PYTHON / PANDAS FOR DATA ANALYSIS

Sander van den Oord - Data Scientist

SIGMADATA Learning

Wat we gaan doen de komende 2 dagen

- Introductie:
  - waar werk je nu,
  - met welke software,
  - gebruik je nu Python,
  - en: wat wil je vandaag leren

- Highover uitleg Python, Pandas en belangrijkste Python data libraries
- Intro Jupyter Notebook + Python refresher + opstarten VM
- Data inlezen ---> pd.read_csv()
- Data inspectie ---> df.info() df.head() df.describe()
- Data selectie ---> df[df.column == 'value'] df.loc[df.column == 'value', :]
- Data wrangling ---> df['column'].fillna() of df.drop_duplicates()
- Data joinen ---> df.merge(df2)
- Data visualisatie ---> px.scatter(df, x, y)
- Eigen functies toepassen ---> df.apply(my_own_function)
- Installeren Jupyter + Python + Pandas ---> pip install pandas

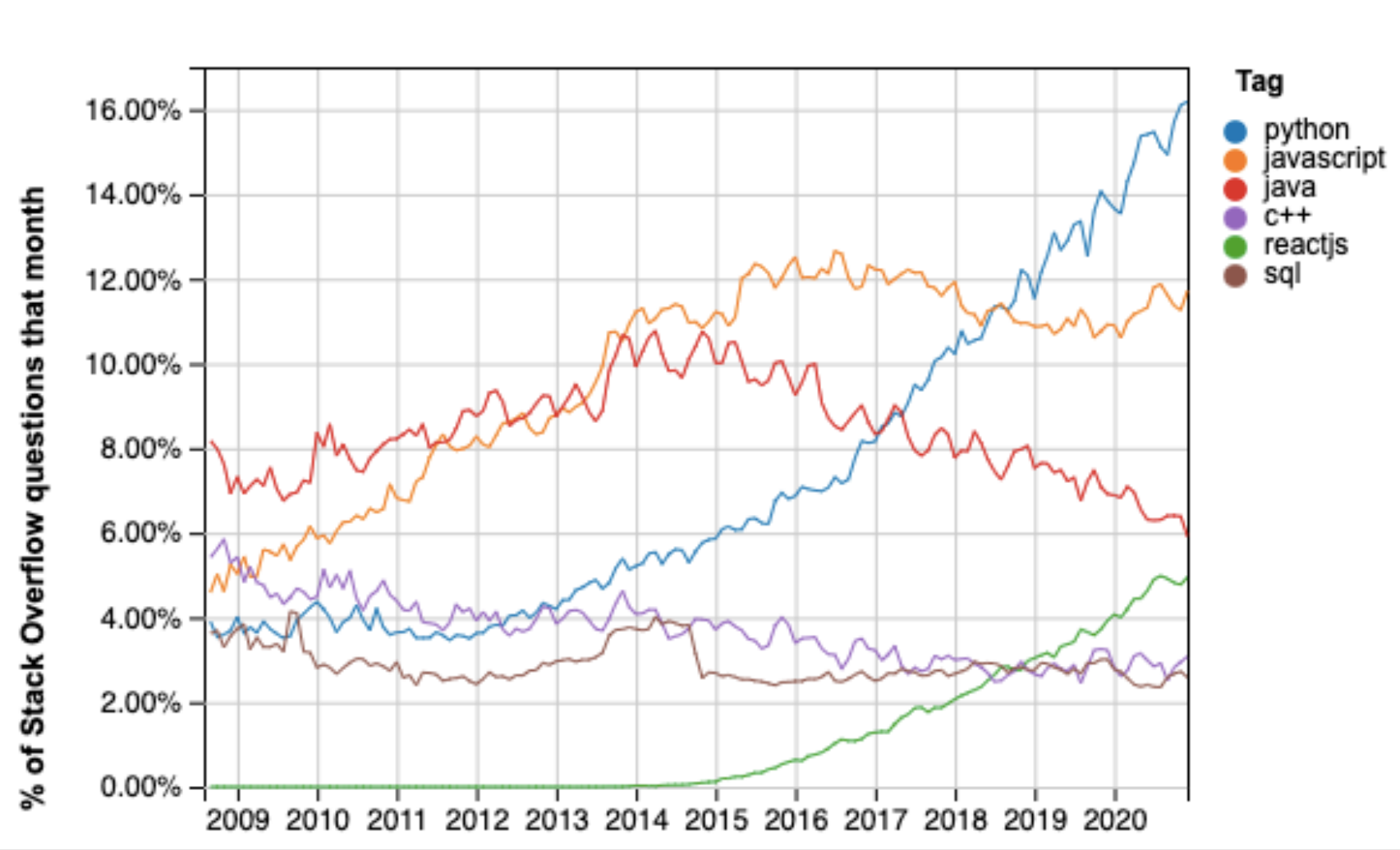
Introductie

- Introductie:
  - waar werk je nu,
  - met welke software,
  - gebruik je nu Python,
  - en: wat wil je vandaag leren

Highover uitleg

- Highover uitleg:
  - Python
  - Pandas
  - en belangrijkste Python data libraries

