Foundations of Social and Cultural Data Analysis

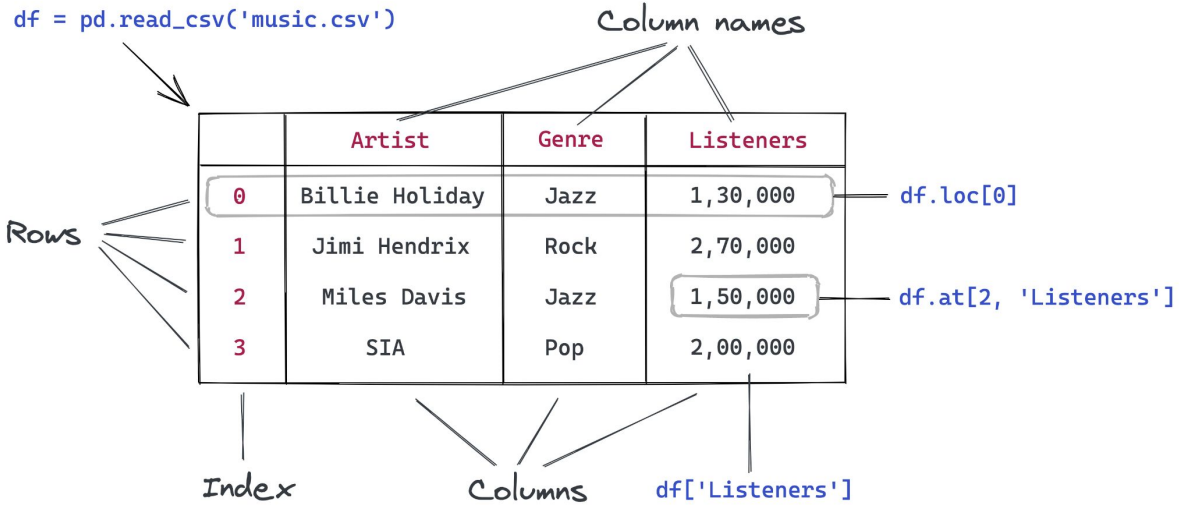
Dr. Nanne van Noord & Dr. Melvin Wevers

Structured <-> Unstructured Data

- **Structured** data is highly organized and formatted in a way that is easy for machines to read and query. It adheres to a predefined schema such as tables with rows and columns. (CSV, JSON, SQL)
- **Unstructured** data lacks a predefined format or structure, making it more complex for computer programs to understand without advanced processing techniques. (Raw text, such as books, or emails)
- Most data is unstructured data. We can add structure by applying processing methods such as Natural Language Processing or Computer Vision.

Tabular Data

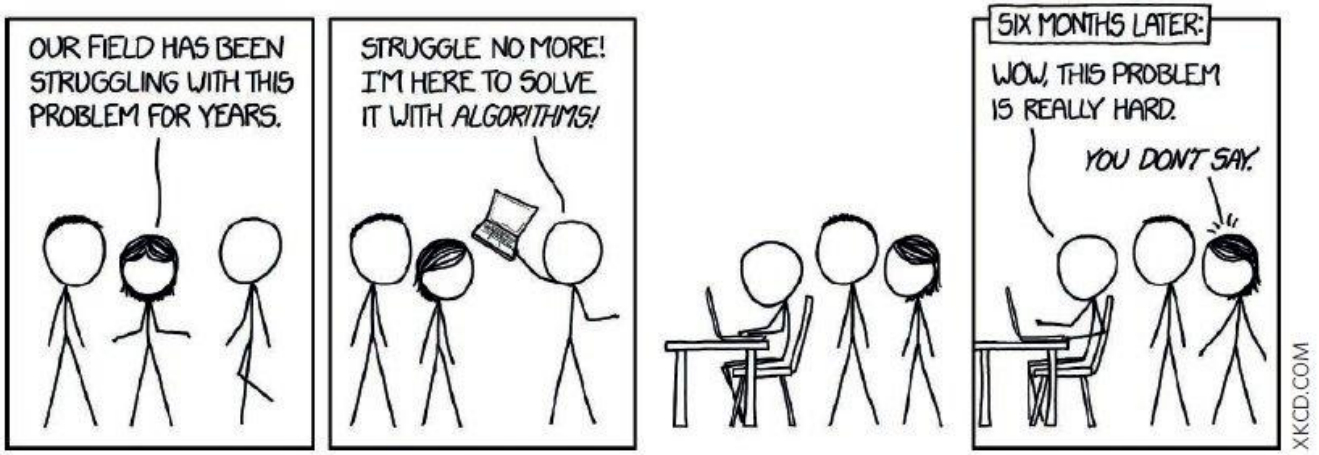
Tabular data refers to data that is organized into tables composed of **rows** and **columns**. Each row represents a single record and each column represents a specific attribute or variable.



Data Transformations

Data Transformations

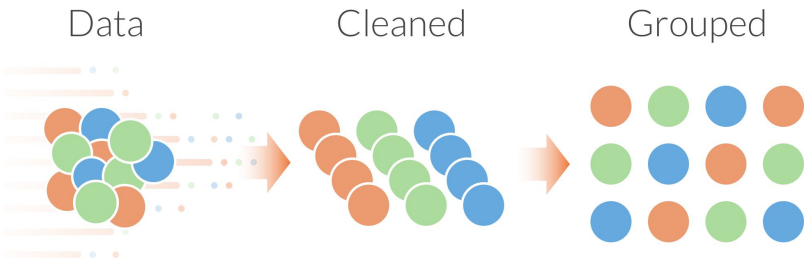
- Real world data is often very messy, and not in a format that readily enables analysis/visualisation.



