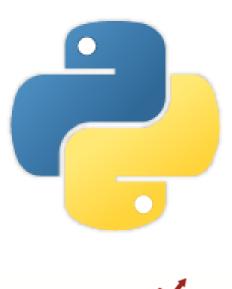
# Introduction to Python

Katowice, IntQuant 2024





# Agenda

- Introduction: why Python?
- Python: 15 mins course
- Data: pandas
- Regression: scikit-learn
- Visualization: matplotlib

1

Introduction: why Python?

# Introduction: why Python?

- Why Python? https://www.python.org/
- Materials
  - Introductory: https://www.py4e.com/, https://www.tutorialspoint.com/python/index.htm
  - Advanced: Fluent Python by Luciano Ramalho
- Programming environment
  - Anaconda https://www.anaconda.com/
  - conda

Python: 30 minutes course

### Python: 15 minutes course

### **Data Types and Operators**

- Basic types: str, int, float, bool
- Containers: list, dict, set
- Arithmetic operators: +, -, \*, /, \*\*
- Logical operators: ==, !=, >, <, <=, >=

#### **Control Flow**

- if condition
- for loop

```
x = 1
   V = 2.0
   print(x+y, type(x), type(x+y))

√ 0.5s

3.0 <class 'int'> <class 'float'>
   z = str(x) + str(y)
   print(z,type(z))

√ 0.5s

12.0 <class 'str'>
   b = (x==y)
   print(b,type(b))

√ 0.4s

False <class 'bool'>
   first list = [x,y,z]
   if not b:
        for i in first list:
            print(i)

√ 0.7s

2.0
12.0
```

### Python: 15 minutes course

#### **Functions**

- built-in functions
- methods
- def key word

#### **Modules**

- import statement
- Python standard library
- Externals modules: numpy

```
len(first_list)

√ 0.4s

   first_list.pop(-1)

√ 0.2s

2.0
   def sum of elements(x):
        s = 0
        for i in x:
            s +=i
        return s
   sum of elements(first list)
 ✓ 0.6s
3.0
   import os
   os.mkdir("test")

√ 0.5s

    import numpy as np
   np.std(first_list)

√ 2.7s
```

Data: pandas

### Data: pandas

### https://pandas.pydata.org/

```
import pandas as pd
  df_iris = pd.read_csv("iris.csv", index_col=0)
  df iris

√ 0.8s

      Sepal.Length Sepal.Width Petal.Length Petal.Width
                                                              Species
               5.1
                             3.5
                                                        0.2
                                           1.4
                                                               setosa
   2
               4.9
                             3.0
                                           1.4
                                                        0.2
                                                               setosa
                             3.2
                                                        0.2
   3
               4.7
                                           1.3
                                                               setosa
   4
               4.6
                             3.1
                                                        0.2
                                           1.5
                                                               setosa
   5
               5.0
                             3.6
                                           1.4
                                                        0.2
                                                               setosa
                                                        2.3 virginica
146
               6.7
                             3.0
                                           5.2
147
               6.3
                             2.5
                                           5.0
                                                        1.9 virginica
148
               6.5
                             3.0
                                           5.2
                                                        2.0 virginica
149
               6.2
                             3.4
                                           5.4
                                                        2.3 virginica
               5.9
                             3.0
                                           5.1
                                                        1.8 virginica
150
150 rows × 5 columns
```

Regression: scikit-learn

## Regression: scikit-learn

### https://scikit-learn.org/stable/

```
from sklearn.linear model import LinearRegression

√ 4.5s

   X = df iris[['Sepal.Length', 'Sepal.Width', 'Petal.Length']]
   y = df_iris['Petal.Width']
   reg = LinearRegression()
   reg.fit(X,y)

√ 0.6s

LinearRegression()
   print("predict: " +str(reg.predict([X.loc[1,:]])[0]), "actual: " + str(y[1]))

√ 0.5s

predict: 0.21625189892792285 actual: 0.2
   y preds = reg.predict(X)
   1 - np.sum((y_preds-y)**2)/np.sum((y- np.mean(y))**2)

√ 0.1s

0.9378502736046809
```

Visualization: matplotlib

# Visualization: matplotlib

### https://matplotlib.org/

```
import matplotlib.pyplot as plt
   species list = df iris.Species.unique()
   sepal_lengths = [np.mean(df_iris[df_iris.Species == species]['Sepal.Length']) for species in species_list]
   sepal width = [np.mean(df_iris[df_iris.Species == species]['Sepal.Width']) for species in species_list]
   petal_lengths = [np.mean(df_iris[df_iris.Species == species]['Petal_Length']) for species in species_list]
   petal width = [np.mean(df iris[df iris.Species == species]['Petal.Width']) for species in species list]
   plt.figure()
   plt.subplot(121)
   plt.bar(species list, sepal width)
   plt.ylabel('Mean sepal width')
   plt.subplot(122)
   plt.bar(species_list, sepal_lengths)
   plt.ylabel('Mean sepal lengths')
   plt.show
✓ 0.5s
<function matplotlib.pyplot.show>
   3.5
                              6
   3.0
                           Mean sepal lengths
 2.5 Wean sebal width 1.0
   10
                              1
   0.5
        setosa versicolor virginica
                                 setosa versicolor virginica
```