Python is eating the world



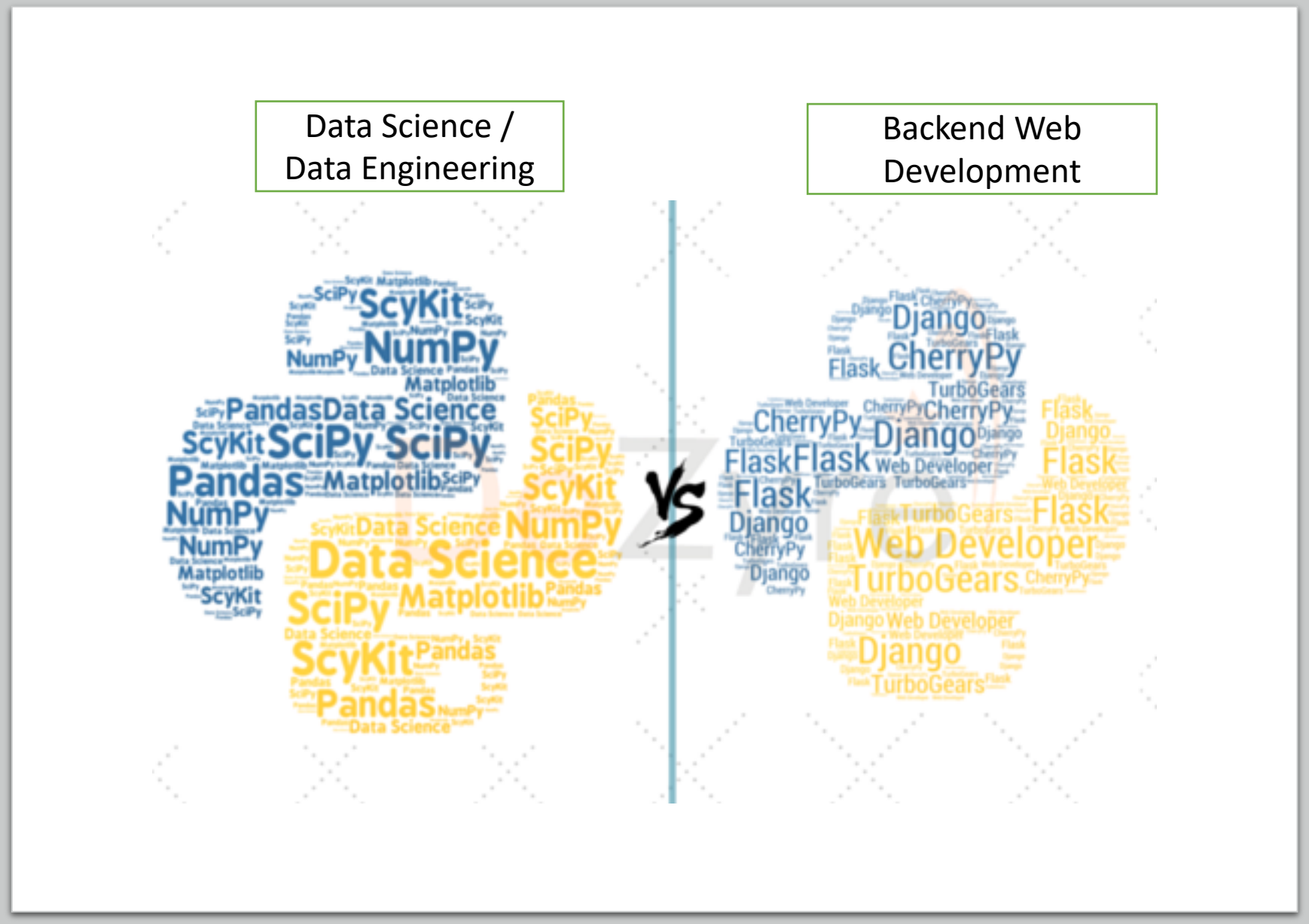
Why is it handy to know some python / pandas?

- veel gebruikt voor data analysis en machine learning
- steeds vaker ingezet voor data engineering: pandas en python zijn zeer flexibel voor data wrangling
- als je de pandas API enigszins kent, dan helpt dat bijv. ook weer met databricks / pyspark
- steeds groter Python data ecosystem, bijv. scheduling met Airflow
- makkelijk om met python API's aan te spreken
- websites scrapen
- zeer leesbare code (relatief)
- en hopelijk: leuk om wat nieuws te leren

