Data Manipulation

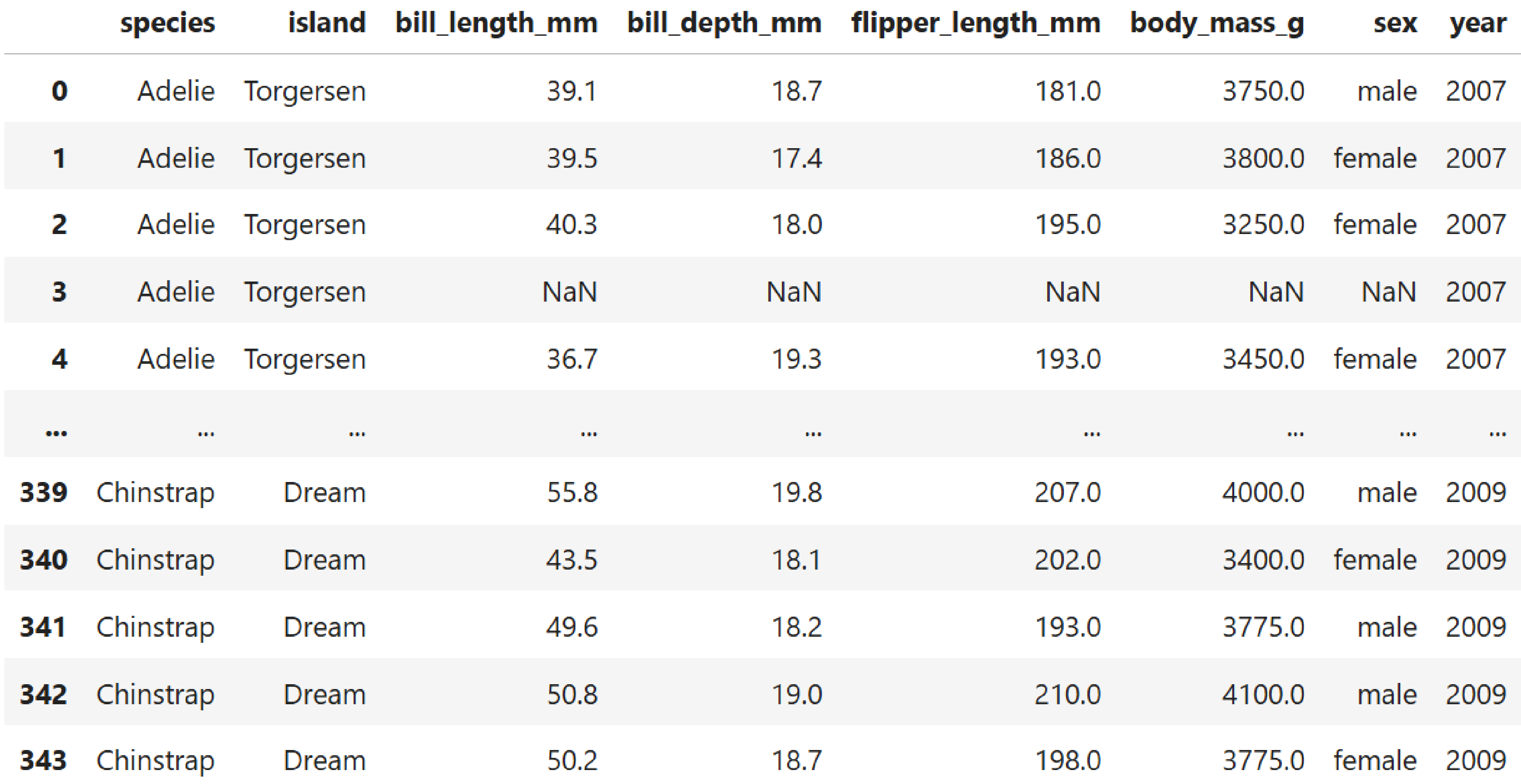
```python
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

import scipy, sklearn, statsmodels
from collections import Counter
```

Dataframes, series, arrays

- Pandas **DataFrame**: table with data, with row (“index”) and column names; same type in each column
- Pandas **Series**: list, with names (“index”), all elements are of the same type
- Numpy **array**: 1- or 2-dimensional array, with no names; all elements are of the same type.

Exercise: which is which?

