transformation & Normalization
6. Data Reduction
7. Conclusion



Semantic Problems - Examples

Data integration

- Merging systems
- Merging data sources



Region Common Format

blue	dd-mm-yyyy mm-dd-yyyy
green	dd-mm-yyyy yyyy-mm-dd
grey	mm-dd-yyyy yyyy-mm-dd dd-mm-yyyy
yellow	yyyy-mm-dd
red	mm-dd-yyyy yyyy-mm-dd
cyan	dd-mm-yyyy

06/07/2023 vs 07/06/2023

Data Integration

Data is collected by different systems and stored separately → merging can introduce problems

Redundancies:

- multiple entries referring to the same instance (often caused by maintenance inconsistencies)
- e.g. two entries for the same purchase but with different addresses (duplication instead of update)
- ‘Wil van der Aalst’, ‘van der Aalst, Wil’, ‘W. Aalst’, ‘Willibrordus Van’, ‘W.M.P.’…

Name	Age	Date of Admission
Sara Johnson	55	30.09.2022
Sara Johnson	56	30.09.2022
Bob Smith	28	8/24/22
...		...

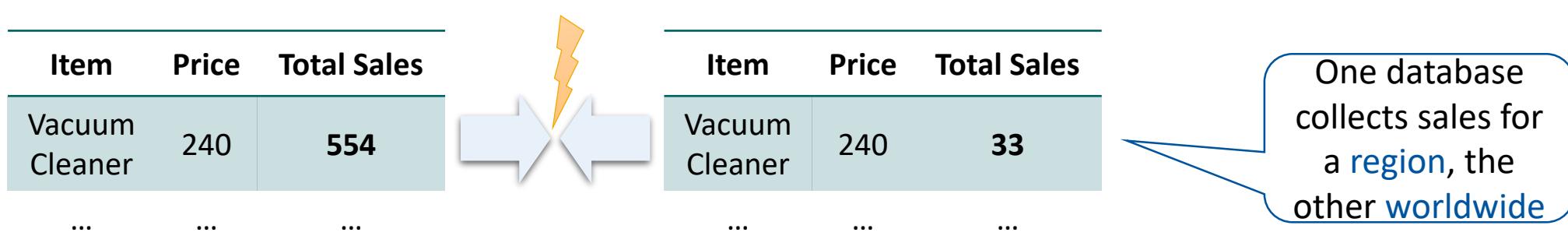
Different instances
or duplication
instead of update?

Data Integration

Data is collected by different systems and stored separately → merging can introduce problems

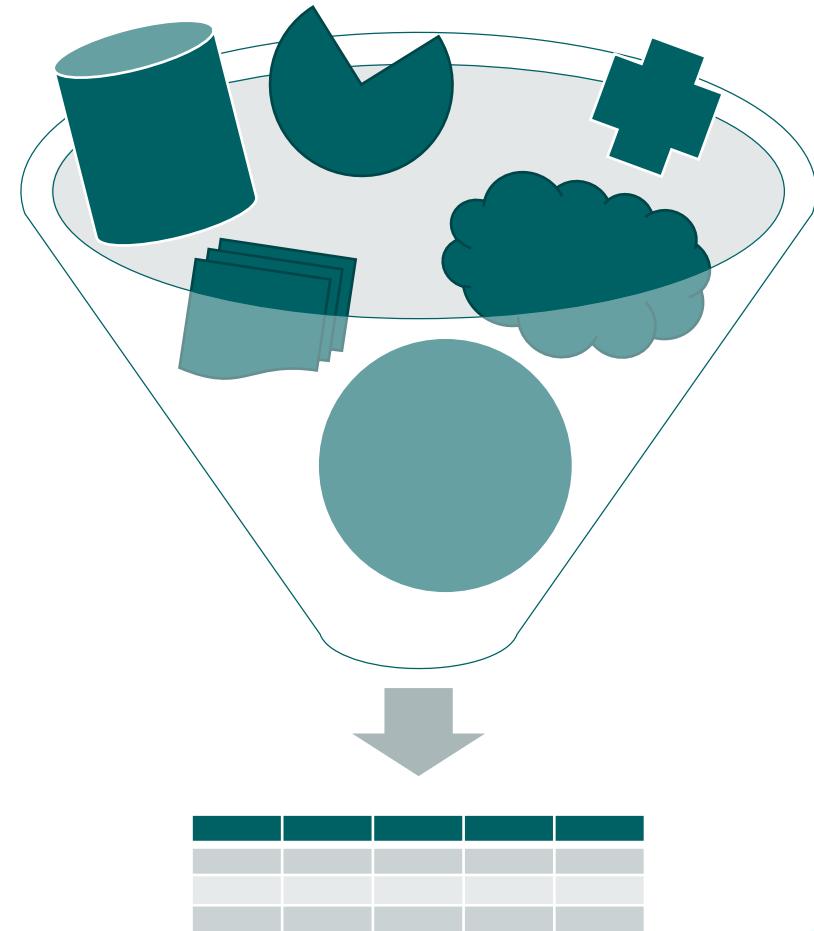
Inconsistencies & Semantic problems:

- data sources using different scales, scopes, encoding, representation, abstraction levels ...
- e.g., prices use different currencies, and may be represented with or without VAT
- e.g., 'total sales' might refer to company wide sales, or to one specific region/country



Data Quality & Preprocessing

1. Introduction
2. Missing Values
3. Outliers
4. Semantic Problems
- 5. Transformation & Normalization**
6. Data Reduction
7. Conclusion



Preprocessing – Preparing the Data for Analysis

- Transformation: change the data to the right data type
- Normalization: adjust the influence of features
- Reduction: make the data smaller for analysis



[1]

Preprocessing – Transformation

- One-hot encoding: categorical to numerical
- Binning: numerical to categorical

The diagram illustrates the transformation of a dataset from raw features to one-hot encoded features. On the left, a table shows raw data with columns f_1 , f_2 , and class. The f_1 column has categories high, high, medium, low, high, and medium. The f_2 column has values 88, 76, 32, 89, 21, and 45. The class column has values A, B, B, C, C, and A. A blue arrow points from this table to a second table on the right, which represents the one-hot encoding of the f_1 column.