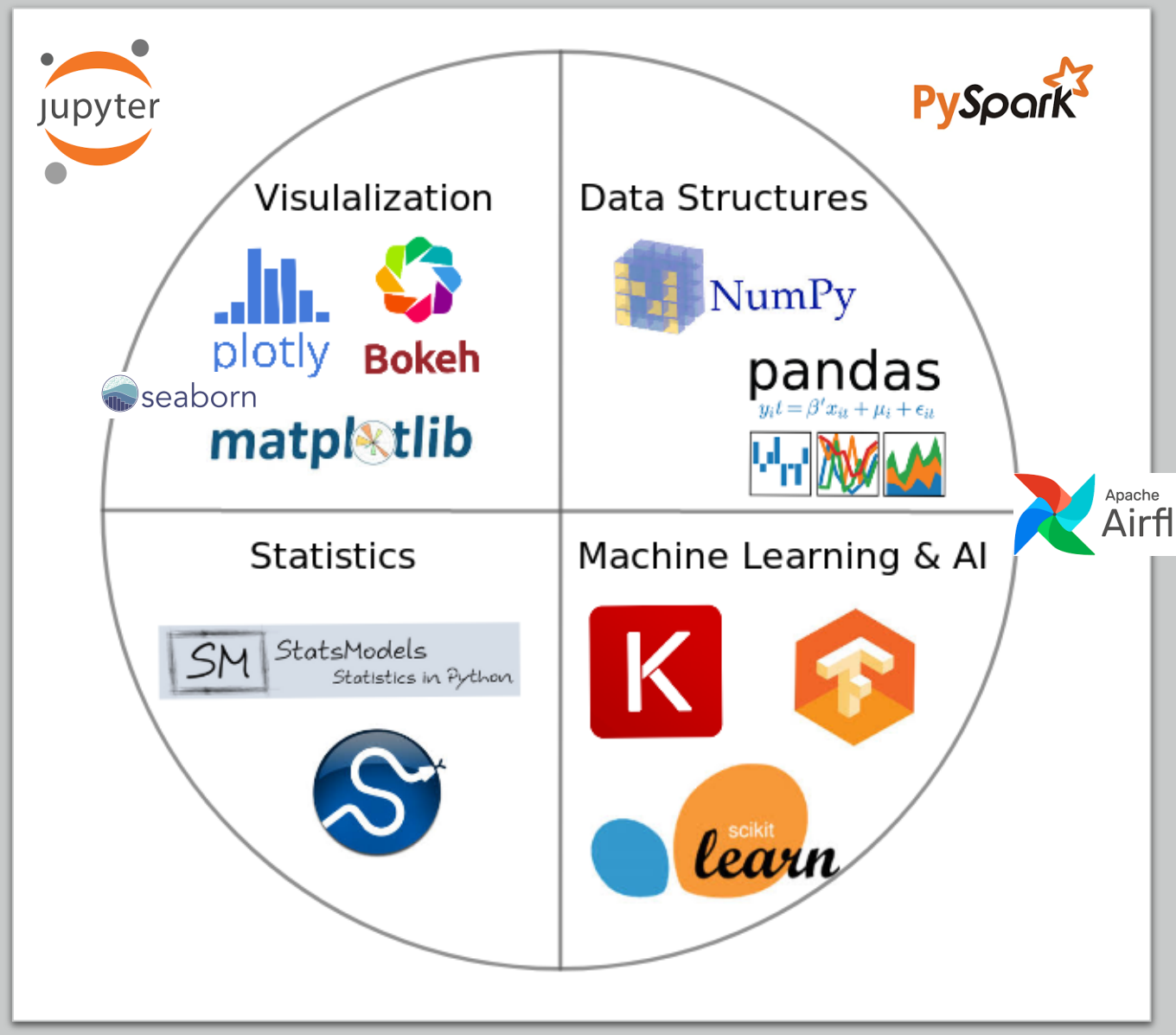
Main disadvantages of python / pandas

- pandas doet alles in-memory, dus voor gedistribueerde oplossingen heb je bijv. Spark of Dask nodig
- higher-level language: gebouwd bovenop C. Omdat het higher-level is, is het ook langzamer
- niet geschikt voor mobile development

Twee wegen die naar Python leiden



Python Data Science Libraries



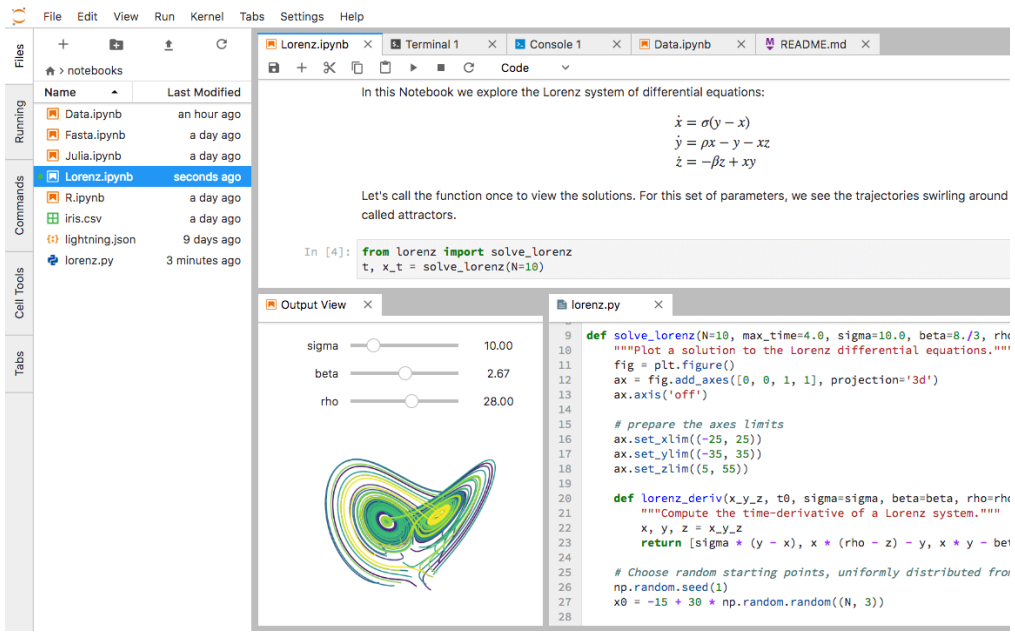
Jupyter Notebooks + Python

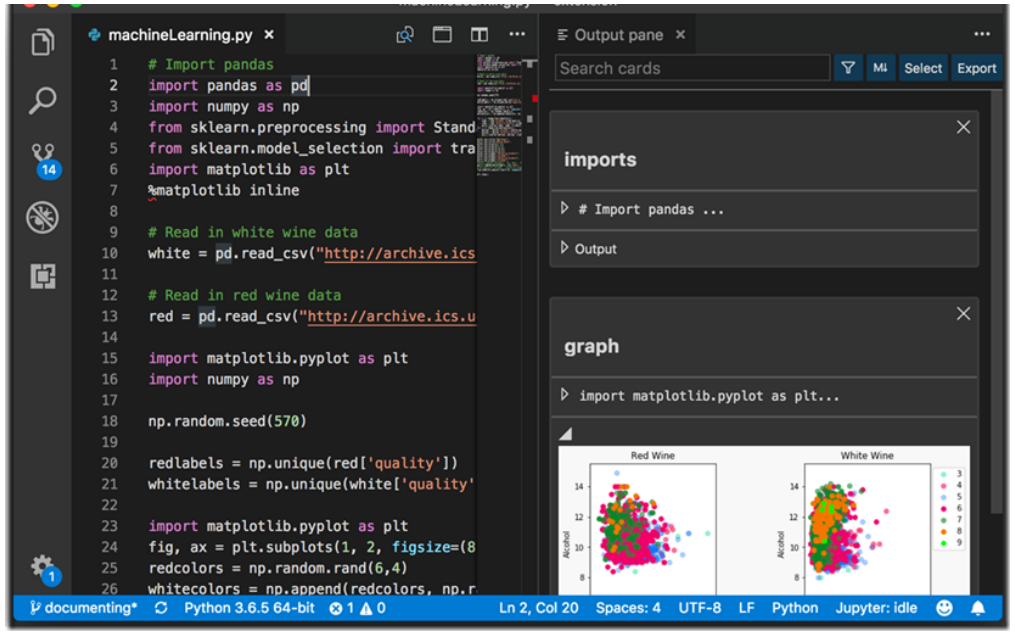
- Intro Jupyter Notebook
- Python refresher
- opstarten VM