- https://pandas.pydata.org/getting_started.html

Data Transformations

- Filter
- Aggregate
- Grouping
- Sort
- Concatenate
- Join



Filter

- Reducing a set of data based on certain criteria

Stad	Inwoners
Amsterdam	853,312
Rotterdam	639,587
Den Haag	526,439
Utrecht	344,384
Eindhoven	227,100
Tilburg	214,157

```
df[df['inwoners'] > 500000]
```

Stad	Inwoners
Amsterdam	853,312
Rotterdam	639,587
Den Haag	526,439

Aggregate

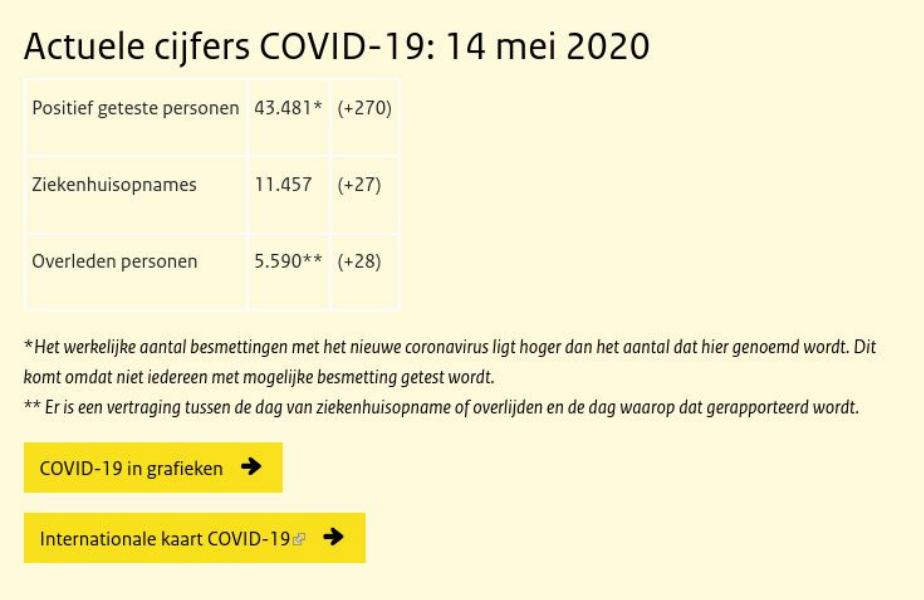
- Reducing a series of data to a single descriptive

City	Population
Amsterdam	853,312
Rotterdam	639,587
Den Haag	526,439
Utrecht	344,384
Eindhoven	227,100
Tilburg	214,157

df[‘inwoners’].mean()

Population	467,496

Aggregated data



Actuele cijfers COVID-19: 14 mei 2020

Positief geteste personen	43.481*	(+270)
Ziekenhuisopnames	11.457	(+27)
Overleden personen	5.590**	(+28)

*Het werkelijke aantal besmettingen met het nieuwe coronavirus ligt hoger dan het aantal dat hier genoemd wordt. Dit komt omdat niet iedereen met mogelijke besmetting getest wordt.

** Er is een vertraging tussen de dag van ziekenhuisopname of overlijden en de dag waarop dat gerapporteerd wordt.

COVID-19 in grafieken →

Internationale kaart COVID-19 →

https://www.rivm.nl/coronavirus-covid-19/actueel

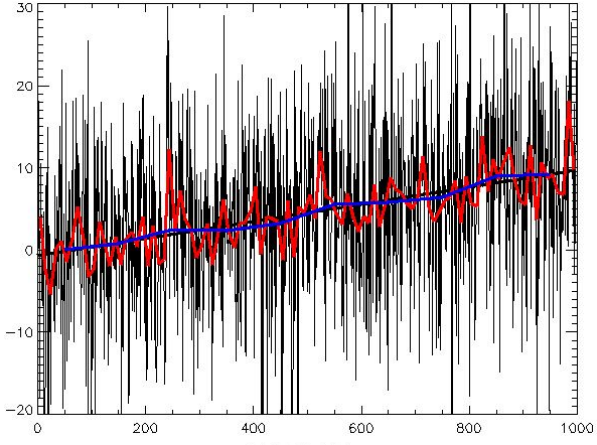
Aggregation: Descriptive statistics

Aggregating is a way of summarising data

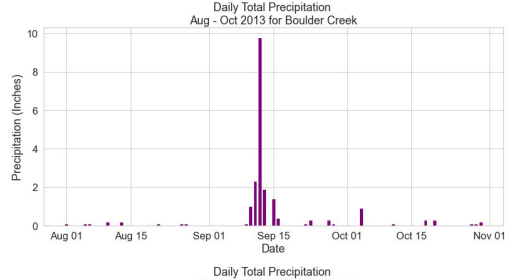
Measurement scale	**Descriptive Statistic**
Nominal	Frequency Relative Frequency (proportion) Percentage Mode
Ordinal	Frequency Relative Frequency (proportion) Percentage Mode Median
Interval and Ratio	Central tendency: mean, median, mode Range, standard deviation Interquartile range (IQR) Skewness, Kurtosis