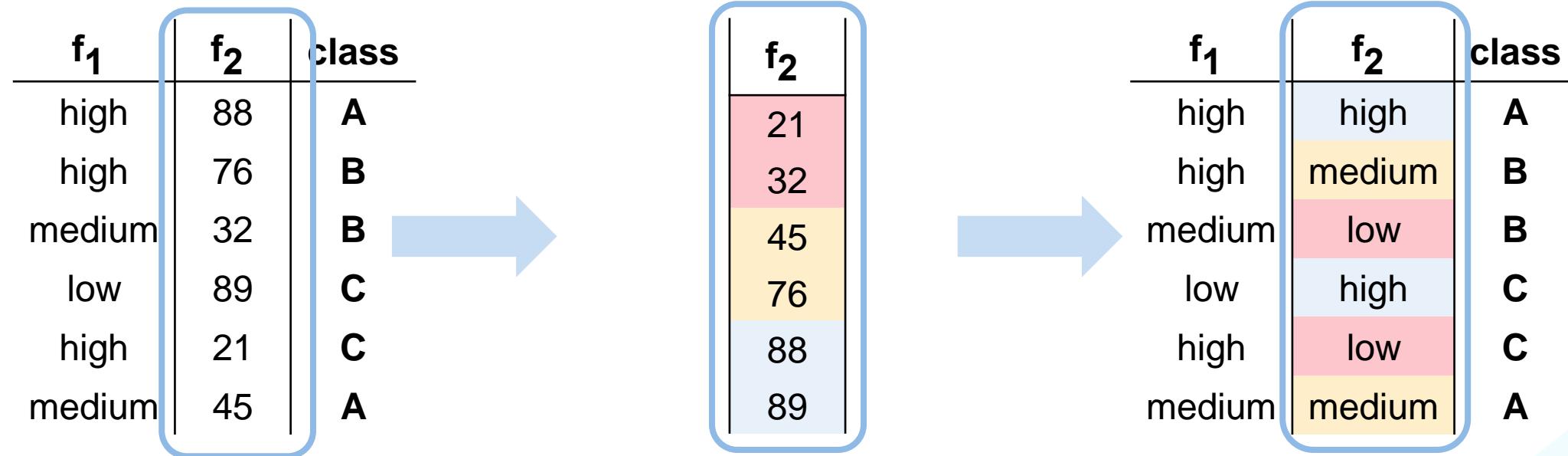
f_1	f_2	class
high	88	A
high	76	B
medium	32	B
low	89	C
high	21	C
medium	45	A

→

f_1 - high	f_1 - medium	f_1 - low	f_2	class
1	0	0	88	A
1	0	0	76	B
0	1	0	32	B
0	0	1	89	C
1	0	0	21	C
0	1	0	45	A

Preprocessing – Transformation

- One-hot encoding: categorical to numerical
- Binning: numerical to categorical

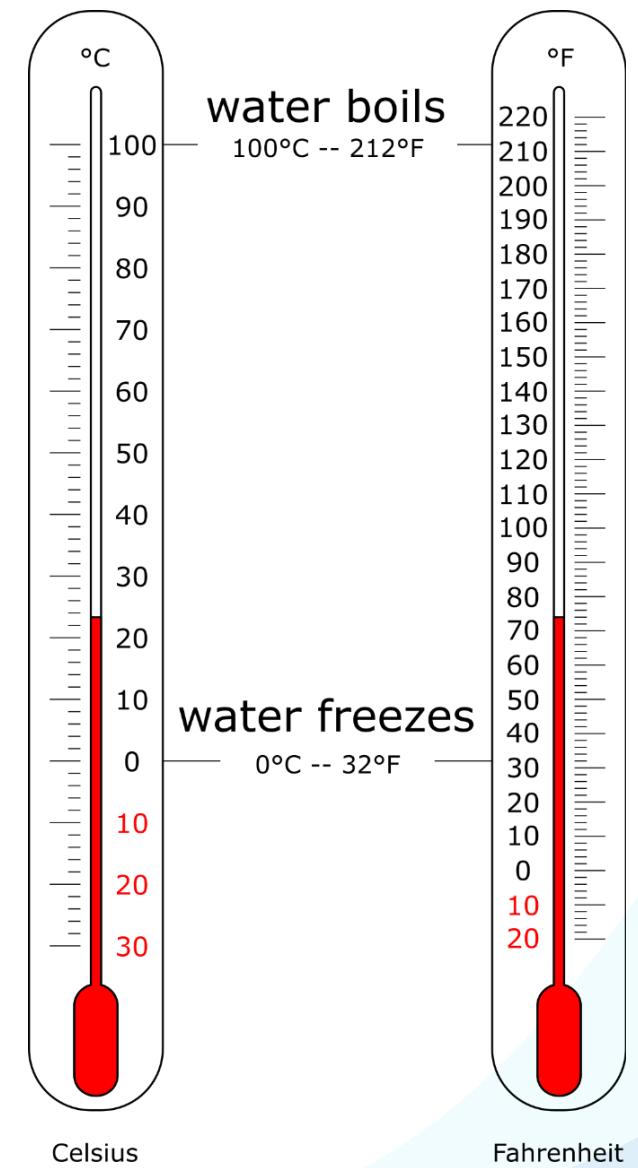


Preprocessing – Normalization

Adjusting the influence of features

- Feature weight and range often depends on the chosen unit (km, mm, miles, ...)
 - Algorithms tend to give more weight to features with a large range
- May introduce an unwanted bias
- May hinder interpretability
- Scales may be non-linear (e.g. logarithmic)

$$\text{Sum of squared errors: } \frac{1}{2} \sum_{i=1}^N (t_i - \mathbb{M}(\mathbf{x}_i))^2$$



Preprocessing – Normalization

Min-max normalization

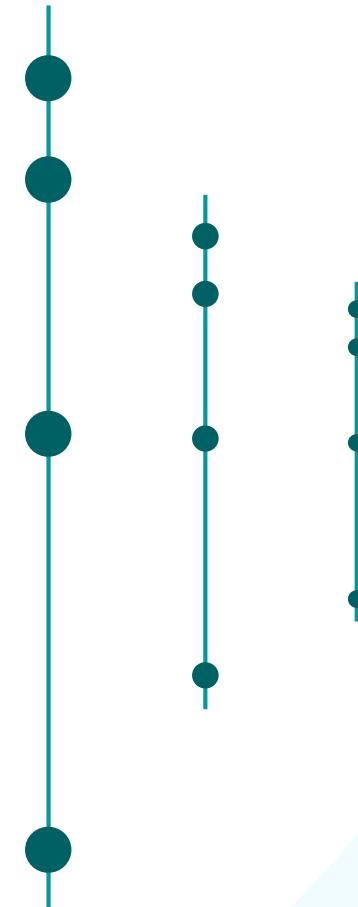
- Maps the values onto a **predefined range** [low, high]
- Preserves **relative differences**, i.e., relations between the data values