Python data types

Name	Type	Description
Integers	int	Whole numbers, such as: **3 300 200**
Floating point	float	Numbers with a decimal point: **2.3 4.6 100.0**
Strings	str	Ordered sequence of characters: **"hello" 'Sammy' "2000" "楽しい"**
Lists	list	Ordered sequence of objects: **[10,"hello",200.3]**
Dictionaries	dict	Unordered Key:Value pairs: **{"mykey" : "value" , "name" : "Frankie"}**
Tuples	tup	Ordered immutable sequence of objects: **(10,"hello",200.3)**
Sets	set	Unordered collection of unique objects: **{"a","b"}**
Booleans	bool	Logical value indicating **True** or **False**

Jupyter Notebooks or Visual Studio Code





Jupyter Notebooks tips and tricks

- Shift + Enter to run code
- Tab completion
- Nieuwe cell: Escape gevolgd door a (above) of b (below) of dd (delete)
- Shift Tab to see arguments and information about methods, functions or classes
- Magic commands, such as ls
- ? or ?? to get extra help and info

Pandas Introductie

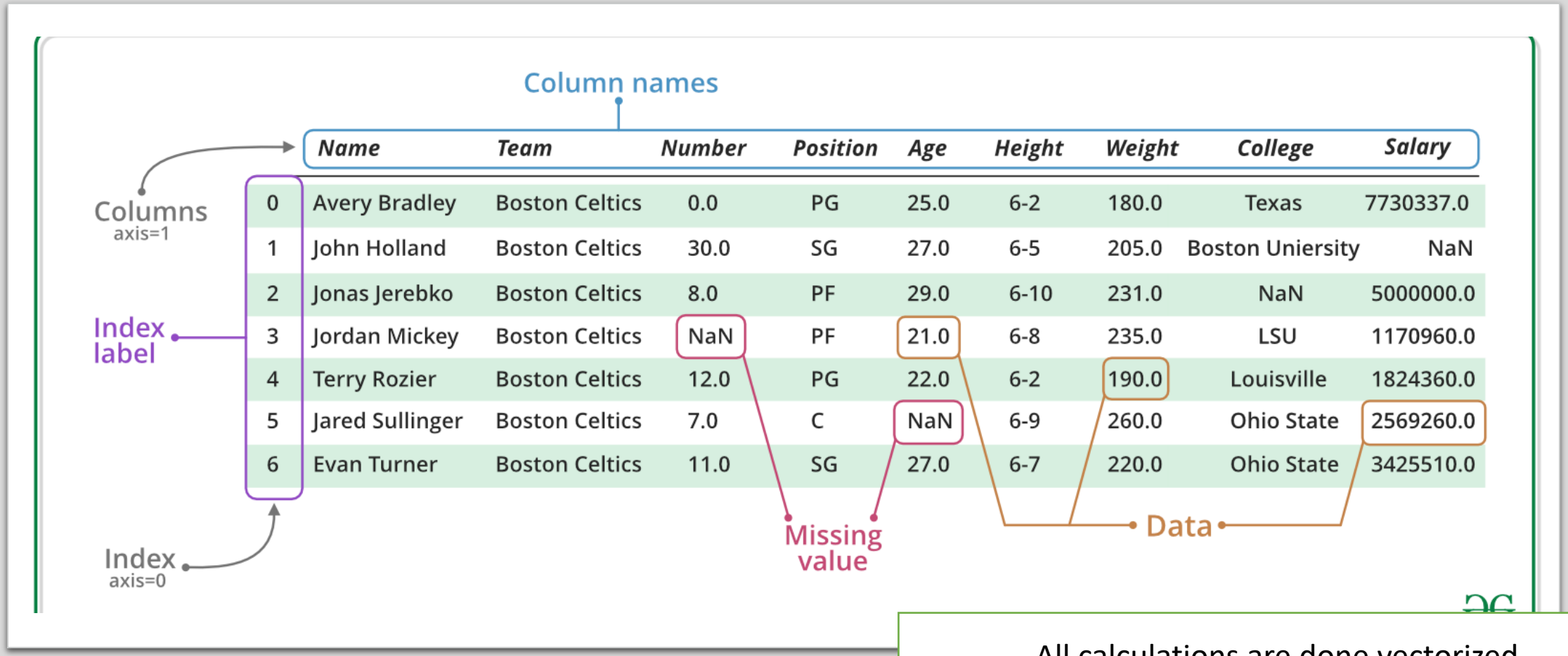
- Data inlezen ---> pd.read_csv()
- Data inspectie ---> df.info() df.head() df.describe()
- Data selectie ---> df[df.column == ‘value’] df.loc[df.column == ‘value’, :]
- Data wrangling ---> df[‘column’].fillna() df.drop_duplicates()
- Data joinen ---> df.merge(df2, how=‘inner’, on=‘column_name’)
- Data visualisatie ---> px.scatter(df, x, y)