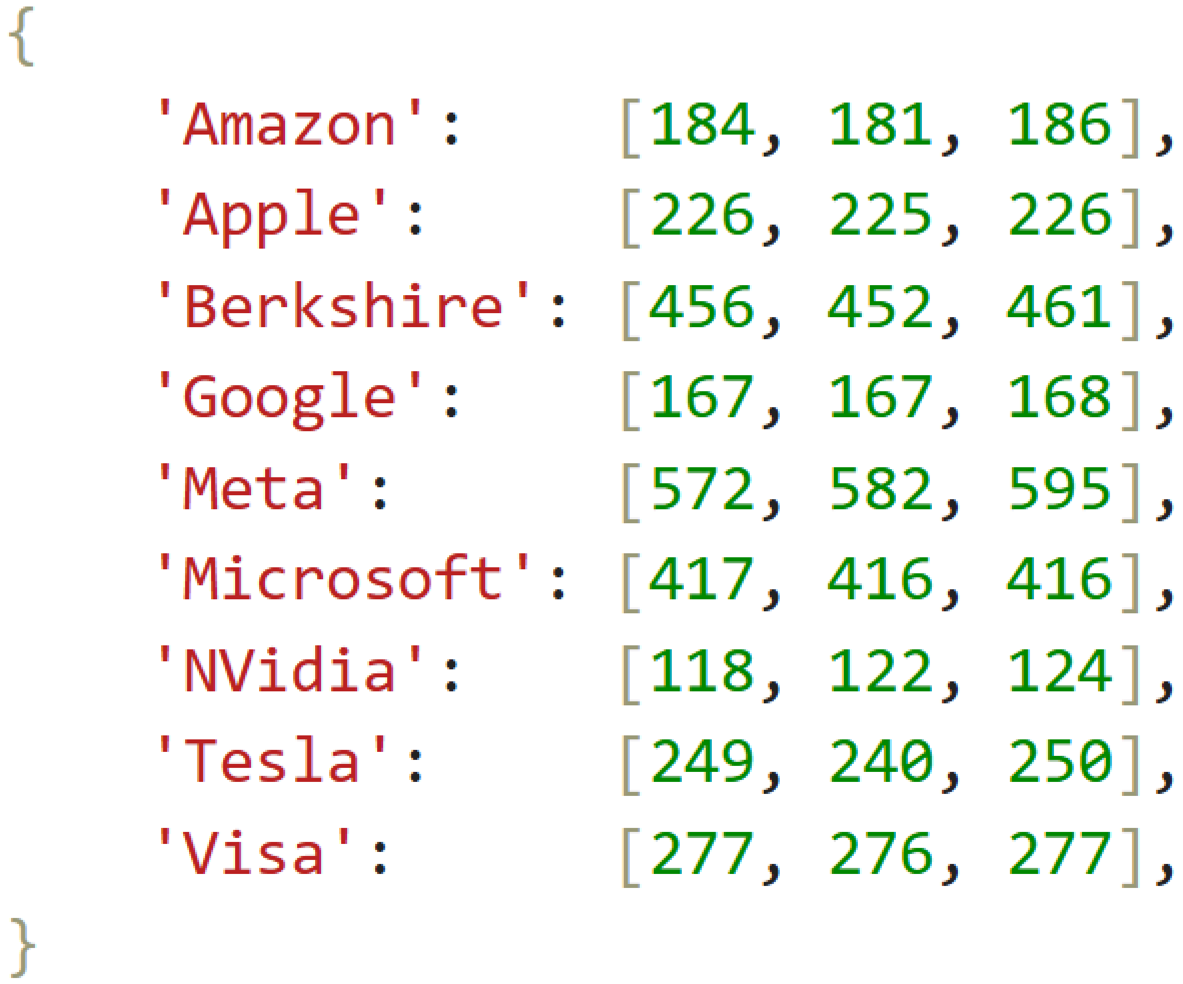
- List
- Dictionary
- Numpy array
- Dataframe
- Series



	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	male	2007
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	female	2007
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	female	2007
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN	2007
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	female	2007
...	...	...	...	...	...	...	...	...
339	Chinstrap	Dream	55.8	19.8	207.0	4000.0	male	2009
340	Chinstrap	Dream	43.5	18.1	202.0	3400.0	female	2009
341	Chinstrap	Dream	49.6	18.2	193.0	3775.0	male	2009
342	Chinstrap	Dream	50.8	19.0	210.0	4100.0	male	2009
343	Chinstrap	Dream	50.2	18.7	198.0	3775.0	female	2009

Exercise: which is which?

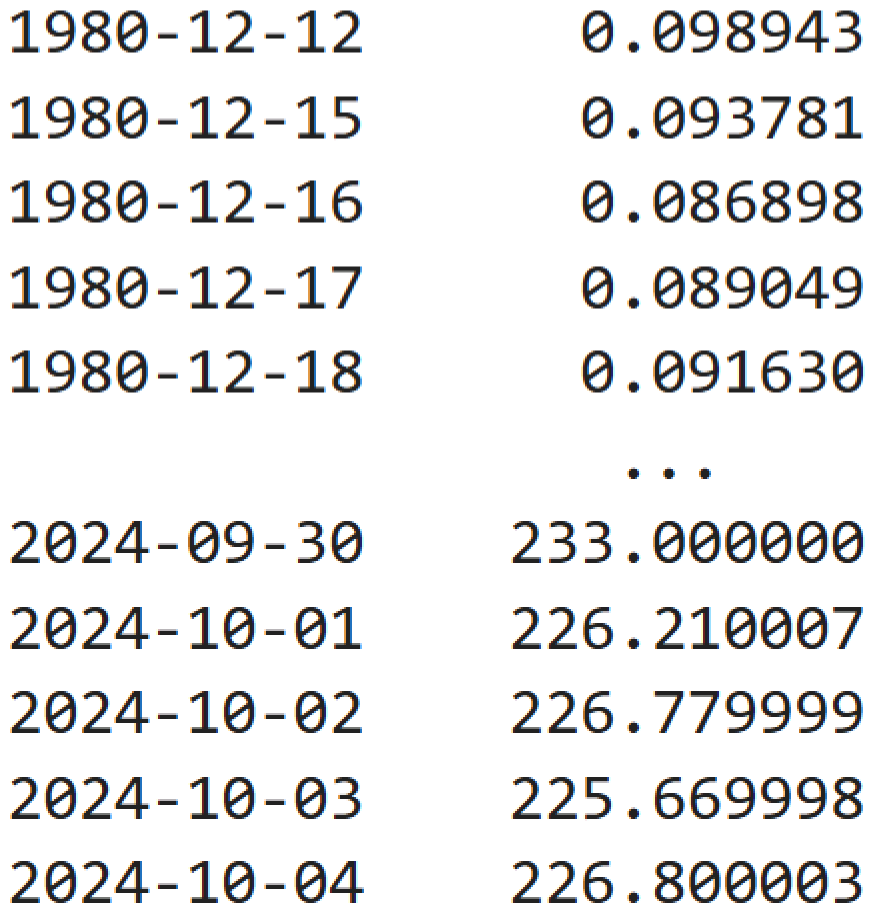
- List
- Dictionary
- Numpy array
- Dataframe
- Series



```
{
    'Amazon':    [184, 181, 186],
    'Apple':     [226, 225, 226],
    'Berkshire': [456, 452, 461],
    'Google':    [167, 167, 168],
    'Meta':      [572, 582, 595],
    'Microsoft': [417, 416, 416],
    'NVidia':    [118, 122, 124],
    'Tesla':     [249, 240, 250],
    'Visa':      [277, 276, 277],
}
```