We normalize feature d by replacing its value for each instance i as follows:

$$\text{norm}(\mathbf{x}_i[d]) = \frac{\mathbf{x}_i[d] - d_{\min}}{d_{\max} - d_{\min}} \cdot (\text{high} - \text{low}) + \text{low}$$

maximal value
of feature d

minimal value
of feature d



Preprocessing – Normalization

Min-max normalization

d
11
82
33
12
76

$$\begin{aligned}d_{min} &= 11 \\d_{max} &= 82\end{aligned}$$



Consider
high = 100
low = 5

$$\text{norm}(\mathbf{x_i}[d]) = \frac{\mathbf{x_i}[d] - d_{\min}}{d_{\max} - d_{\min}} \cdot (\text{high} - \text{low}) + \text{low}$$

Preprocessing – Normalization

Min-max normalization

d	norm(d)	norm(d)
11	$(11 - 11)/(82 - 11) \cdot (100 - 5) + 5$	5
82	$(82 - 11)/(82 - 11) \cdot (100 - 5) + 5$	100
33	$(33 - 11)/(82 - 11) \cdot (100 - 5) + 5$	34.44
12	$(12 - 11)/(82 - 11) \cdot (100 - 5) + 5$	6.34
76	$(76 - 11)/(82 - 11) \cdot (100 - 5) + 5$	91.97

≈

$$\text{norm}(\mathbf{x_i}[d]) = \frac{\mathbf{x_i}[d] - d_{\min}}{d_{\max} - d_{\min}} \cdot (\text{high} - \text{low}) + \text{low}$$

Preprocessing – Normalization

Standard score (Z-score) normalization

- Uses the standard deviation to quantify the significance of the difference between a value and the overall mean
- Range is $[-\infty, \infty]$, but 0 has a clear meaning
- Useful when actual minimum and maximum of the attribute are unknown
- Useful when outliers may impact min-max normalization

For each i :

$$\text{norm}(\mathbf{x}_i[d]) = \frac{\mathbf{x}_i[d] - \bar{d}}{\text{sd}(d)}$$

\bar{d} is the mean of all values of feature d :

$$\frac{1}{N} \sum_{i=1}^N \mathbf{x}_i[d]$$

$\text{sd}(d)$ is the standard deviation of feature d :

$$\sqrt{\left(\frac{\sum_{i=1}^N (\mathbf{x}_i[d] - \bar{d})^2}{N-1} \right)}$$

Preprocessing – Normalization

Standard score (Z-score) normalization

d
11
82
33
12
76



$$\bar{d} = 42.8$$
$$\text{sd}(d) = 34.259$$

$$\text{norm}(\mathbf{x_i}[d]) = \frac{\mathbf{x_i}[d] - \bar{d}}{\text{sd}(d)}$$

Preprocessing – Normalization

Standard score (Z-score) normalization

d		norm(d)	norm(d)
11		(11 - 42.8)/34.259	-0.93
82	→	(82 - 42.8)/34.259	1.14
33		(33 - 42.8)/34.259	-0.29
12		(12 - 42.8)/34.259	-0.90
76		(76 - 42.8)/34.259	0.97

\approx

$$\text{norm}(\mathbf{x_i}[d]) = \frac{\mathbf{x_i}[d] - \bar{d}}{\text{sd}(d)}$$

Preprocessing – Normalization

Decimal scaling

- Moves the decimal point of the values based on the maximum value
- Scales all values to the interval [-1,1]

For each i :

$$\text{norm}(\mathbf{x}_i[d]) = \frac{\mathbf{x}_i[d]}{10^j}$$

j is chosen such that
the $|d_{max}|$ is of the
form $0.X$ (where X
does not start with 0)



Preprocessing – Normalization

Decimal scaling

- Moves the decimal point of the values based on the maximum value
- Scales all values to the interval [-1,1]

For each i :

$$\text{norm}(\mathbf{x}_i[d]) = \frac{\mathbf{x}_i[d]}{10^j}$$

j is chosen such that
the $|d_{max}|$ is of the
form $0.X$ (where X
does not start with 0)

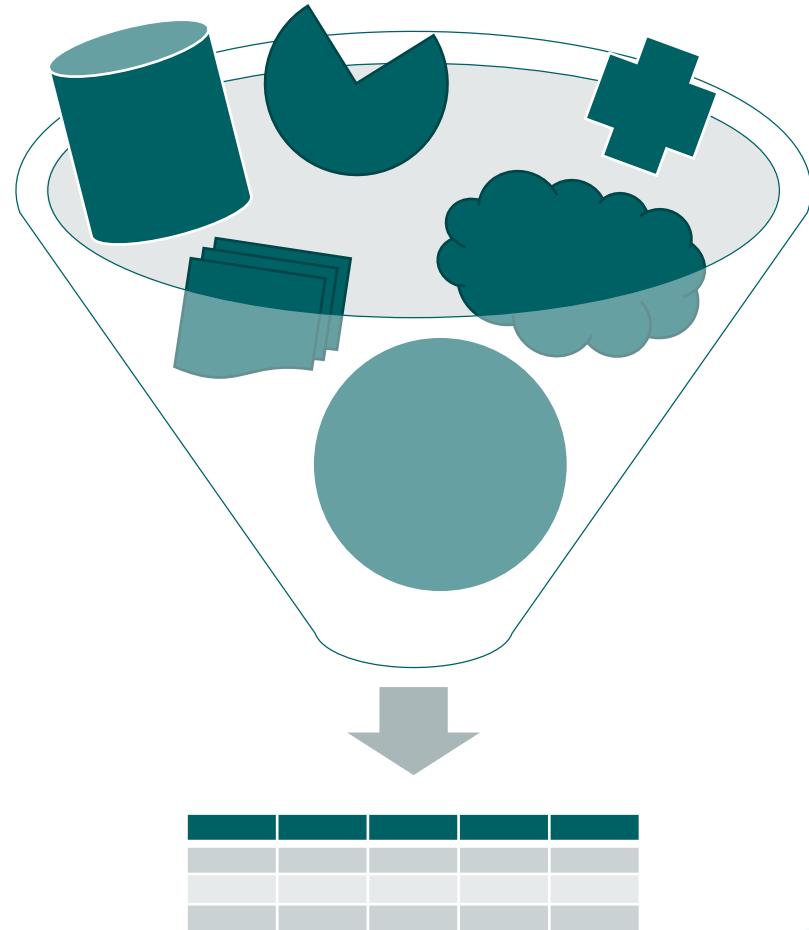
Examples:

- Values in [-877.0, 4.0] are normalized to the range [-0.877, 0.004]
- Values in [0.0003, 0.08] are normalized to the range [0.003, 0.8]



Data Quality & Preprocessing

1. Introduction
2. Missing Values
3. Outliers
4. Semantic Problems
5. Transformation & Normalization
6. **Data Reduction**
7. Conclusion



Preprocessing – Data Reduction

- Analysis may become unfeasible due to size of data
- Goal: reduce the data size but maintain same (or similar) analysis results
- Feature reduction: remove or replace some features
- Instance reduction: remove, replace or aggregate some instances

Feature reduction

ID	f_1	f_2	...	f_D
1				
2				
...				
N				

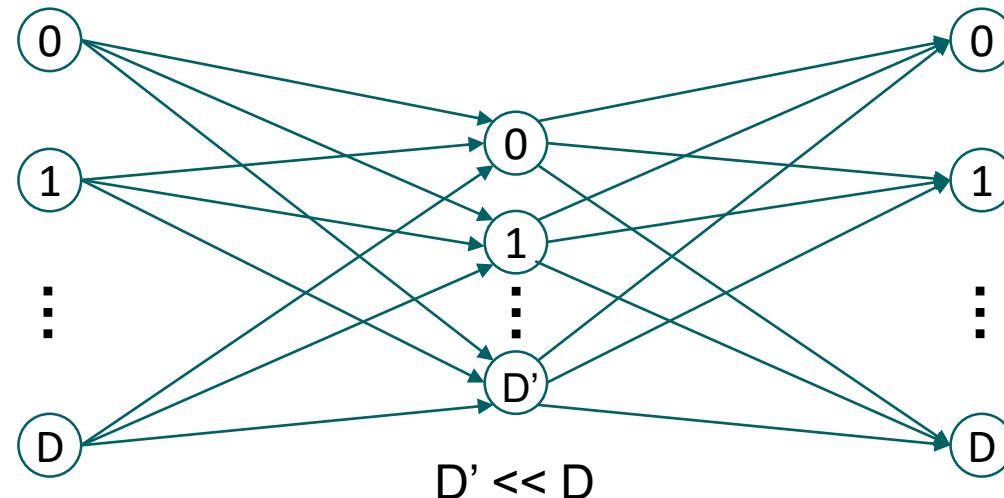
Instance reduction



Preprocessing – Feature Reduction

Projecting data on **fewer** dimensions

- **Autoencoders** (compare text mining): a special type of NN which transforms the input data into a representation with less dimensions (encoding)
- **Principal Component Analysis (PCA)**: represent original features by a few orthogonal (uncorrelated) variables that capture most of the variability



Preprocessing – Feature Reduction

Projecting data on **fewer dimensions**

- [Autoencoders](#) (compare text mining): a special type of NN which transforms the input data into a representation with less dimensions (encoding)
- [Principal Component Analysis \(PCA\)](#): represent original features by a few orthogonal (uncorrelated) variables that capture most of the variability

Feature subset selection: detect and remove **irrelevant/redundant features**

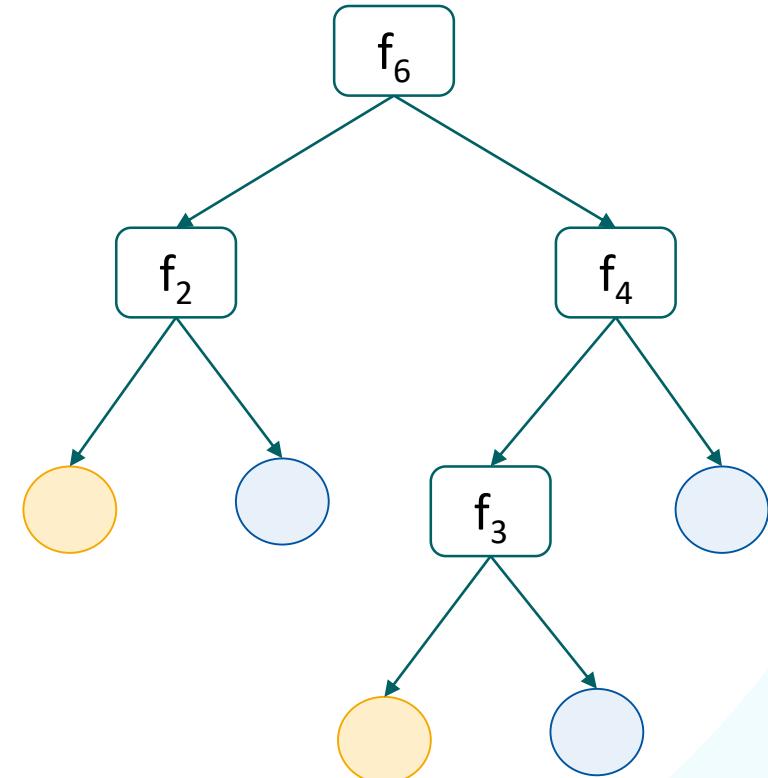
- Use domain knowledge (e.g., remove identifiers)
- Exploit dependencies (e.g., delete features that can be estimated from others using regression)
- Model-driven (e.g. delete features that are not used in a constructed decision tree or, more general, features that can be left out without reducing the quality of the model much)