Wat is Pandas?

pandas

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

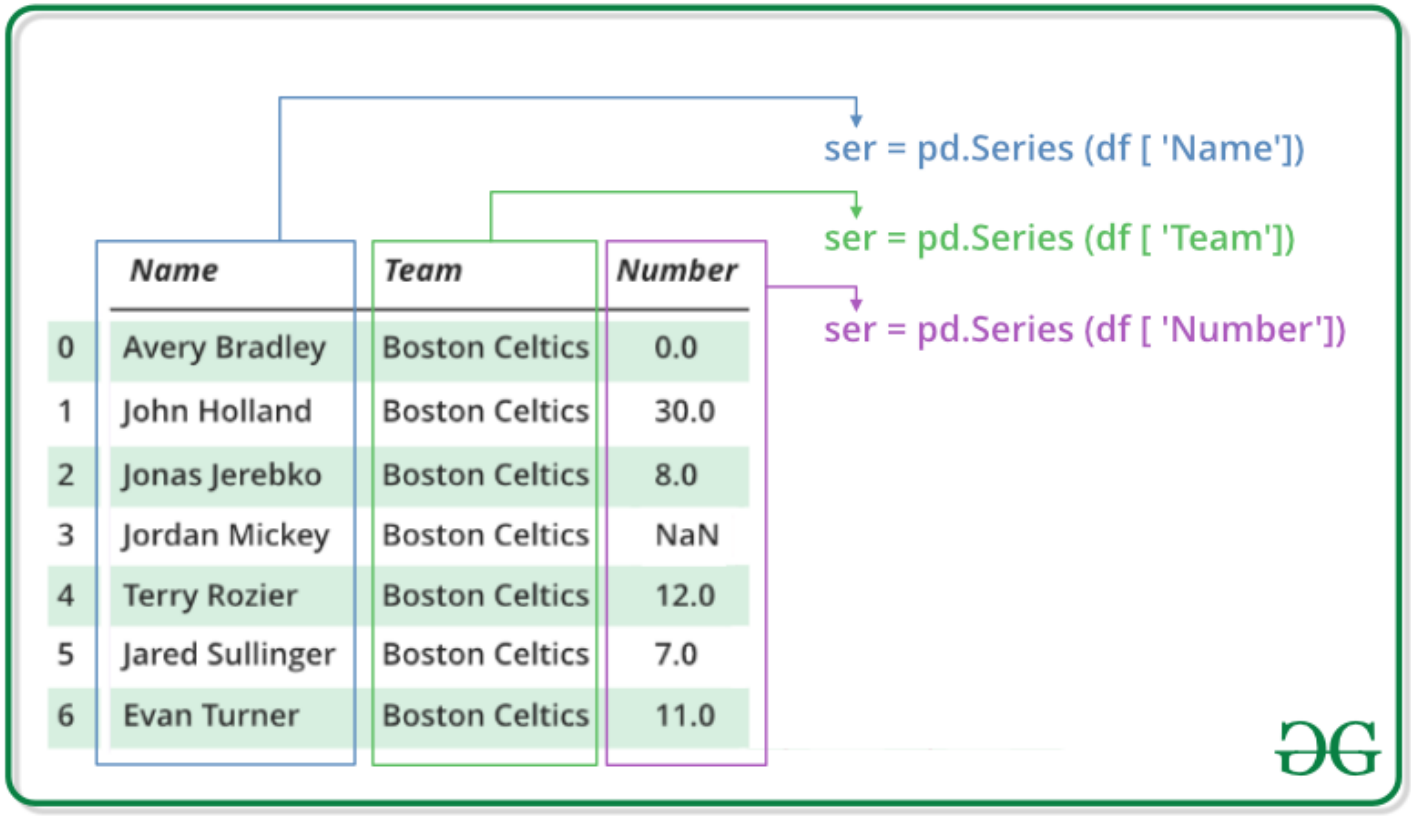
Most important concept is a DataFrame



	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston Uniersity	NaN
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0	6-8	235.0	LSU	1170960.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0	6-2	190.0	Louisville	1824360.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN	6-9	260.0	Ohio State	2569260.0
6	Evan Turner	Boston Celtics	11.0	SG	27.0	6-7	220.0	Ohio State	3425510.0

All calculations are done vectorized.
Don’t use for loops with pandas!!!

Example of a Pandas Series



	Name	Team	Number
0	Avery Bradley	Boston Celtics	0.0
1	John Holland	Boston Celtics	30.0
2	Jonas Jerebko	Boston Celtics	8.0
3	Jordan Mickey	Boston Celtics	NaN
4	Terry Rozier	Boston Celtics	12.0
5	Jared Sullinger	Boston Celtics	7.0
6	Evan Turner	Boston Celtics	11.0