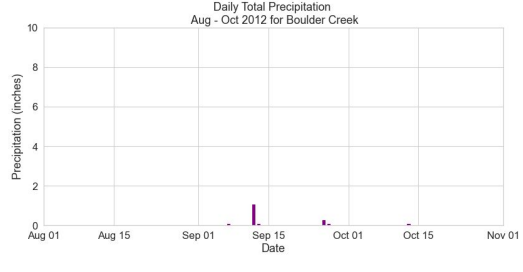
Plotting over Time: Time Series

Data points indexed in time order



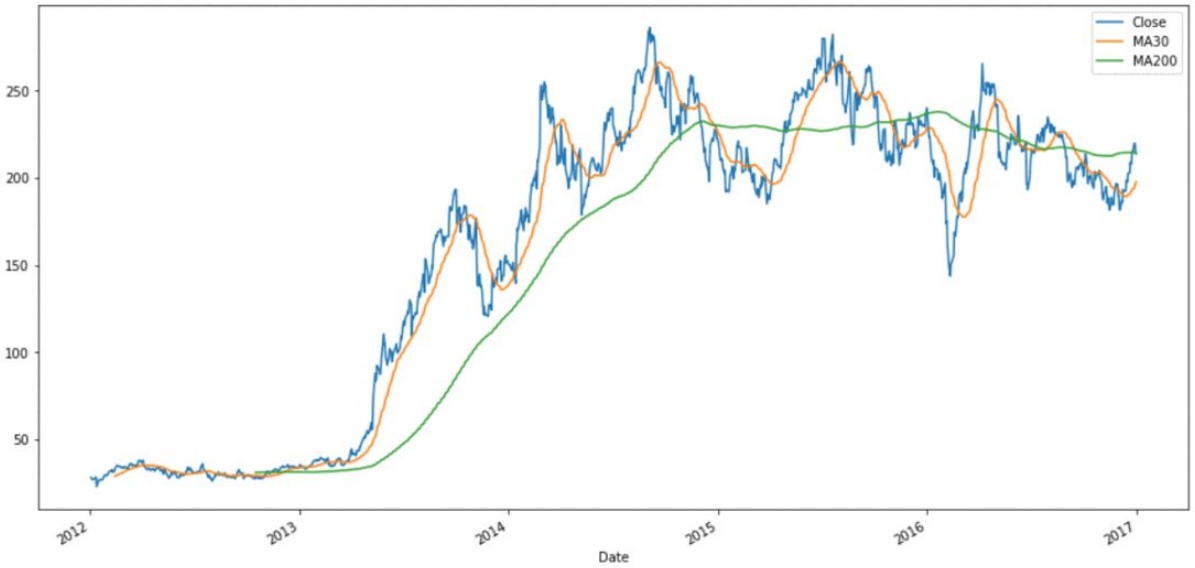

```
import pandas as pd

df = pd.read_csv('data.csv')

df['date'] = pd.to_datetime(df['date'])# recognize data format

df.set_index('date', inplace=True)# set dates as index

df.resample('M').mean #get monthly averages

df['2020-01':'2020-06'] # get data from january to june 2020
```





Plotting over Time: Rolling Mean

- Rolling mean: A moving average that helps smooth out time series data.
- Why? Easier to spot trends
- `rolling_mean = df['temperature'].rolling(window=7).mean()`
-



Chaining transformations

City	Population
Amsterdam	853,312
Rotterdam	639,587
Den Haag	526,439
Utrecht	344,384
Eindhoven	227,100
Tilburg	214,157

```
df[df['Population'] > 500000].mean()
```

Population	673,112

Pandas aggregation methods:
https://pandas.pydata.org/pandas-docs/stable/api.html#api-dataframe-stats

Grouping

- Dividing a dataset into coherent ‘subdatasets’
- Followed by an aggregation

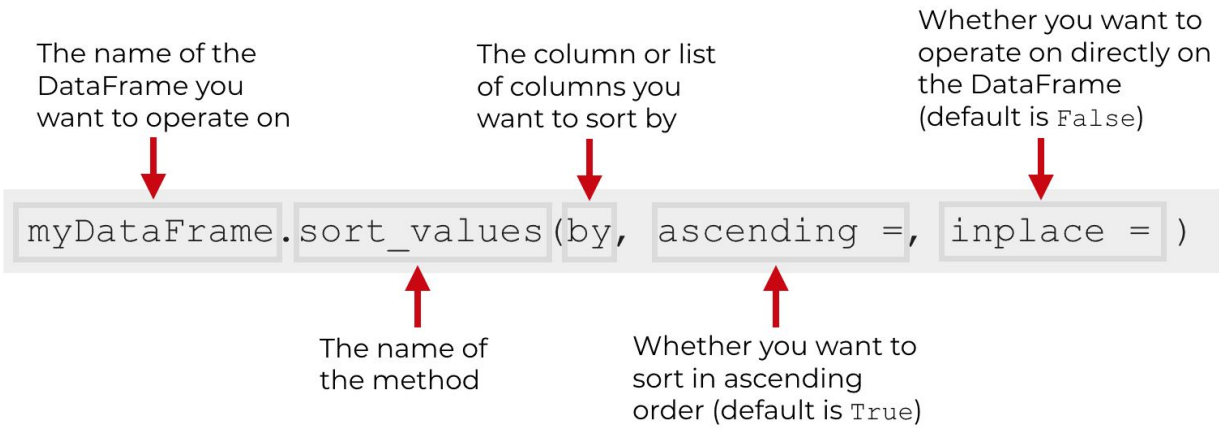
City	Province	Population
Amsterdam	Noord-Holland	853,312
Rotterdam	Zuid-Holland	639,587
Den Haag	Zuid-Holland	526,439
Utrecht	Utrecht	344,384
Eindhoven	Noord-Brabant	227,100
Tilburg	Noord-Brabant	214,157

df.groupby(‘Province’).sum()

Province	Population
Noord-Holland	853,312
Zuid-Holland	1,166,026
Utrecht	344,384
Noord-Brabant	441,257

Sort

- Ordering the values in a series
- Useful for data inspection and presentation
- .sort_values() in pandas



Concatenate

- Combining multiple datasets which have the same variables

City	Population
Amsterdam	853,312
Rotterdam	639,587
Den Haag	526,439

+

City	Population
Utrecht	344,384
Eindhoven	227,100
Tilburg	214,157

=

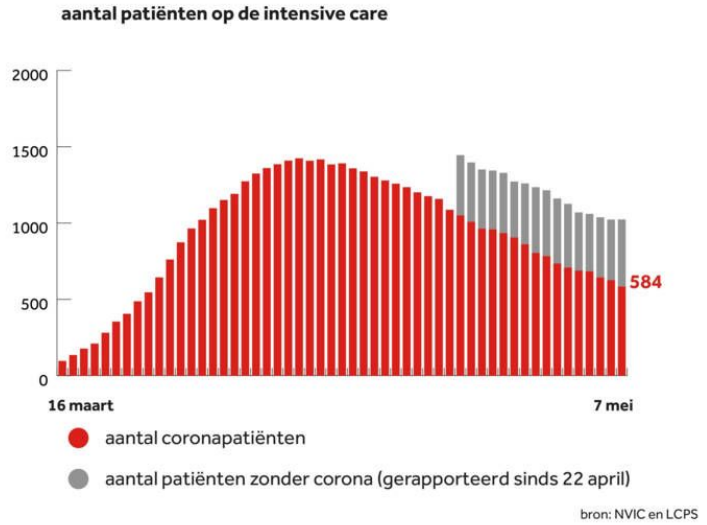
City	Population
Amsterdam	853,312
Rotterdam	639,587
Den Haag	526,439
Utrecht	344,384
Eindhoven	227,100
Tilburg	214,157

Concatenated data

Month	Case ID	Activity	Timestamp
July	09-45-568	Receive Order	01.07.2018
July	09-45-568	Prepare Order	02.07.2018
...	...	...	...