Exercise: which is which?

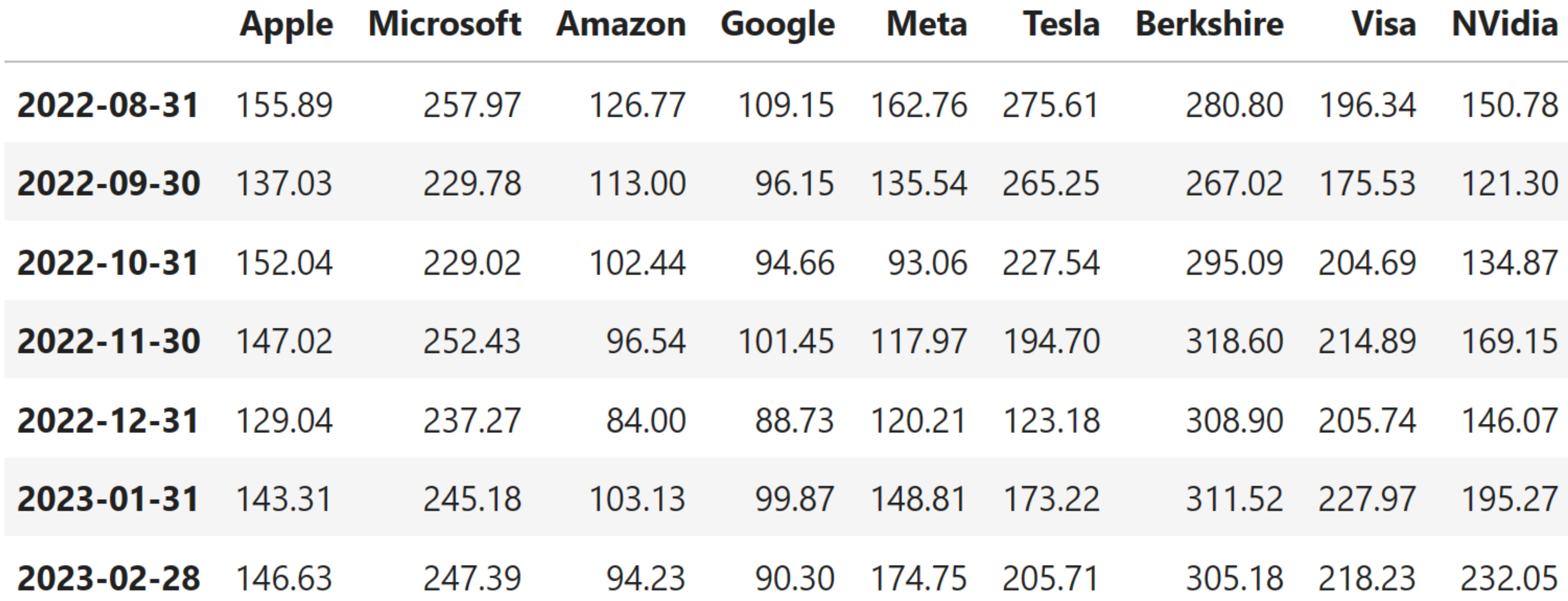
- List
- Dictionary
- Numpy array
- Dataframe
- Series



```
1980-12-12      0.098943
1980-12-15      0.093781
1980-12-16      0.086898
1980-12-17      0.089049
1980-12-18      0.091630
                  ...
2024-09-30    233.000000
2024-10-01    226.210007
2024-10-02    226.779999
2024-10-03    225.669998
2024-10-04    226.800003
```

Exercise: which is which?