Preprocessing – Feature Subset Selection

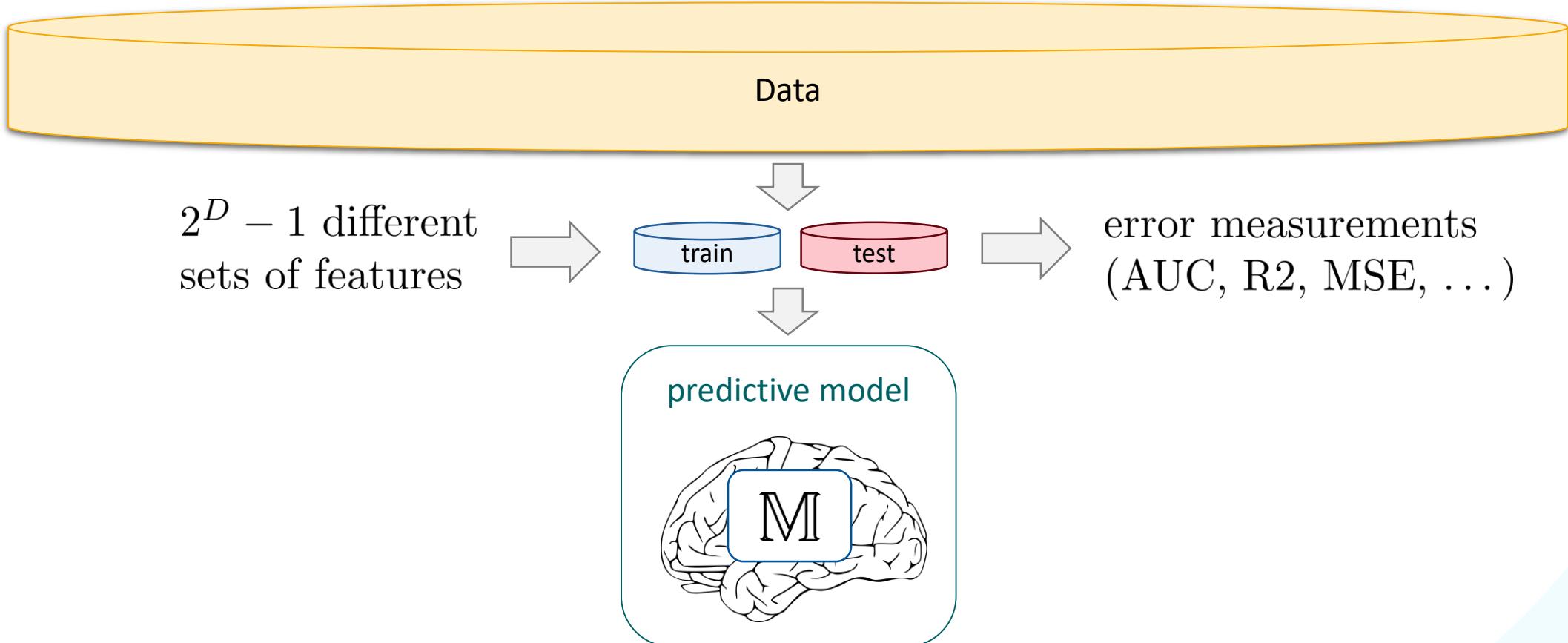
Example

1. Initial features: $f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9$
2. Construct a tree
3. f_2, f_3, f_4, f_6 are relevant (according to the tree)

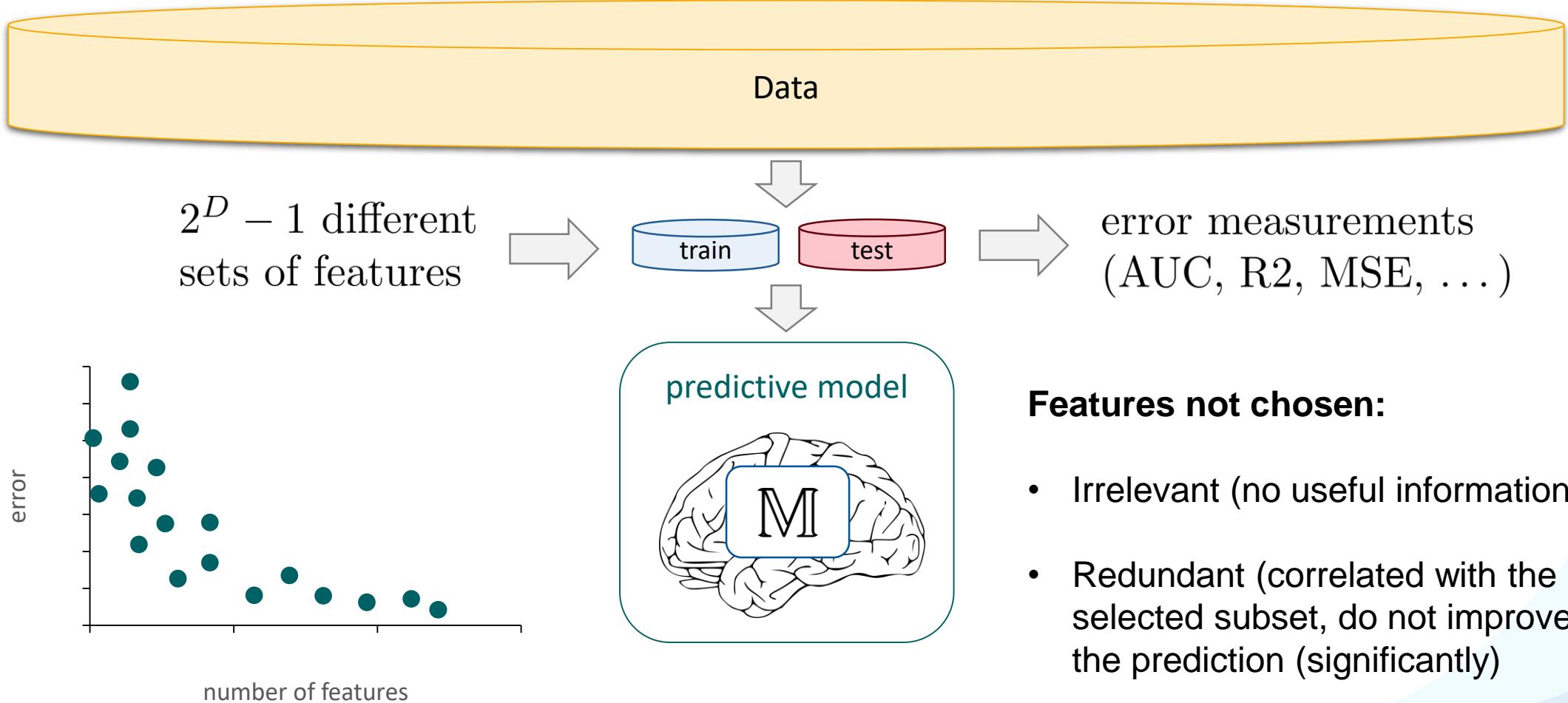
Idea: features correlated with the chosen subset would not improve the classification (significantly) and are therefore not part of the tree.