Month	Case ID	Activity	Timestamp
August	09-55-789	Receive Order	01.08.2018
August	09-45-568	Complete Order	02.08.2018
...	...	...	...

Combine monthly records to form the basis of the annual report

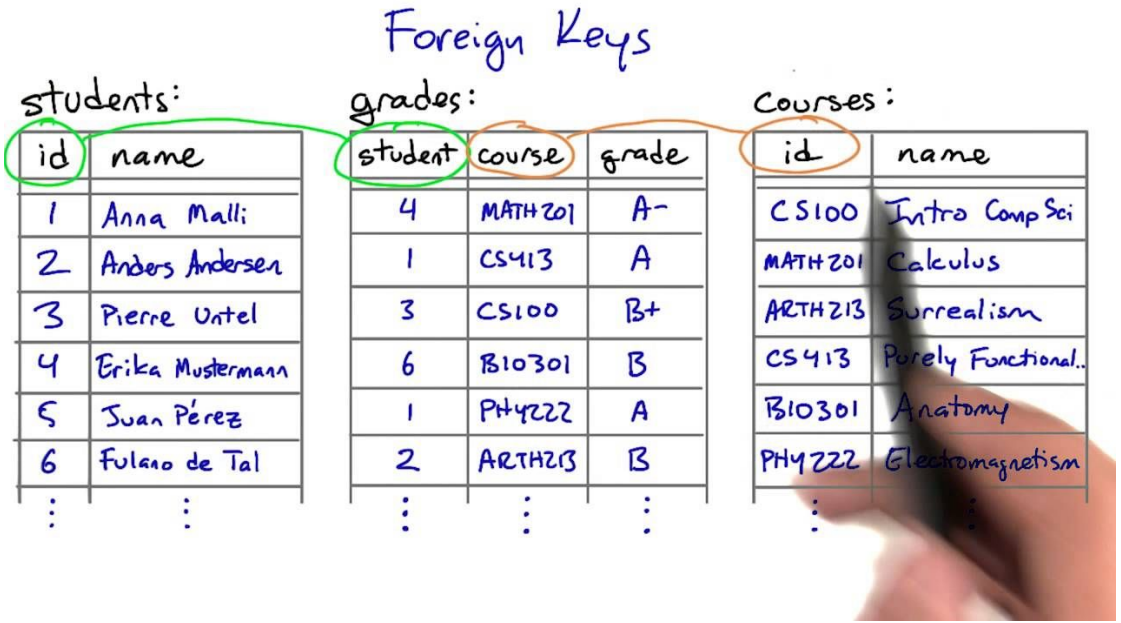


Reported daily. Concatenated to form overview

Join

- Combining two datasets which describe an overlapping set of instances, with different features
- Powerful method to combine different data sources
- Common in relational databases

Keys



https://www.youtube.com/watch?v=fnbLMcd0FGQ

Inner Join

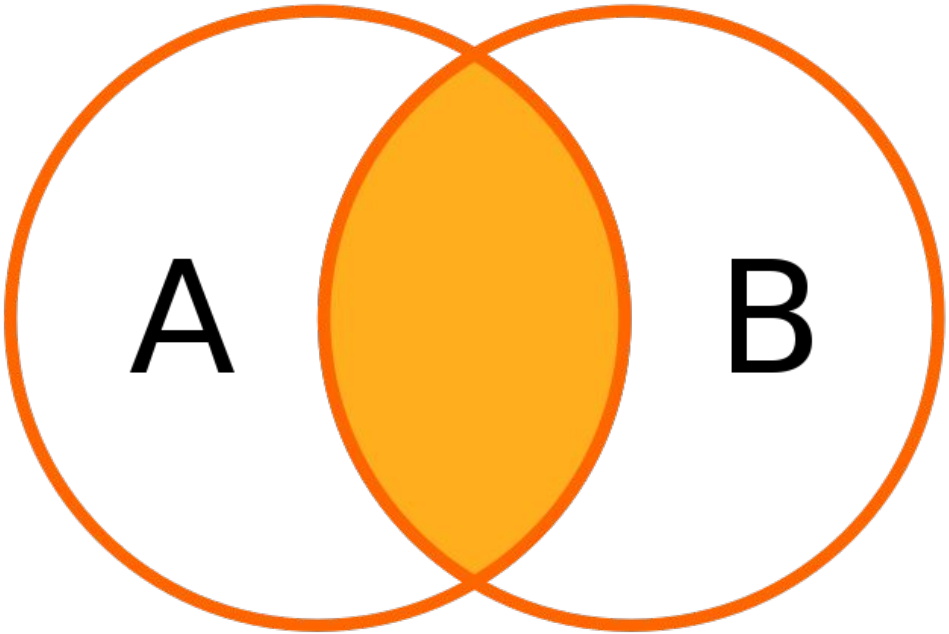


Image: wikipedia.org

Inner Join

City	Population
Amsterdam	853,312
Rotterdam	639,587
Den Haag	526,439
Utrecht	344,384
Eindhoven	227,100
Tilburg	214,157

+

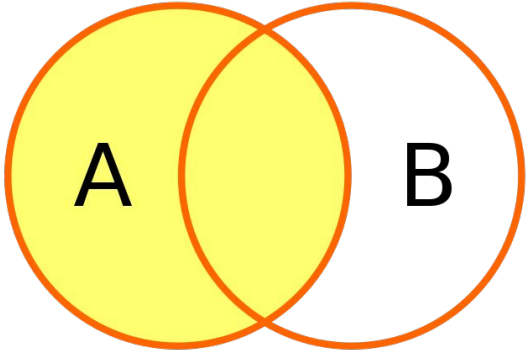
City	Air quality
Amsterdam	42.4
Rotterdam	40.9
Den Haag	41.1
Utrecht	41.4
Eindhoven	43.8
Zwolle	40.9