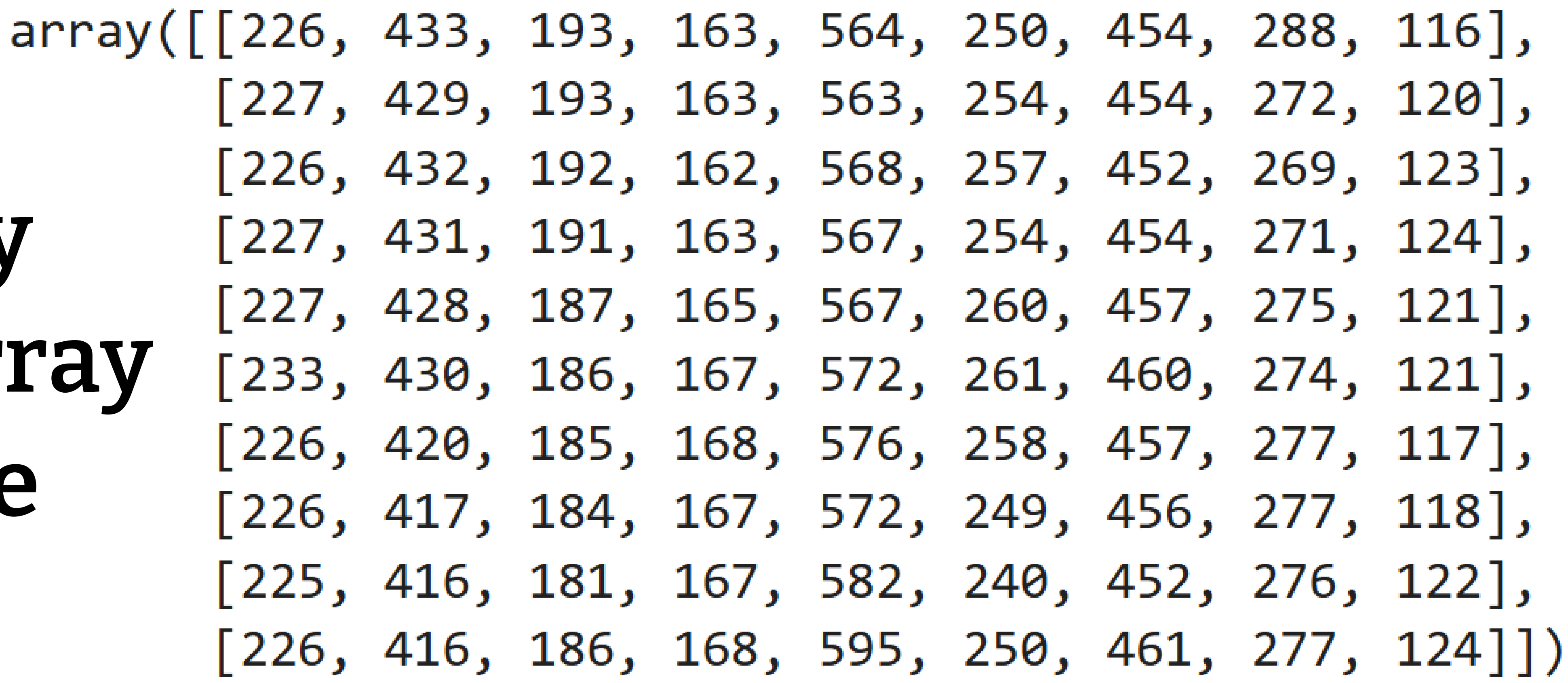
- List
- Dictionary
- Numpy array
- Dataframe
- Series



	Apple	Microsoft	Amazon	Google	Meta	Tesla	Berkshire	Visa	NVidia
2022-08-31	155.89	257.97	126.77	109.15	162.76	275.61	280.80	196.34	150.78
2022-09-30	137.03	229.78	113.00	96.15	135.54	265.25	267.02	175.53	121.30
2022-10-31	152.04	229.02	102.44	94.66	93.06	227.54	295.09	204.69	134.87
2022-11-30	147.02	252.43	96.54	101.45	117.97	194.70	318.60	214.89	169.15
2022-12-31	129.04	237.27	84.00	88.73	120.21	123.18	308.90	205.74	146.07
2023-01-31	143.31	245.18	103.13	99.87	148.81	173.22	311.52	227.97	195.27
2023-02-28	146.63	247.39	94.23	90.30	174.75	205.71	305.18	218.23	232.05

Exercise: which is which?

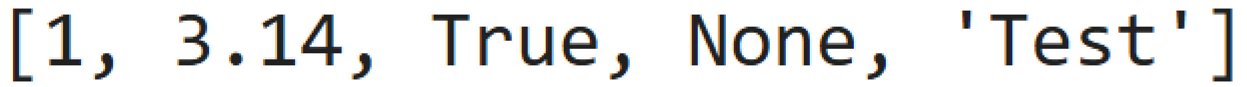
- List
- Dictionary
- Numpy array
- Dataframe
- Series



```
array([[226, 433, 193, 163, 564, 250, 454, 288, 116],
       [227, 429, 193, 163, 563, 254, 454, 272, 120],
       [226, 432, 192, 162, 568, 257, 452, 269, 123],
       [227, 431, 191, 163, 567, 254, 454, 271, 124],
       [227, 428, 187, 165, 567, 260, 457, 275, 121],
       [233, 430, 186, 167, 572, 261, 460, 274, 121],
       [226, 420, 185, 168, 576, 258, 457, 277, 117],
       [226, 417, 184, 167, 572, 249, 456, 277, 118],
       [225, 416, 181, 167, 582, 240, 452, 276, 122],
       [226, 416, 186, 168, 595, 250, 461, 277, 124]])
```

Exercise: which is which?