Preprocessing – Feature Subset Selection

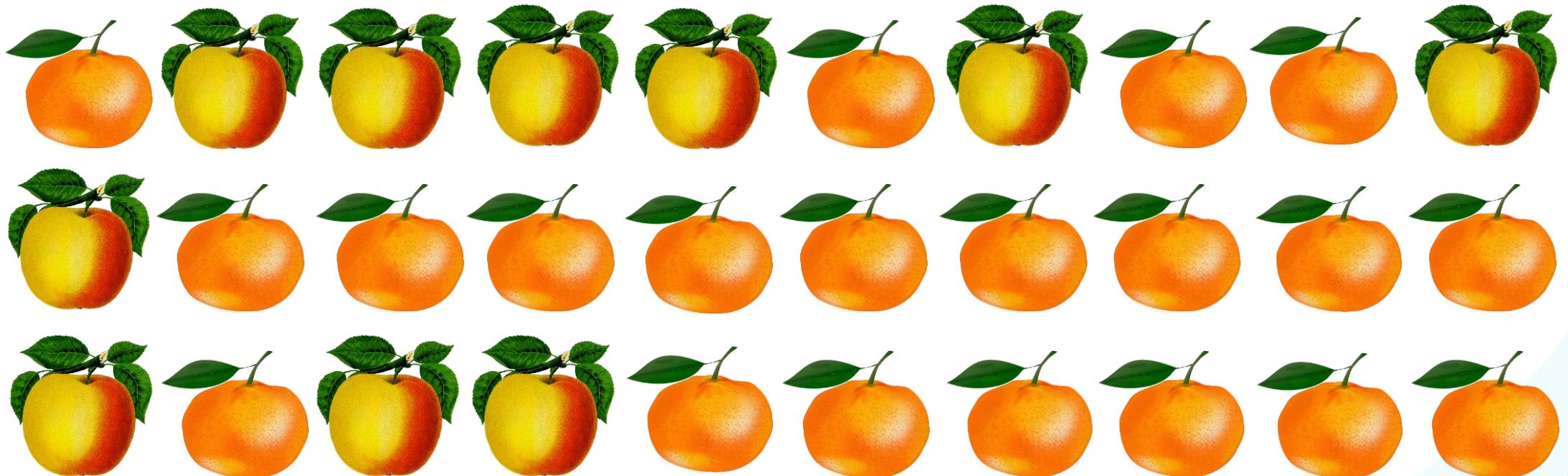


Preprocessing – Feature Subset Selection

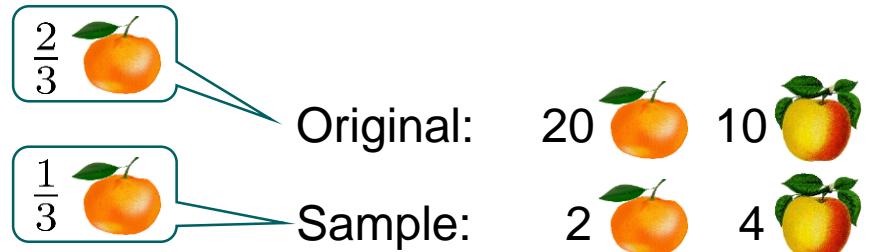


Preprocessing – Sampling

Goals: make the data smaller, remove or introduce biases

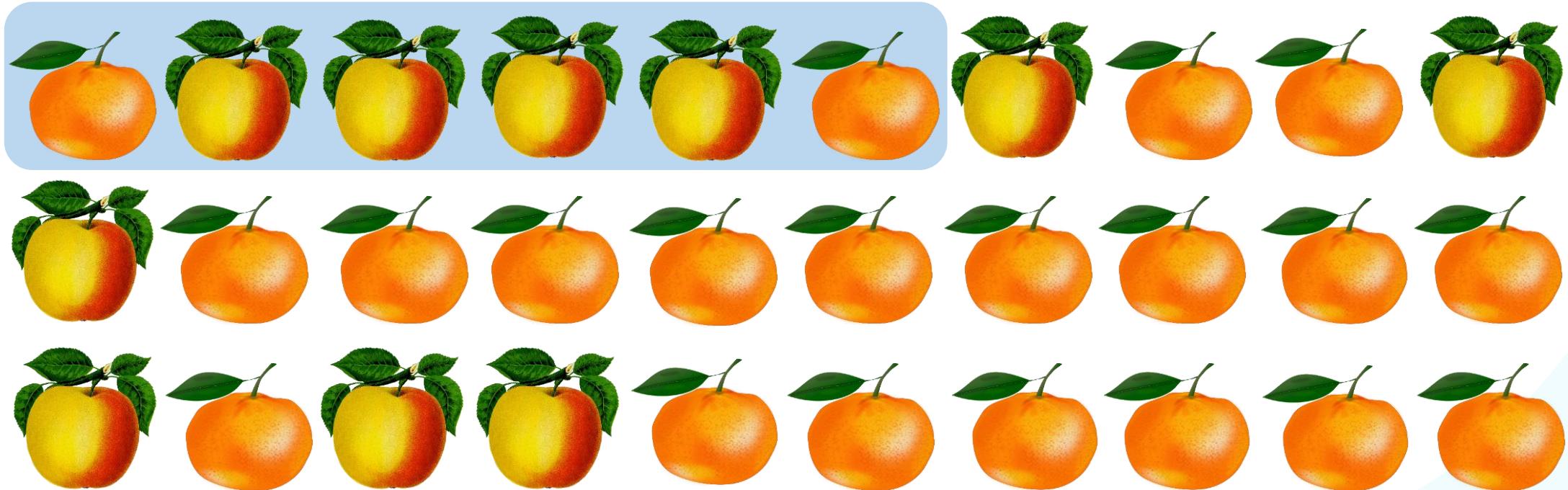


Preprocessing – Sampling

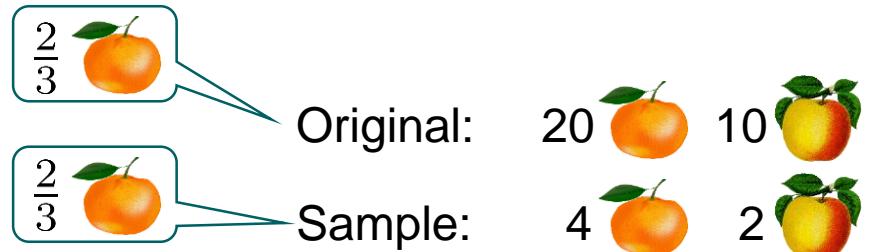


Top sampling:

take the first N instances

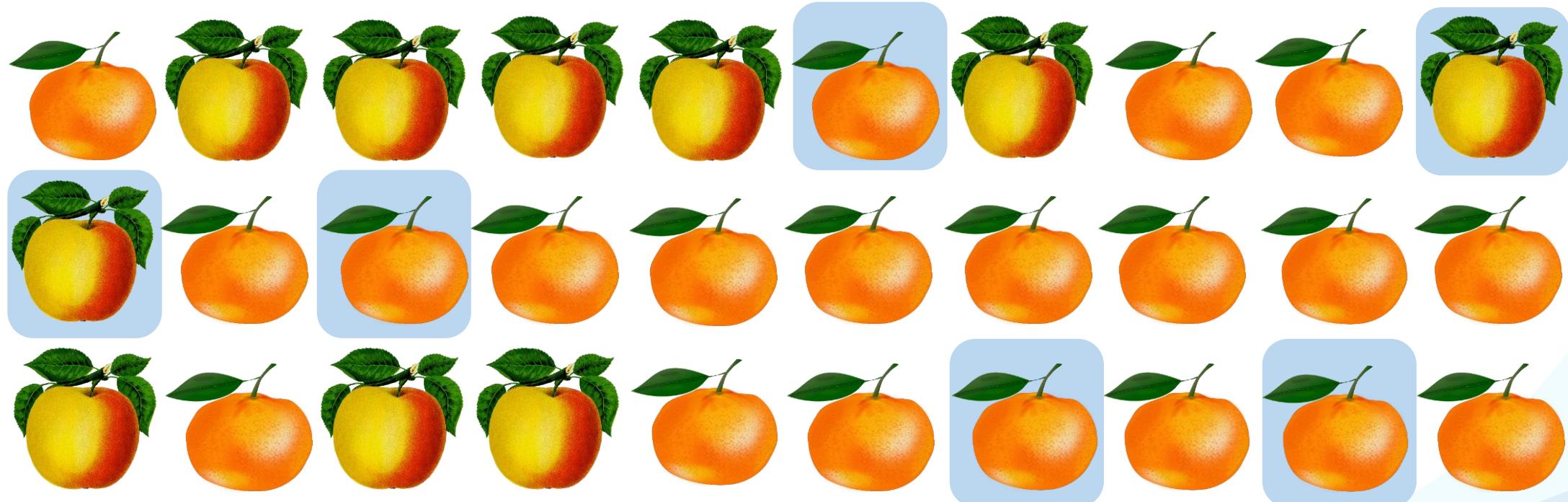


Preprocessing – Sampling

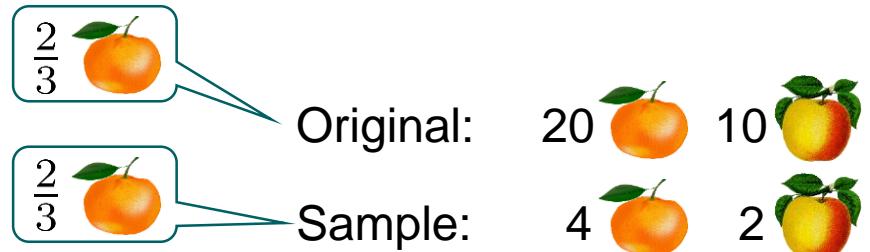


Random sampling:

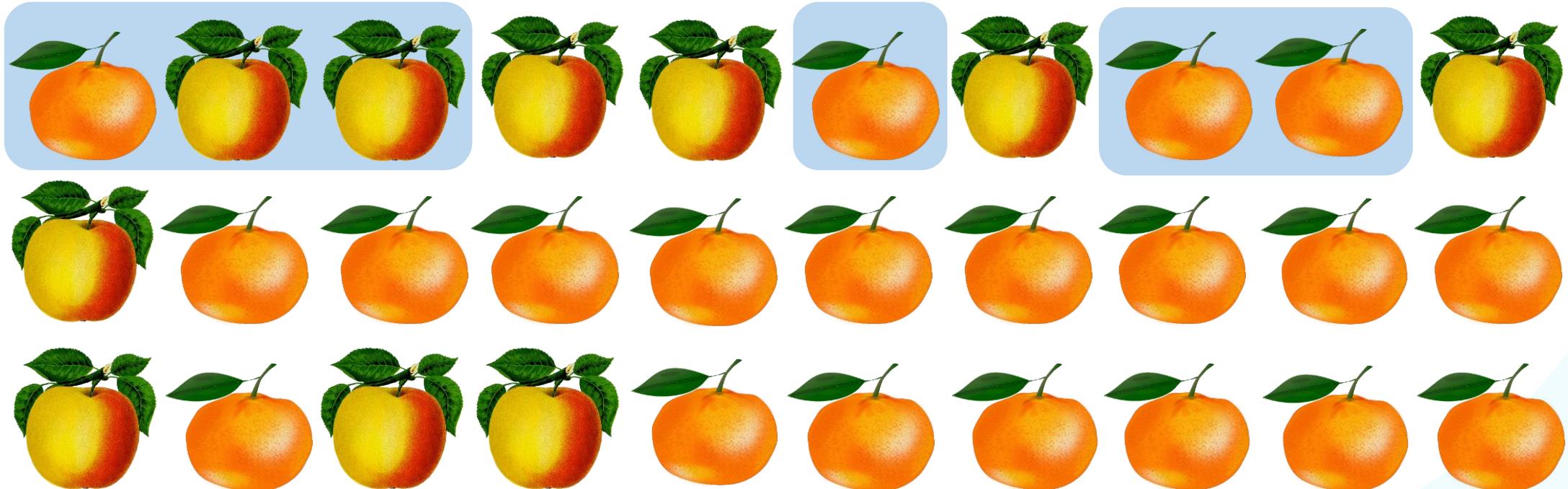
take N arbitrary instances (based on random generator)



Preprocessing – Sampling



Stratified sampling: ensure that relative frequencies are maintained (e.g., take the same percentage from every group)