=

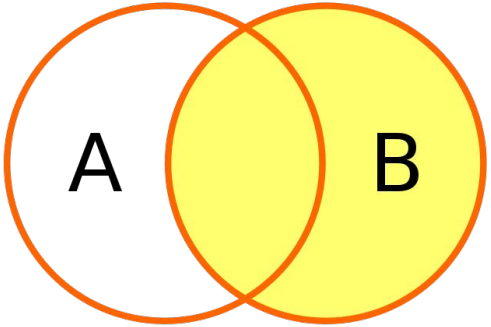
City	Population	Air quality
Amsterdam	853,312	42.4
Rotterdam	639,587	40.9
Den Haag	526,439	41.1
Utrecht	344,384	41.4
Eindhoven	227,100	43.8

Outer Join

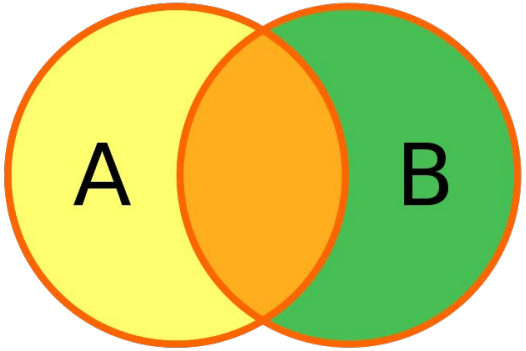
Left (outer) join



Right (outer) join



Full outer join



Images: wikipedia.org

Left Join

City	Population
Amsterdam	853,312
Rotterdam	639,587
Den Haag	526,439
Utrecht	344,384
Eindhoven	227,100
Tilburg	214,157

+

City	Air quality
Amsterdam	42.4
Rotterdam	40.9
Den Haag	41.1
Utrecht	41.4
Eindhoven	43.8
Zwolle	40.9

=

City	Population	Air quality
Amsterdam	853,312	42.4
Rotterdam	639,587	40.9
Den Haag	526,439	41.1
Utrecht	344,384	41.4
Eindhoven	227,100	43.8
Tilburg	214,157	N/A

Right Join

City	Population
Amsterdam	853,312
Rotterdam	639,587
Den Haag	526,439
Utrecht	344,384
Eindhoven	227,100
Tilburg	214,157

+

City	Air quality
Amsterdam	42.4
Rotterdam	40.9
Den Haag	41.1
Utrecht	41.4
Eindhoven	43.8
Zwolle	40.9

=

City	Population	Air quality
Amsterdam	853,312	42.4
Rotterdam	639,587	40.9
Den Haag	526,439	41.1
Utrecht	344,384	41.4
Eindhoven	227,100	43.8
Zwolle	N/A	40.9

Full outer Join

City	Population
Amsterdam	853,312
Rotterdam	639,587
Den Haag	526,439
Utrecht	344,384
Eindhoven	227,100
Tilburg	214,157

+

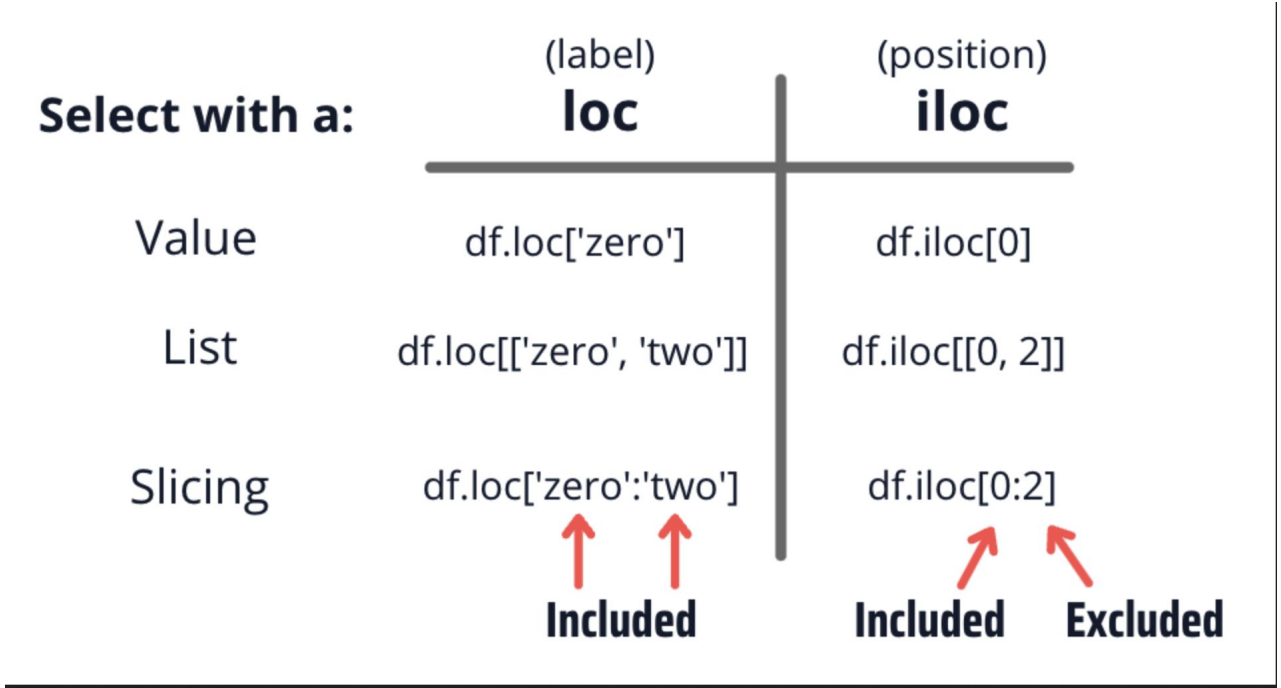
City	Air quality
Amsterdam	42.4
Rotterdam	40.9
Den Haag	41.1
Utrecht	41.4
Eindhoven	43.8
Zwolle	40.9