- List
- Dictionary
- Numpy array
- Dataframe
- Series



```
[1, 3.14, True, None, 'Test']
```

Manipulating dataframes

Check the cheat sheet:

https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf

Exercise: read_csv, describe

- Read the data in the `gapminder.csv` file, with the `pd.read_csv` function.

country	continent	year	lifeExp	pop	gdpPercap
Afghanistan	Asia	1952	28.801	8425333	779.445314
Afghanistan	Asia	1957	30.332	9240934	820.853030
Afghanistan	Asia	1962	31.997	10267083	853.100710
Afghanistan	Asia	1967	34.020	11537966	836.197138
Afghanistan	Asia	1972	36.088	13079460	739.981106

Exercise

- Look at the `shape`, `columns`, `index` attributes
- What does the `describe()` method return?
- Use the `head()` and `tail()` methods to look at the first and last 10 rows
- Use the `min()` and `max()` methods to find the first and last year with data

Indexing

`df[['width’, 'length’, 'species']]`

Select multiple columns with specific names.

`df['width']` *or* `df.width`

Select single column with specific name.

`df[df.Length > 7]`

Extract rows that meet logical criteria.

Use **`df.loc[]`** and **`df.iloc[]`** to select only rows, only columns or both.

`df.iloc[10:20]`

Select rows 10-20.

`df.iloc[:, [1, 2, 5]]`

Select columns in positions 1, 2 and 5 (first column is 0).

`df.loc[:, 'x2':'x4']`

Select all columns between x2 and x4 (inclusive).

`df.loc[df['a'] > 10, ['a’, 'c']]`

Select rows meeting logical condition, and only the specific columns .

https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf

Exercise: indexing

- Extract the numeric columns (using their names)
- Using `iloc`, extract the first row
- Using `iloc`, extract the last 5 values of the last column
- Using `loc`, extract the `pop` column
- What are the minimum and maximum values of the `lifeExp` column?

Decompose complex problems into smaller tasks

Exercise

- Use the `unique()` method to find for which year we have data,
- Use the `len()` function to find how many countries there are.

Boolean Indexing

- For dataframes/series/arrays, the Boolean operators are `&`, `|` and `~` (instead of `and`, `or` and `not`)

Exercise: Boolean indexing

- Extract the rows where `lifeExp` is below 25
- Extract the rows where country is `China`
- Which countries had a life expectancy below 50 years and a GDP per capita above 10,000?

Exercise: sort_values

- Where and when was the lowest life expectancy of all times?
- For the latest year, find the 5 countries with the largest GDP per capita

groupby

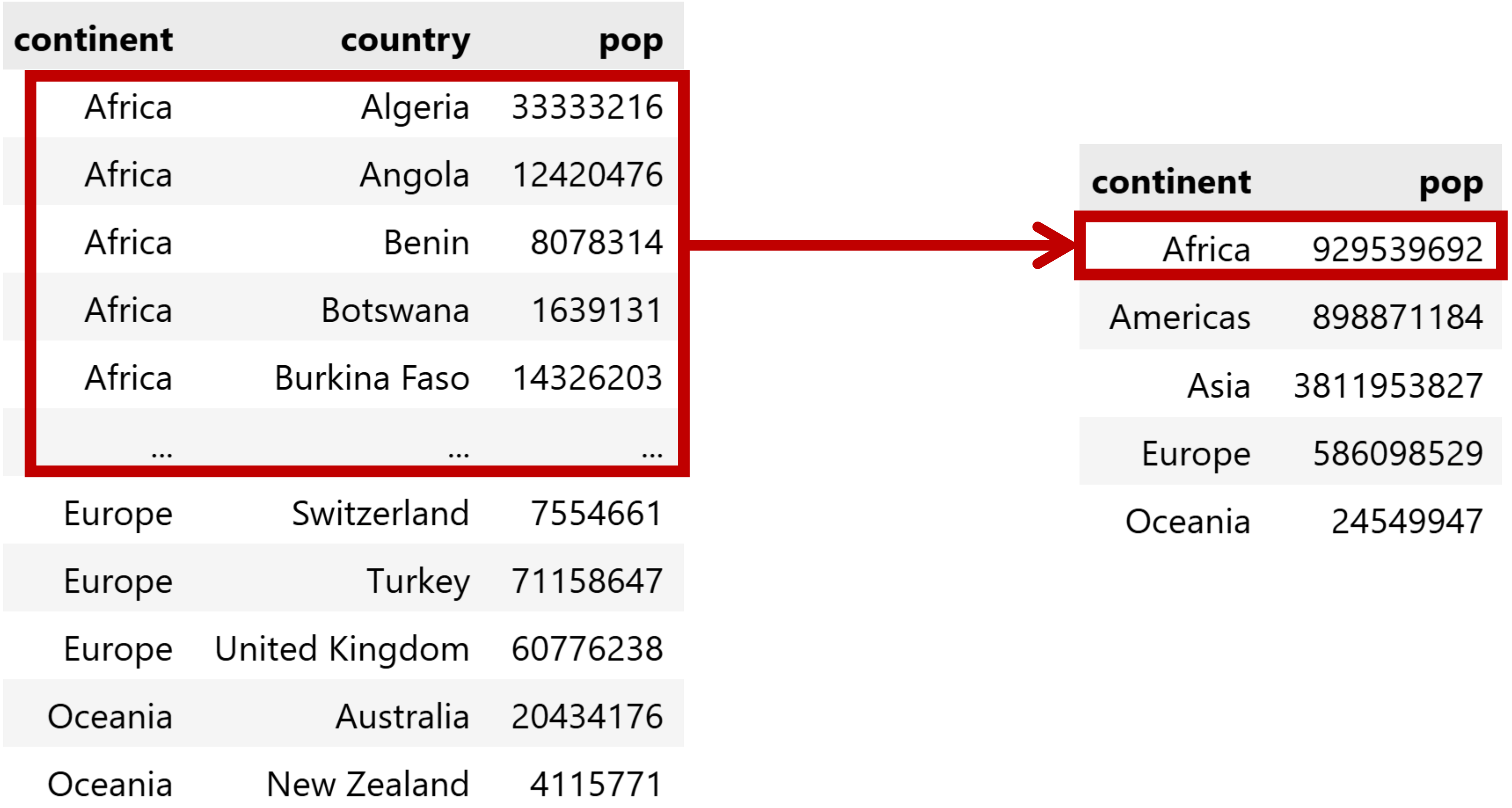
- Compute the average or sum of the values in one column, for each “group”, where the group is defined by another column
- Examples:
  - Total population of each continent
  - Average life expectancy for each year and country

```
d.groupby('continent')['pop'].sum()
```

