

Introduction to Python

Katowice, IntQuant 2024



March 12, 2024

Agenda

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

Section 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

Section 2

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
y = 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

1
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

3

```
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

0.5

Section 3

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
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
...
146	6.7	3.0	5.2	2.3	virginica
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
150	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

Section 4

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
```

Section 5

Visualization: matplotlib

Visualization: matplotlib

<https://matplotlib.org/>

```
import matplotlib.pyplot as plt
✓ 3.7s

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]
✓ 0.6s

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>
```