=

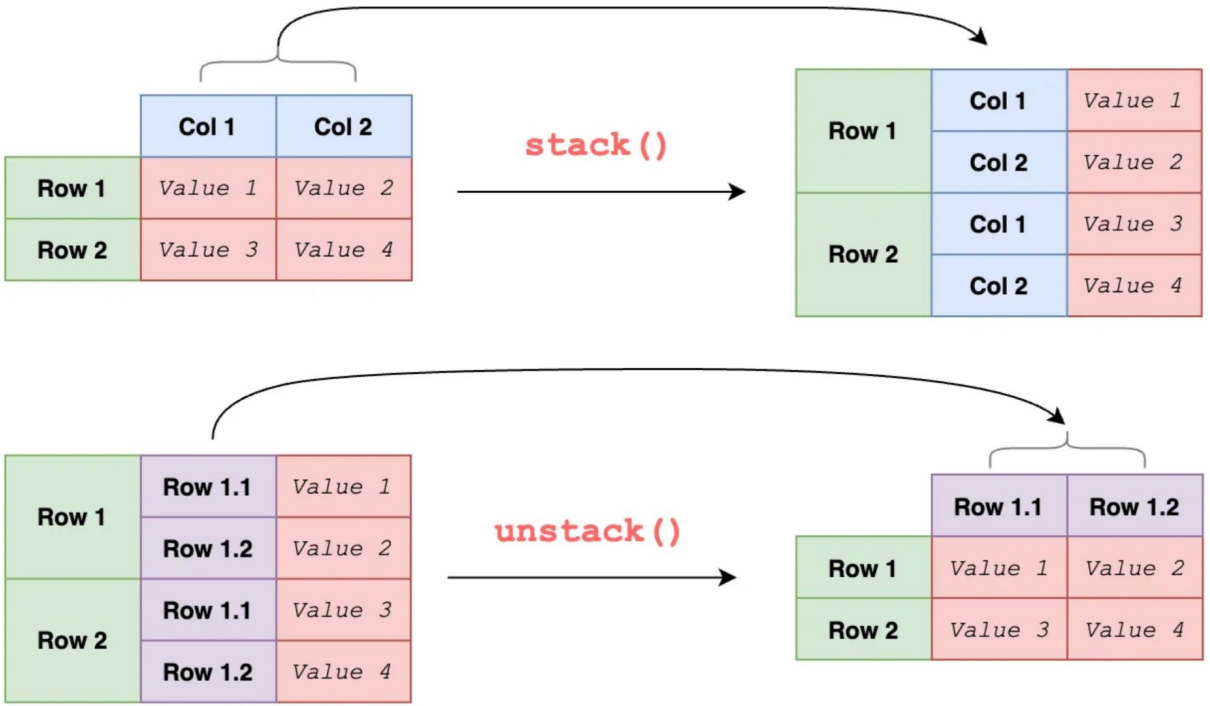
City	Population	Air quality
Amsterdam	853,312	42.4
Rotterdam	639,587	40.9
Den Haag	526,439	41.1
Utrecht	344,384	41.4
Eindhoven	227,100	43.8
Tilburg	214,157	N/A
Zwolle	N/A	40.9

Indexing: iloc vs loc

- iloc = Integer-based indexing
- Loc = label-based indexing



Stacking and Unstacking



Goal

- Transformations are not a goal on their own. We perform them to prepare the data for analysis and/or visualisation.

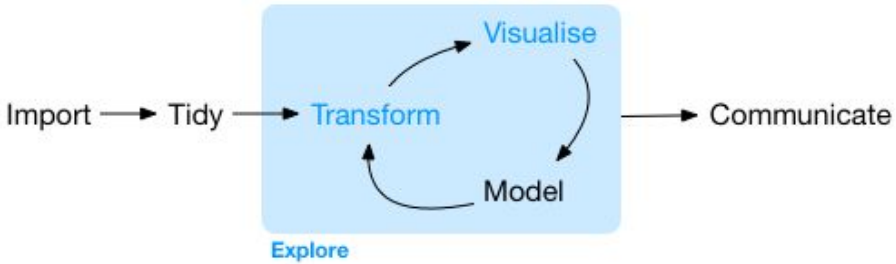
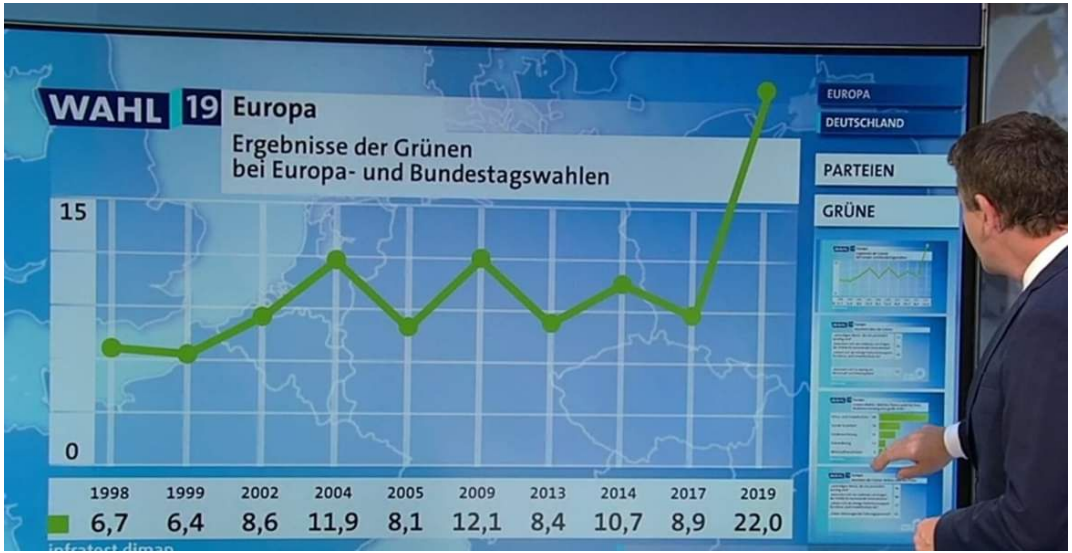


Image: https://aldosolari.github.io/DM/

Data Pre-processing

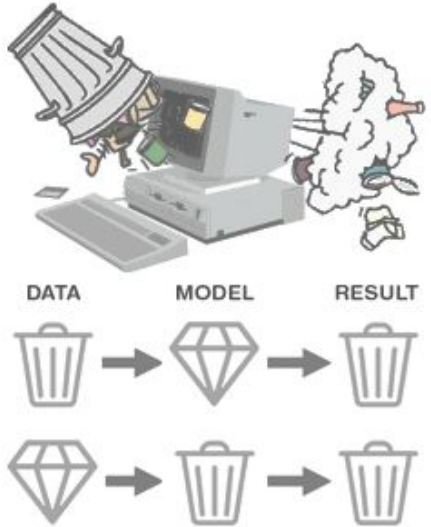
- Data pre-processing
  - Selection
  - Importing
  - Cleaning
  - Normalisation