continent	country	pop
Africa	Algeria	33333216
Africa	Angola	12420476
Africa	Benin	8078314
Africa	Botswana	1639131
Africa	Burkina Faso	14326203
...	...	...
Europe	Switzerland	7554661
Europe	Turkey	71158647
Europe	United Kingdom	60776238
Oceania	Australia	20434176
Oceania	New Zealand	4115771

continent	pop
Africa	929539692
Americas	898871184
Asia	3811953827
Europe	586098529
Oceania	24549947

```
d.groupby('continent')['pop'].sum()
```



continent	country	pop
Africa	Algeria	33333216
Africa	Angola	12420476
Africa	Benin	8078314
Africa	Botswana	1639131
Africa	Burkina Faso	14326203
...	...	...
Europe	Switzerland	7554661
Europe	Turkey	71158647
Europe	United Kingdom	60776238
Oceania	Australia	20434176
Oceania	New Zealand	4115771

continent	pop
Africa	929539692
Americas	898871184
Asia	3811953827
Europe	586098529
Oceania	24549947

```
d.groupby('continent')['pop'].sum()
```

continent	country	pop
Africa	Algeria	33333216
Africa	Angola	12420476
Africa	Benin	8078314
Africa	Botswana	1639131
Africa	Burkina Faso	14326203
...	...	...
Europe	Switzerland	7554661
Europe	Turkey	71158647
Europe	United Kingdom	60776238
Oceania	Australia	20434176
Oceania	New Zealand	4115771

continent	pop
Africa	929539692
Americas	898871184
Asia	3811953827
Europe	586098529
Oceania	24549947

```
d.groupby('continent')['pop'].sum()
```

continent	country	pop
Africa	Algeria	33333216
Africa	Angola	12420476
Africa	Benin	8078314
Africa	Botswana	1639131
Africa	Burkina Faso	14326203
...	...	...
Europe	Switzerland	7554661
Europe	Turkey	71158647
Europe	United Kingdom	60776238
Oceania	Australia	20434176
Oceania	New Zealand	4115771

continent	pop
Africa	929539692
Americas	898871184
Asia	3811953827
Europe	586098529
Oceania	24549947

Exercise: groupby

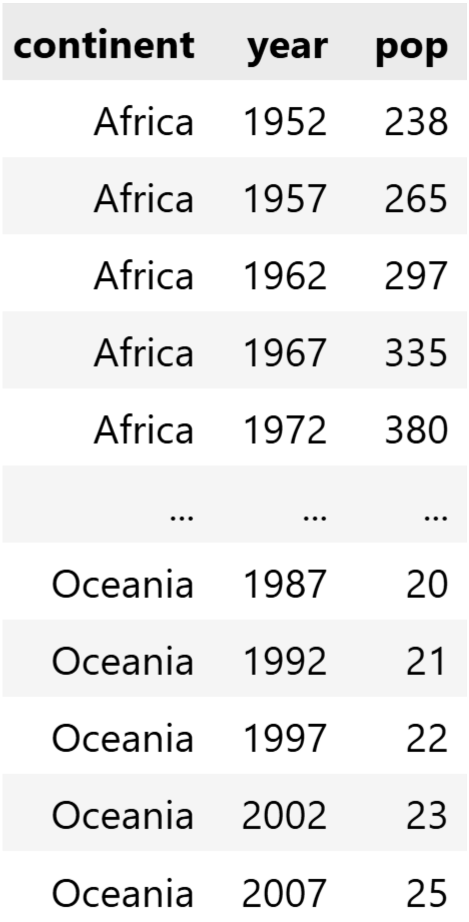
- How many countries are there in each continent?
- Is the number of countries the same each year?
- What is the total population each year?