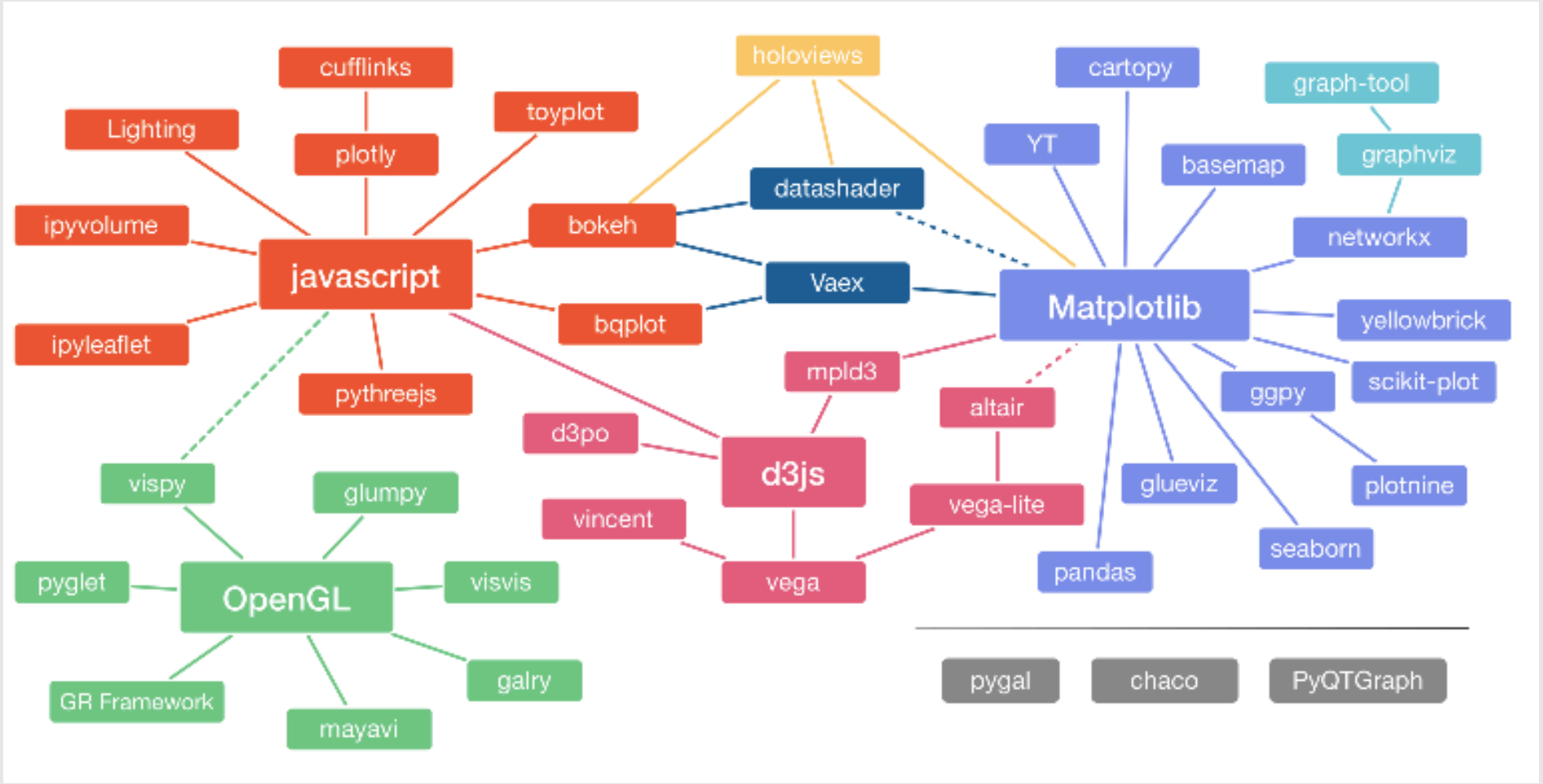
Plotting

- Data visualisatie

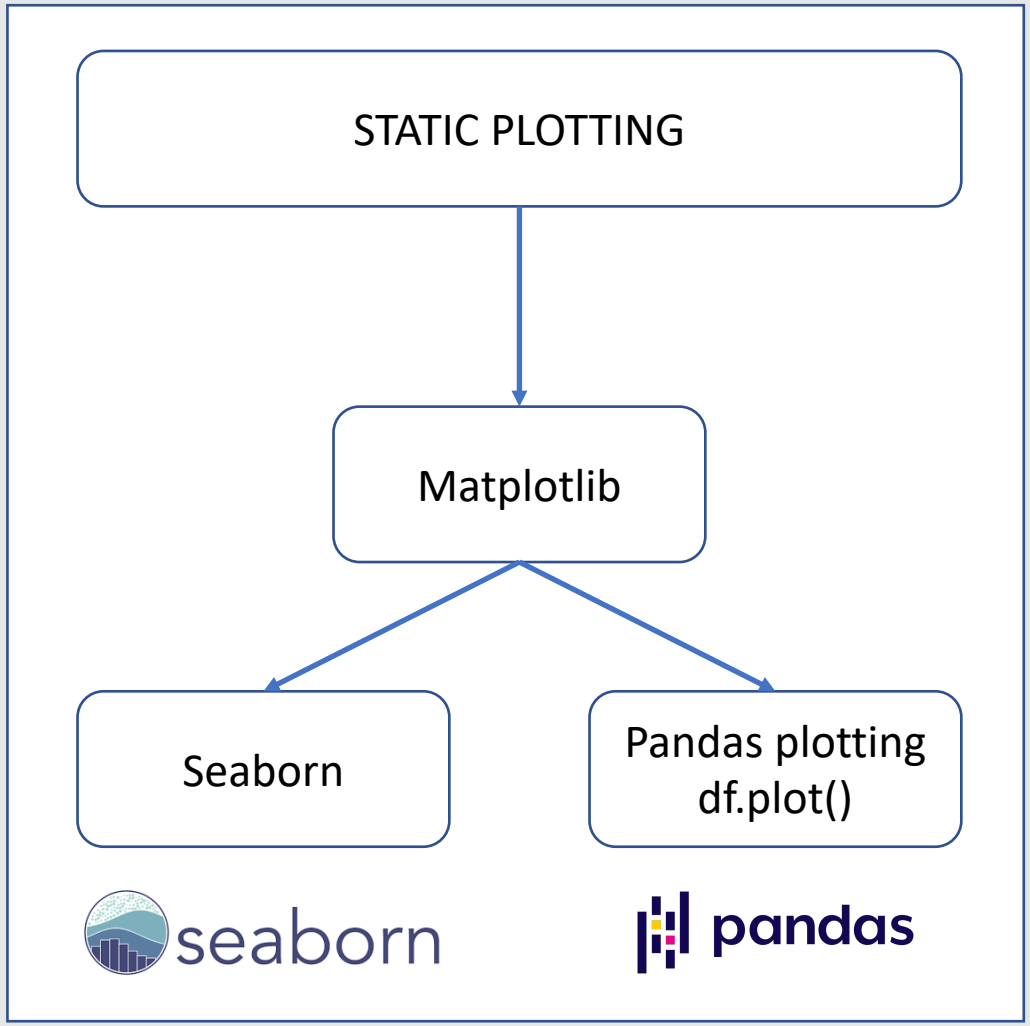
---> px.scatter(df, x, y)