Incorrectly prepared data



Garbage In, Garbage Out

https://medium.com/nyc-design/gigo-garbage-in-garbage-out-concept-for-ux-research-7e3f50695b82



https://thedailyomnivore.net/2015/12/02/garbage-in-garbage-out/

Google Books is heavily biased

TECHNOLOGY

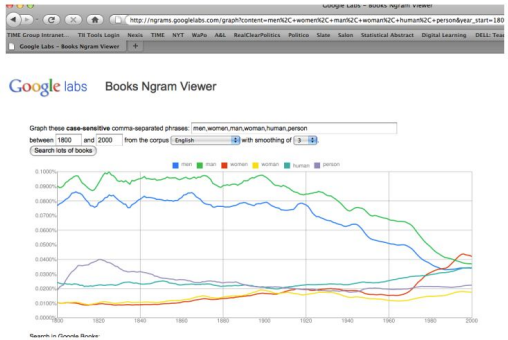
Culturomics: Google Tracks Culture Trends Through Books

By Alexandra Silver | Dec. 17, 2010

Share Pin it Read Later

Want to know if the Beatles really were bigger than Jesus? Then head over to the Google Books NGram Viewer.

Using this tool, you can compare words and phrases from several languages across centuries by charting their frequency in books, an exercise that can shed light on everything from rock stars and religious figures to gender studies and epidemiology. (An n-gram is a sequence of words or numbers, i.e. "Civil War" or "1600 Pennsylvania Avenue"). Simply enter one word, or several. Then click the wildly understated "Search lots of books." Observe and marvel.



A screen shot of the Google Books Ngram Viewer

- "We find that only the English Fiction data set from the second version of the corpus is not heavily affected by professional texts" (Pechenick et al. 2015)
- The corpus only encodes a small-scale kind of popularity, i.e. frequency of n-grams with all books given equal importance in their year of publication

Data Selection: Gathering

Before we can import data we need to ask:

- What data do I need? → Follows from research question
  - What should it contain
  - How should it be structured
- Where can I obtain this data?
- Is the data available representative?

Data Selection: Reducing

- What's in my data (EDA) and do I need it all?
  - Out of scope
  - Privacy or ethical considerations
  - Practical reasons
- Reduce by
  - Filtering
  - Removing attributes

Importing data

- Highly dependent on format and software
- Dataset might require conversion before we can even load it
  - File format conversion
  - Structural conversion
    - Document-based → Table

Data Cleaning

The goal of data cleaning is to ensure the quality of our dataset

- Time consuming
- Manual
- Uncertain

Data Quality

- Accuracy: is the data correct?
- Completeness: does it include all the relevant data?
- Consistency: is my data equal across sources?
- Validity: does it adhere to my requirements?
- Uniqueness: is there duplicate data?

https://liliendahl.com/2019/04/20/a-data-quality-mind-map/

Data cleaning tasks

- Data inspection (EDA)
- Fix structural errors
- De-duplicate
- Remove outliers
- Deal with missing values

De-duplicating

- Removing duplicate records
- Removing records that concern the same ‘key’

City	Population	Year
Amsterdam	853,312	2018
Rotterdam	639,587	2018
Den Haag	526,439	2018
Utrecht	344,384	2018
Eindhoven	227,100	2018
Amsterdam	862,965	2019
Utrecht	344,384	2018

Structural errors

- Typos
  - “Information Visuaisation”, “Information visualisation”, “Information Visualisation”
- Mislabeled or inconsistent labeling
  - “N/A”, “Not applicable”, “NA”
  - “M”, “Female”, “F”, “male”, “Male”
- Duplicate attribute names
  - Merging datasets that each have a ‘year’ column

Dealing with missing values

- Listwise deletion
- Column deletion
- Imputation

City	Population	Air quality
Amsterdam	853,312	42.4
Rotterdam	639,587	40.9
Den Haag	526,439	41.1
Utrecht	344,384	41.4
Eindhoven	227,100	43.8
Tilburg	214,157	
Zwolle		40.9

Listwise deletion

Deleting records with missing values

City	Population	Air quality
Amsterdam	853,312	42.4
Rotterdam	639,587	40.9
Den Haag	526,439	41.1
Utrecht	344,384	41.4
Eindhoven	227,100	43.8
Tilburg	214,157	
Zwolle		40.9

Amsterdam	853,312	42.4
Rotterdam	639,587	40.9
Den Haag	526,439	41.1
Utrecht	344,384	41.4
Eindhoven	227,100	43.8

Column deletion

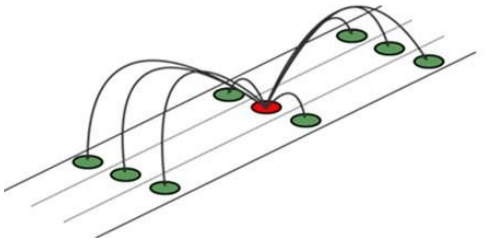
Deleting attributes with missing values

City	Population	Air quality
Amsterdam	853,312	42.4
Rotterdam	639,587	40.9
Den Haag	526,439	41.1
Utrecht	344,384	41.4
Eindhoven	227,100	43.8
Tilburg	214,157	

City	Population
Amsterdam	853,312
Rotterdam	639,587
Den Haag	526,439
Utrecht	344,384
Eindhoven	227,100
Tilburg	214,157

Missing values: Imputation

- Substituting a missing value for a ‘plausible’ value
  - With a constant (U / 0)
  - Mean / median substitution
  - Random sampling (from distribution)
  - Modeling



Impute: constant

Replacing missing with an indicative value

City	Population	Air quality
Amsterdam	853,312	42.4
Rotterdam	639,587	40.9
Den Haag	526,439	41.1
Utrecht	344,384	41.4
Eindhoven	227,100	43.8
Tilburg	214,157	NaN
Zwolle	NaN	40.9

np.nanmean()

