intro_python

September 16, 2021

1 Getting started with data analysis in Python

Bartosz Telenczuk, 2021 https://github.com/btel

Download the full notebook from: https://github.com/btel/2021-09-15-eitn-python-tutorial/blob/main/intro_python.ipynb

1.1 Intro to Python for (data) scientists

1.1.1 JupyterLab

- starting
- creating jupyter notebook
- keyboard shorcuts: `Enter` (to enter edito mode), `Shit-Enter` (Run), `Esc` (enter command mode), `M` (markdown, in command mode), `X` (remove cell, in command mode)

1.1.2 Variables

- defining strings and integers
- variables stay defined even if you remove a cell
- indexing with integers and slices
- zero-based indexing!
- type

[ ]:
```
first_name = 'Adam'
age = 100
print(first_name, 'is', age, 'years old')
```

[ ]:
```
first_name[0]
```

[ ]:
```
first_name[1:3]
```

[ ]:
```
type(first_name)
```

[ ]:
```
type(age)
```

Exercise (types) Test the following operations in your notebook. Which output do the produce? What is the type?

```
first_name = 'Adam'
age = 100

variable_1 = 'hello' + first_name
variable_2 = age + 1
variable_3 = 5.1
variable_4 = first_name + 1
```

1.1.3 Built-in functions, methods and and help

- builtin functions
- positional arguments
- string methods
- official Python docs: https://docs.python.org/3/
- types have methods

```
[ ]: max(1, 5, -2)
```

```
[ ]: help(max)
```

```
[ ]: max(first_name)
```

```
[ ]: first_name.upper()
```

Exercise (comparing strings) What will the following program show:

```
rich = "gold"
poor = "tin"
print(max(rich, poor))
```

1.2 Data analysis with pandas

1.2.1 Working with data

- openning files in jupyter lab
- importing extra function libraries (pandas)
- importing csv data with read_csv
- keyword arguments
- showing dataframe

```
[ ]: import pandas as pd
```

```
[ ]: # df = pd.read_csv("https://www4.stat.ncsu.edu/~boos/var.select/diabetes.tab.
      ↪txt", delimiter='\t')
     df = pd.read_csv("diabetes.tab.txt", delimiter='\t')
```

```
[ ]: df
```

Try `pd.read_<Tab>` to find other formats (or look them up in docs)

1.2.2 Plotting

- line and dot plots
- histograms
- scatter plots

[ ]:
```python
df.plot()
```

[ ]:
```python
df.plot('S1', 'S2')
```

[ ]:
```python
df.plot('S1', 'S2', style='.')
```

[ ]:
```python
df.plot(kind='hist')
```

Exercise (plotting styles) Plot the relation between age and BMI using different ploting styles (such as 'o', ':', '-.', 'ro', 'bo')

1.2.3 Indexing data frame

- extract column
- iloc vs loc
- dataframe index
- two-dimensional indexing
- using empty slice

[ ]:
```python
df['AGE'].plot(kind='hist')
```

[ ]:
```python
stats = df.describe()
stats
```

[ ]:
```python
stats.iloc[1]
```

[ ]:
```python
mean_ = stats.loc['mean']
std_ = stats.loc['std']
```

[ ]:
```python
stats.loc['mean' , 'S1']
```

[ ]:
```python
stats.loc[:, 'S1']
```

Exercise (automatic alignment) Normalize all variables in the data frame (subtract mean and divide by standard deviation)

1.3 Linear regression with sklearn

- split data into train/test set
- plotting with matplotlib
- fitting scikit learn linear regression on train set

- predicting on test set

```
[ ]: df.plot(x='S1', y='S2', style='.')
```

```
[ ]: from sklearn.model_selection import train_test_split
```

```
[ ]: X = df.loc[:, ['AGE', 'BMI', 'S1']]
     y = df.loc[:, 'S2']
```

```
[ ]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

```
[ ]: import matplotlib.pyplot as plt
     plt.plot(X_test.loc[:, 'S1'], y_test, '.')
```

```
[ ]: from sklearn.linear_model import LinearRegression
```

```
[ ]: lr = LinearRegression()
     lr.fit(X_train, y_train)
```

```
[ ]: lr.coef_
```

Question Why do we have 3 different coefficients?

```
[ ]: y_pred = lr.predict(X_test)
```

```
[ ]: plt.plot(X_test.loc[:, 'S1'], y_test, '.')
     plt.plot(X_test.loc[:, 'S1'], y_pred, 'r.')
```

```
[ ]:
```

```
[ ]:
```