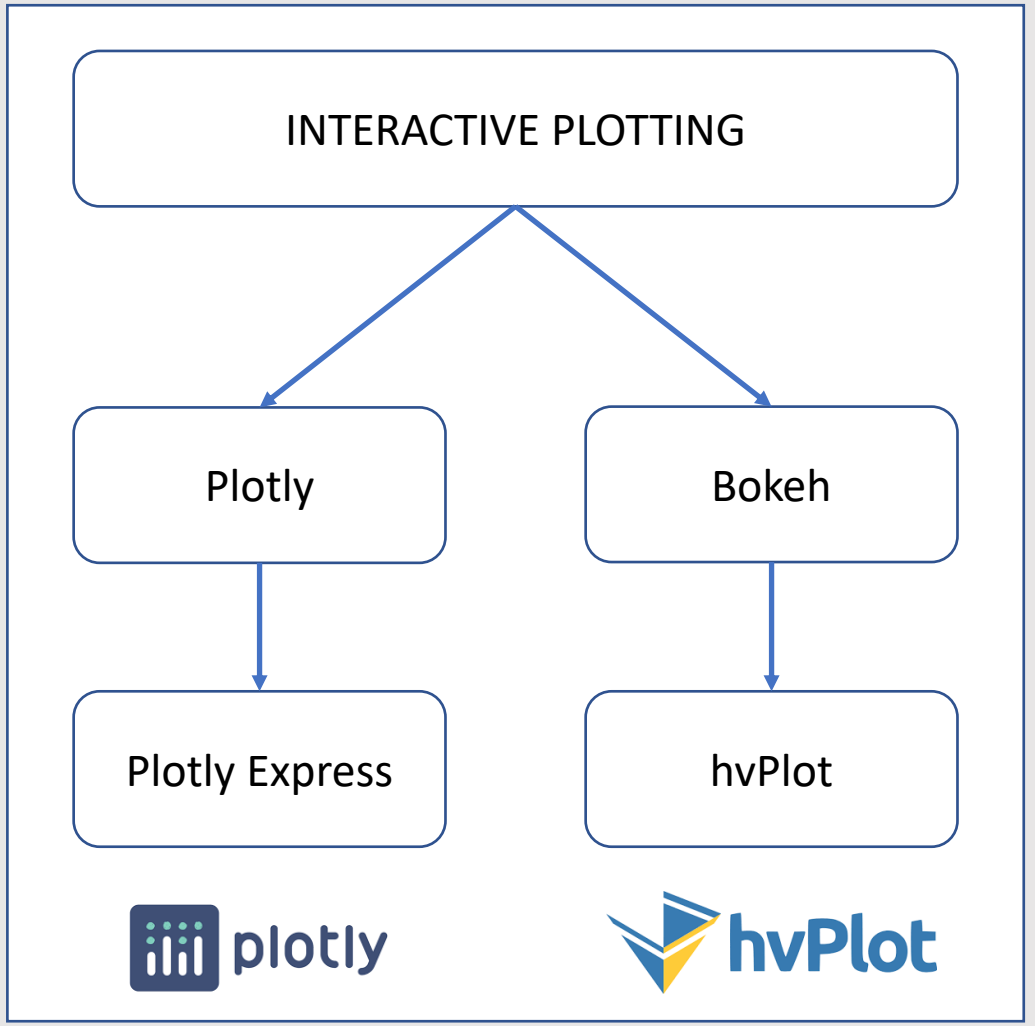
---> sns.scatterplot(df, x, y)