City	Population	Air quality
Amsterdam	853,312	42.4
Rotterdam	639,587	40.9
Den Haag	526,439	41.1
Utrecht	344,384	41.4
Eindhoven	227,100	43.8
Tilburg	214,157	-1
Zwolle	-1	40.9

Impute: Mean substitution

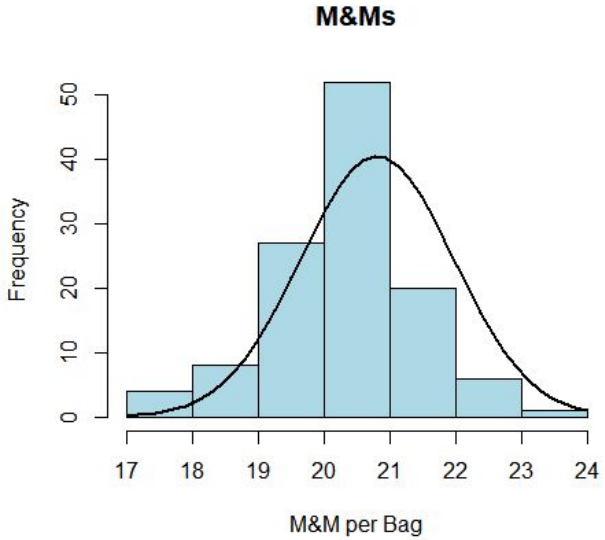
Replacing missing values with mean value of series

City	Population	Air quality
Amsterdam	853,312	42.4
Rotterdam	639,587	40.9
Den Haag	526,439	41.1
Utrecht	344,384	41.4
Eindhoven	227,100	43.8
Tilburg	214,157	
Zwolle		40.9

City	Population	Air quality
Amsterdam	853,312	42.4
Rotterdam	639,587	40.9
Den Haag	526,439	41.1
Utrecht	344,384	41.4
Eindhoven	227,100	43.8
Tilburg	214,157	**41,75**
Zwolle	**467,496**	40.9

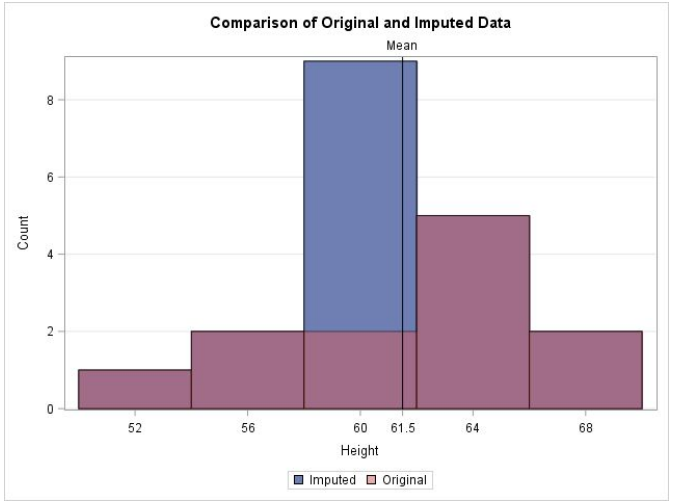
Impute: Random sampling

If we have a known distribution that fits the variable well we can sample from this distribution to generate a ‘plausible’ value.



Risk of imputation

- Likely to introduce bias to your dataset
  - Reflected in models
  - Might lead to incorrect conclusions

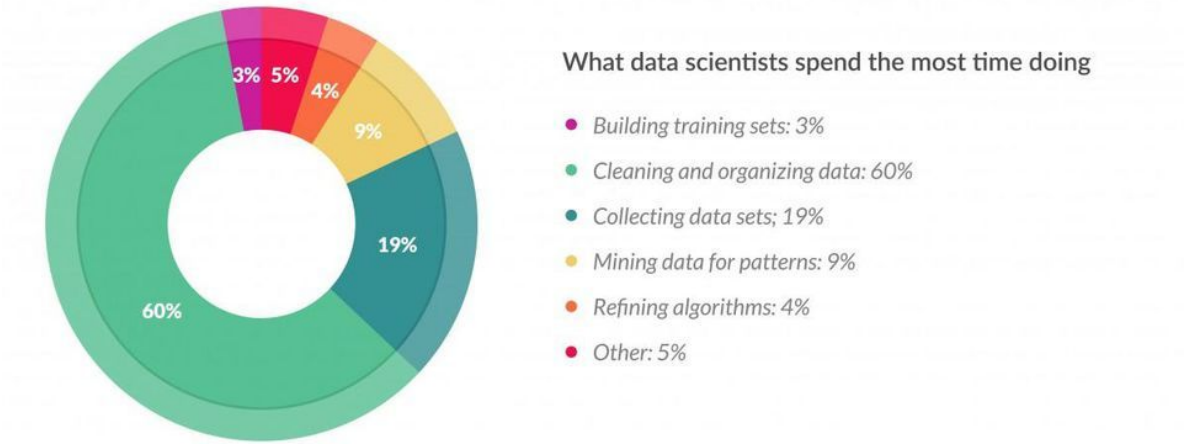


Normalisation

- Mainly useful for modeling, but has use for visualisation too
  - Min-max scaling: $x' = \frac{x - min(x)}{max(x) - min(x)}$
    - Scales values between 0 and 1
  - Zero mean: $x' = x - \mu$
    - Centers values around 0
  - Z-score $x' = \frac{x - \mu}{\sigma}$
    - Centers values around 0
    - Score represents how many standard deviations it is from the mean

Wrap up

- Pre-processing gets our data ready for further analysis
- Familiarises you with the data
- Time consuming



https://jinchuika.com/en/post/1-preprocessing-part-1/

Pandas readings

- Intro to pandas objects:
  https://jakevdp.github.io/PythonDataScienceHandbook/03.01-introducing-pandas-objects.html
- Data indexing and selection:
  https://jakevdp.github.io/PythonDataScienceHandbook/03.02-data-indexing-and-selection.html
- Combining datasets: Merge and join:
  https://jakevdp.github.io/PythonDataScienceHandbook/03.07-merge-and-join.html
- Aggregation and grouping:
  https://jakevdp.github.io/PythonDataScienceHandbook/03.08-aggregation-and-grouping.html