pivot, melt

- Convert between “long” and “wide” formats

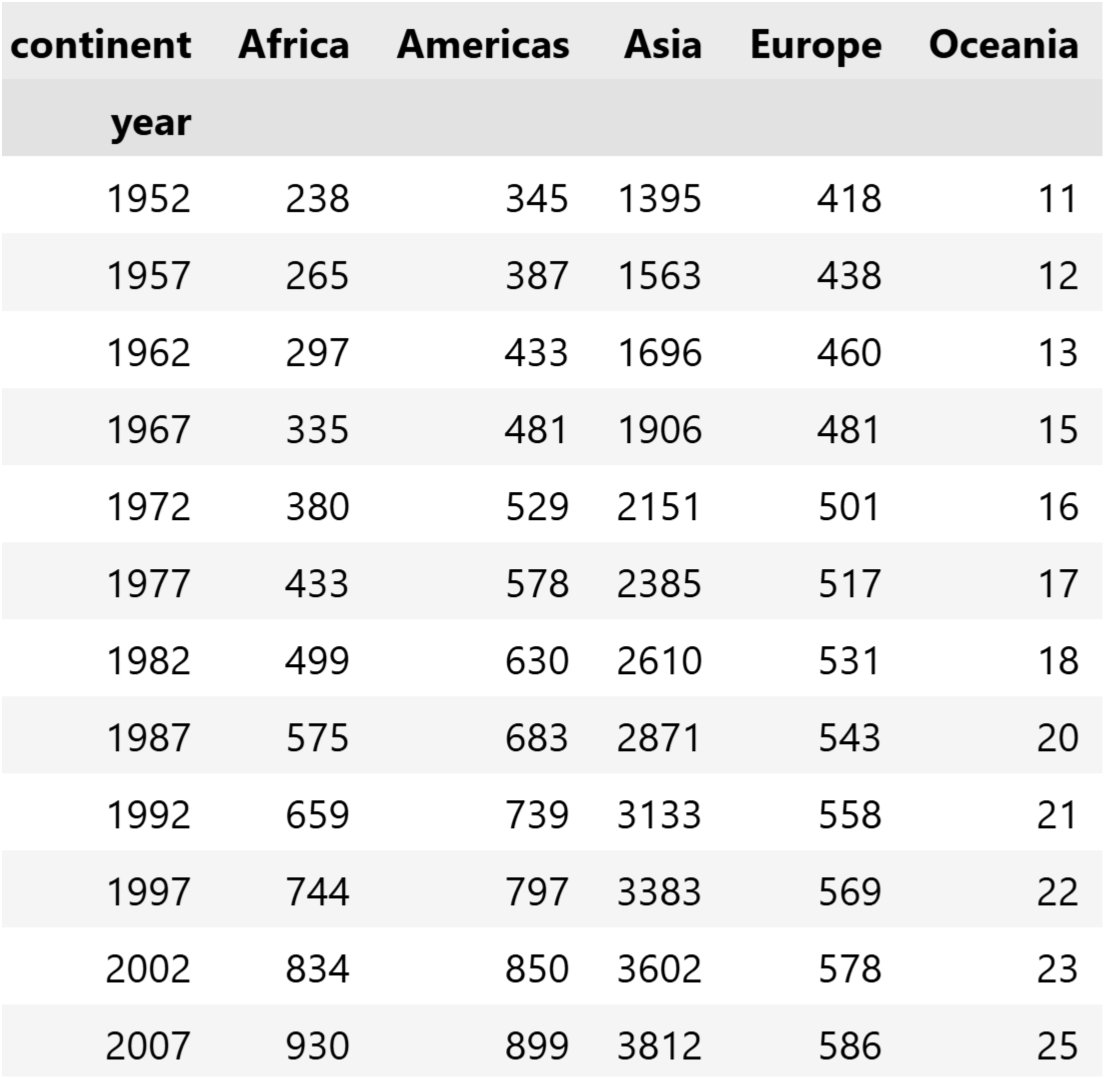
continent	year	pop
Africa	1952	238
Africa	1957	265
Africa	1962	297
Africa	1967	335
Africa	1972	380
...	...	...
Oceania	1987	20
Oceania	1992	21
Oceania	1997	22
Oceania	2002	23
Oceania	2007	25

Long format



continent	year	pop
Africa	1952	238
Africa	1957	265
Africa	1962	297
Africa	1967	335
Africa	1972	380
...	...	...
Oceania	1987	20
Oceania	1992	21
Oceania	1997	22
Oceania	2002	23
Oceania	2007	25

Long format



continent	Africa	Americas	Asia	Europe	Oceania
year					
1952	238	345	1395	418	11
1957	265	387	1563	438	12
1962	297	433	1696	460	13
1967	335	481	1906	481	15
1972	380	529	2151	501	16
1977	433	578	2385	517	17
1982	499	630	2610	531	18
1987	575	683	2871	543	20
1992	659	739	3133	558	21
1997	744	797	3383	569	22
2002	834	850	3602	578	23
2007	930	899	3812	586	25

Wide format

continent	year	pop
Africa	1952	238
Africa	1957	265
Africa	1962	297
Africa	1967	335
Africa	1972	380
...	...	...
Oceania	1987	20
Oceania	1992	21
Oceania	1997	22
Oceania	2002	23
Oceania	2007	25

`d.pivot()` →

← `d.melt()`

continent	Africa	Americas	Asia	Europe	Oceania
year					
1952	238	345	1395	418	11
1957	265	387	1563	438	12
1962	297	433	1696	460	13
1967	335	481	1906	481	15
1972	380	529	2151	501	16
1977	433	578	2385	517	17
1982	499	630	2610	531	18
1987	575	683	2871	543	20
1992	659	739	3133	558	21
1997	744	797	3383	569	22
2002	834	850	3602	578	23
2007	930	899	3812	586	25

Long format

Wide format

continent	year	pop
Africa	1952	238
Africa	1957	265
Africa	1962	297
Africa	1967	335
Africa	1972	380
...	...	...
Oceania	1987	20
Oceania	1992	21
Oceania	1997	22
Oceania	2002	23
Oceania	2007	25