There are too many plotting packages



So my advice is: only focus on these



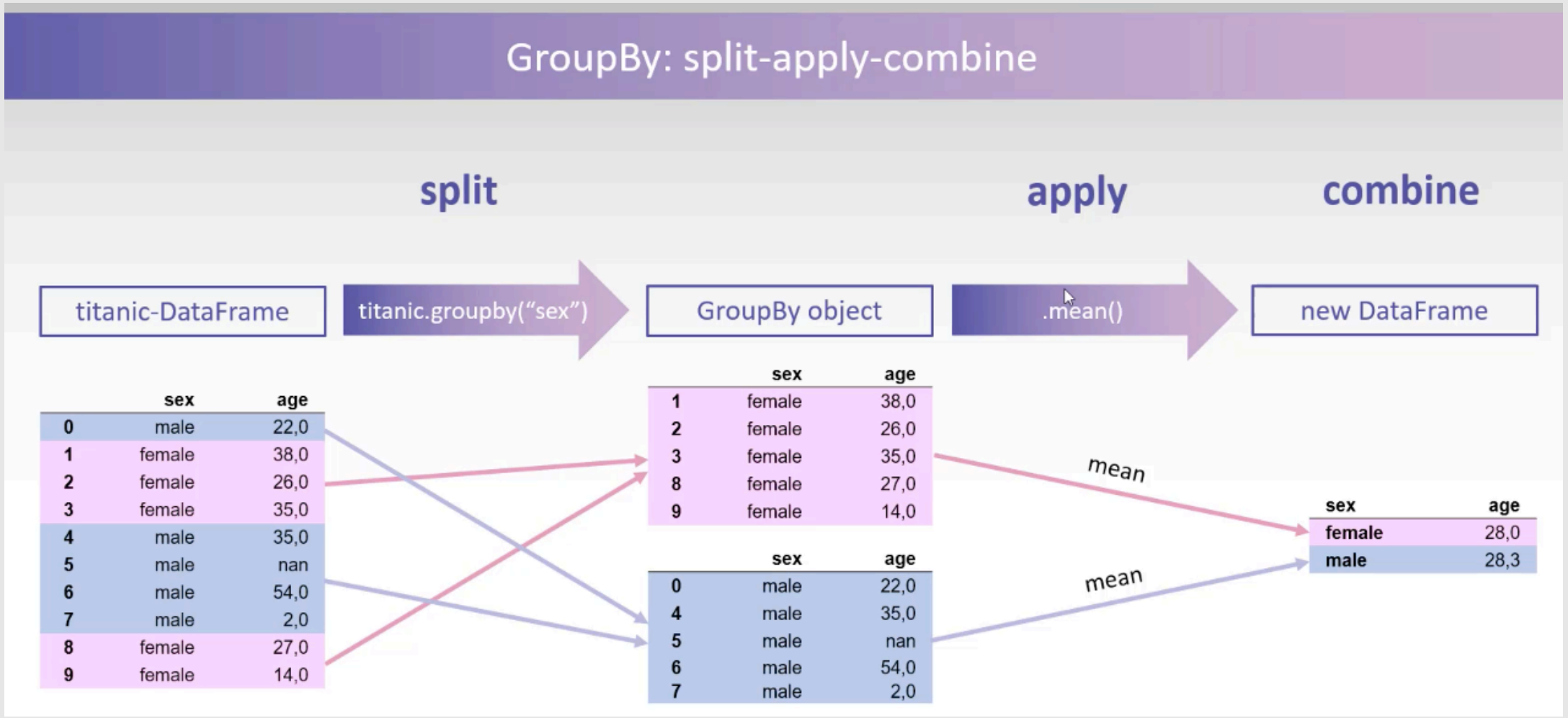


Data aggregation

- Using groupby to aggregate:

---> df.groupby(['startYear'])[['averageRating']].mean()

How groupby() works



	sex	age
0	male	22,0
1	female	38,0
2	female	26,0
3	female	35,0
4	male	35,0
5	male	nan
6	male	54,0
7	male	2,0
8	female	27,0
9	female	14,0

	sex	age
1	female	38,0
2	female	26,0
3	female	35,0
8	female	27,0
9	female	14,0

	sex	age
0	male	22,0
4	male	35,0
5	male	nan
6	male	54,0
7	male	2,0

sex	age
female	28,0
male	28,3

Joining tables

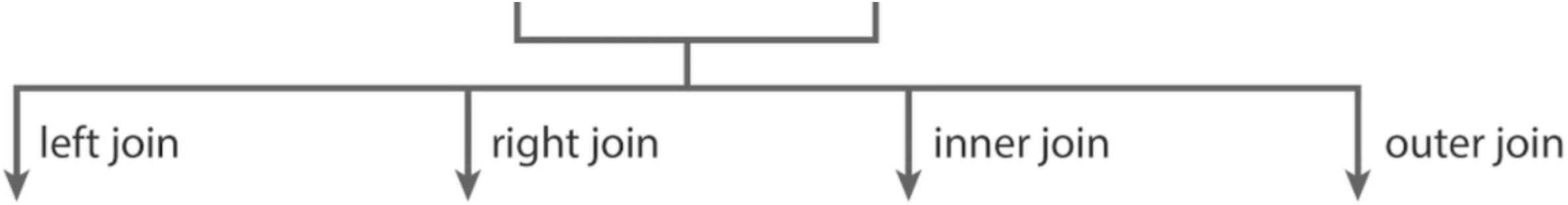
- Using df.merge() to join tables:

---> df.merge(df2, how='inner', on='mutual_column')

Joining tables with df.merge()

	title	runtime
0	Pulp Fiction	90
1	James Bond	120
2	Titanic	115

	title	name
0	Pulp Fiction	John Travolta
1	Pulp Fiction	Samuel L. Jackson
2	James Bond	Sean Connery
3	Terminator	Arnold Schwarzenegger



	title	runtime	name
0	Pulp Fiction	90	John Travolta
1	Pulp Fiction	90	Samuel L. Jackson
2	James Bond	120	Sean Connery
3	Titanic	115	NaN

```
movies.merge(actors, how='left', on='title')
```

	title	runtime	name
0	Pulp Fiction	90.000	John Travolta
1	Pulp Fiction	90.000	Samuel L. Jackson
2	James Bond	120.000	Sean Connery
3	Terminator	NaN	Arnold Schwarzenegger

```
movies.merge(actors, how='right', on='title')
```

	title	runtime	name
0	Pulp Fiction	90	John Travolta
1	Pulp Fiction	90	Samuel L. Jackson
2	James Bond	120	Sean Connery

```
movies.merge(actors, how='inner', on='title')
```

	title	runtime	name
0	Pulp Fiction	90.000	John Travolta
1	Pulp Fiction	90.000	Samuel L. Jackson
2	James Bond	120.000	Sean Connery
3	Titanic	115.000	NaN
4	Terminator	NaN	Arnold Schwarzenegger

```
movies.merge(actors, how='outer', on='title')
```