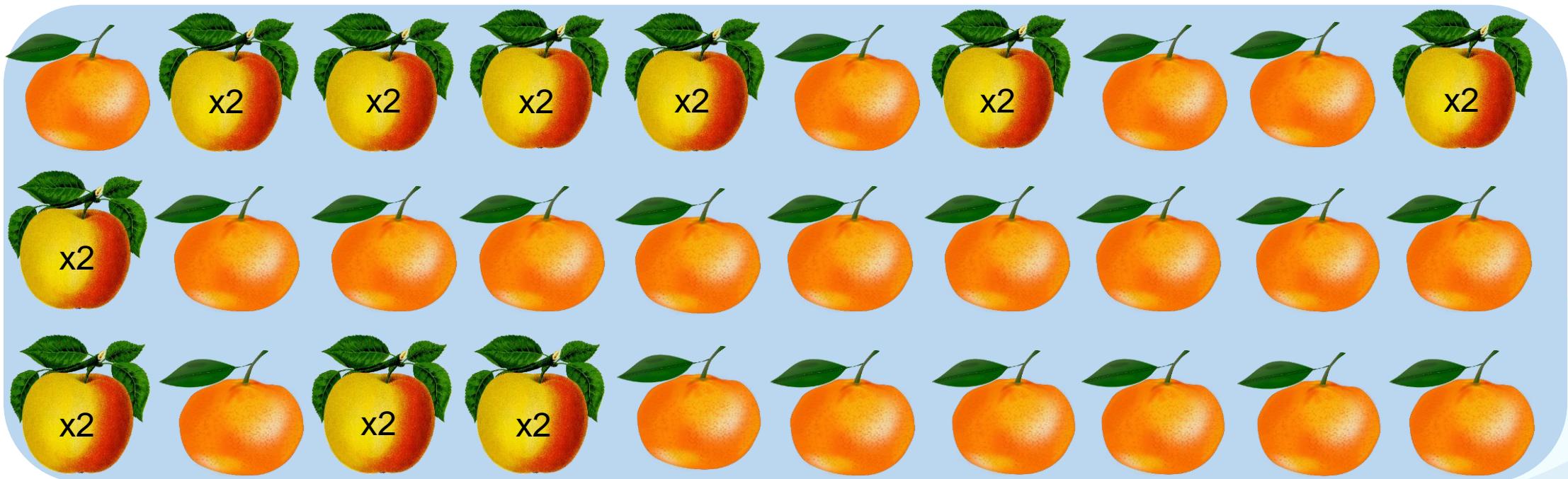


Preprocessing – Sampling

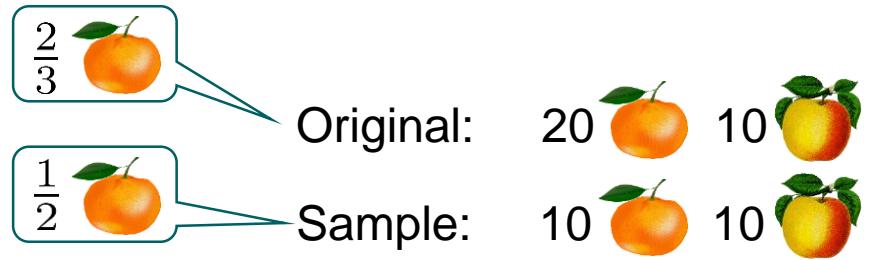


Over-sampling: ensure a certain distribution

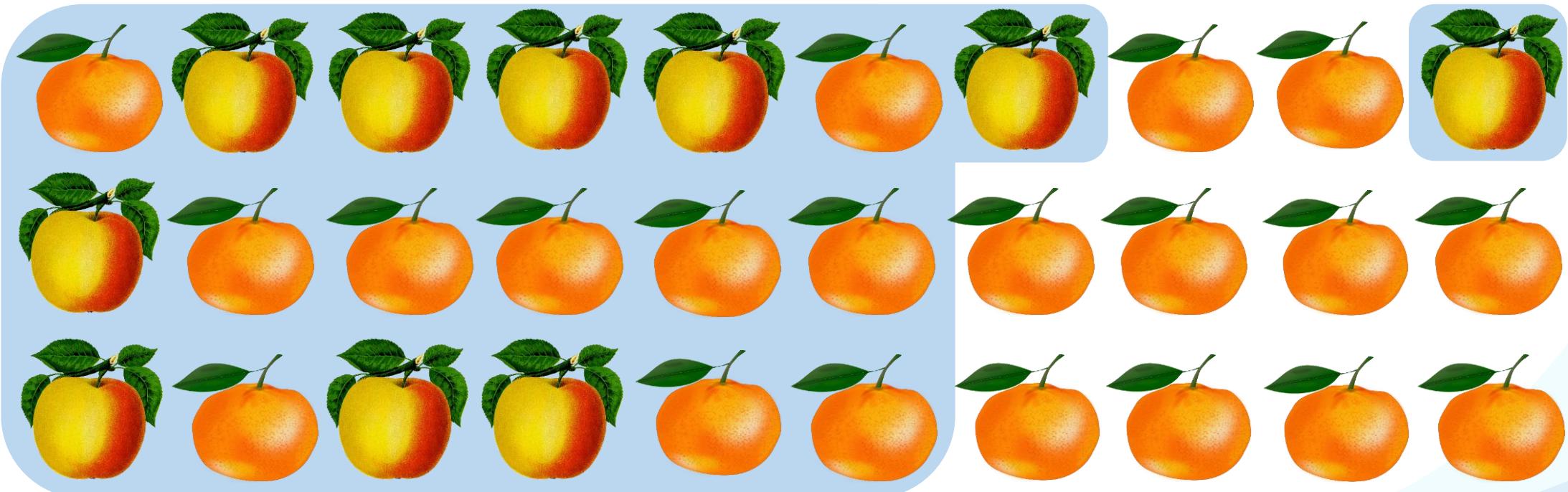
(e.g. equal frequency for each group) by duplicating under-represented instances



Preprocessing – Sampling



Under-sampling: ensure a certain distribution
(e.g. equal frequency for each group) by leaving out over-represented instances



To Conclude



Goal: increase data quality and modify the data to suit the analysis question and applied techniques

Best strategy/solution: depends on the data, context and goal of the analysis

Data quality aspects

- Missing data
- Noise/outliers
- Semantic problems

Garbage in, Garbage out (GIGO)

Data preprocessing

- Transformation
- Normalization
- Data reduction