Long format

`d.pivot()` →

← `d.melt()`

continent	Africa	Americas	Asia	Europe	Oceania
year					
1952	238	345	1395	418	11
1957	265	387	1563	438	12
1962	297	433	1696	460	13
1967	335	481	1906	481	15
1972	380	529	2151	501	16
1977	433	578	2385	517	17
1982	499	630	2610	531	18
1987	575	683	2871	543	20
1992	659	739	3133	558	21
1997	744	797	3383	569	22
2002	834	850	3602	578	23
2007	930	899	3812	586	25

Wide format

```
d.pivot(index='year', columns='continent', values='pop')
```

continent	year	pop
Africa	1952	238
Africa	1957	265
Africa	1962	297
Africa	1967	335
Africa	1972	380
...	...	...
Oceania	1987	20
Oceania	1992	21
Oceania	1997	22
Oceania	2002	23
Oceania	2007	25

Long format

continent	Africa	Americas	Asia	Europe	Oceania
year					
1952	238	345	1395	418	11
1957	265	387	1563	438	12
1962	297	433	1696	460	13
1967	335	481	1906	481	15
1972	380	529	2151	501	16
1977	433	578	2385	517	17
1982	499	630	2610	531	18
1987	575	683	2871	543	20
1992	659	739	3133	558	21
1997	744	797	3383	569	22
2002	834	850	3602	578	23
2007	930	899	3812	586	25

Wide format

```
d.melt( ignore_index = False ).reset_index()
```

continent	year	pop
Africa	1952	238
Africa	1957	265
Africa	1962	297
Africa	1967	335
Africa	1972	380
...	...	...
Oceania	1987	20
Oceania	1992	21
Oceania	1997	22
Oceania	2002	23
Oceania	2007	25

Long format

continent	Africa	Americas	Asia	Europe	Oceania
year					
1952	238	345	1395	418	11
1957	265	387	1563	438	12
1962	297	433	1696	460	13
1967	335	481	1906	481	15
1972	380	529	2151	501	16
1977	433	578	2385	517	17
1982	499	630	2610	531	18
1987	575	683	2871	543	20
1992	659	739	3133	558	21
1997	744	797	3383	569	22
2002	834	850	3602	578	23
2007	930	899	3812	586	25

Wide format

Exercise: pivot, melt

- Use the `pivot()` method to build a data-frame with one row per country, one column per date, and containing the life expectancy
- What happens if you call the `melt()` method on that data-frame?

Exercise: groupby, pivot

- What is the average life expectancy for each continent and each year; use `pivot` to present the result one row per year, one column per continent.

Exercise

- Are there countries whose life expectancy decreased over time?
- With the `diff()` method, compute the change in world population over time
- Compute the change in world GDP over time

Objectives

- Load data from a CSV file into Python
- Extracts rows satisfying some condition
- Aggregate (sum, mean, etc.) some of the columns, with groupby
- Reshape the data, with pivot and melt
- Answer queries requiring multi-step data manipulations

Exercise

- Which other questions could you ask and answer about this dataset (feel free to ask ChatGPT to come up with such questions)

Exercises (ChatGPT)

1. What is the average life expectancy for each continent across all years?

2. Which country had the highest GDP per capita in 2007?

3. What is the total population of each continent in the most recent year available?

4. How has the life expectancy in a specific country changed over time?

5. Which continent has seen the greatest improvement in life expectancy from 1950 to 2007?

Exercises (ChatGPT)

1. What is the average life expectancy for each continent across all years?
→ *Group the data by continent and calculate the mean life expectancy.*

2. Which country had the highest GDP per capita in 2007?
→ *Filter the data for the year 2007 and find the country with the maximum GDP per capita.*

3. What is the total population of each continent in the most recent year available?
→ *Filter the data for the most recent year and group by continent to sum the population.*

4. How has the life expectancy in a specific country changed over time?
→ *Filter the data by country and plot the life expectancy over the years.*

5. Which continent has seen the greatest improvement in life expectancy from 1950 to 2007?
→ *Group the data by continent and compare the difference in average life expectancy between 1950 and 2007.*

Exercises (ChatGPT)

6. What are the top 5 countries with the largest population growth between two years?

7. How is GDP per capita related to life expectancy for different countries?

8. What was the median life expectancy for each continent in 1980?

9. Which countries had a life expectancy above 75 years in 2007?

10. How has the population of each continent changed over time?

Exercises (ChatGPT)

6. What are the top 5 countries with the largest population growth between two years?
→ Calculate the population difference between two years for each country and sort the results.

7. How is GDP per capita related to life expectancy for different countries?
→ Create a scatter plot to visualize the relationship between GDP per capita and life expectancy.

8. What was the median life expectancy for each continent in 1980?
→Filter the data for the year 1980, group by continent, and calculate the median life expectancy.

9. Which countries had a life expectancy above 75 years in 2007?
→Filter the data for the year 2007 and select the countries with a life expectancy above 75 years.

10. How has the population of each continent changed over time?
→ Group the data by continent and year, then plot the total population for each continent over time.

Extra Exercise

- Extract population, GDP per capita, and life expectancy (and more, if you want) from:

  https://github.com/open-numbers/ddf--open_numbers--world_development_indicators

- Is it the same data as before? What are the differences?

Extra Slides

Prompt

We are teaching a 1-week summer school on data science with Python.

The "data manipulation with pandas" session uses the Gapminder dataset: a dataframe with columns "country", "continent", "year", "life expectancy", "population", "GDP per capita".

Give me 10 questions about this dataset, that could be answered with data manipulations in Pandas.