exercise_01_intro-to-python_tasks_solutions

November 15, 2023

1 Exercise 1: Introduction to Python and Jupyter Lab

1.1 Basic Python

Check the tutorial `exercise_01_intro-python_tutorial.ipynb` before you start working on these exercises.

1.1.1 Use Python as a calculator to compute 256^2

[ ]:

```

```

1.1.2 Create a list that contains the 4 strings "alpha", "beta", "gamma", and "delta". Add "epsilon" to the end of this list

[ ]:

```

```

1.1.3 Create a sublist with the first three elements of this list

[ ]:

```

```

1.1.4 Write a Python program to construct the following pattern using for loops

[ ]:

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

[ ]:

```

```

1.1.5 Write a function that takes 3 input parameters and returns the product of these values

[ ]:

1.1.6 Write a Python function that takes an int as a parameter and checks if the number is prime or not.

[ ]:

1.1.7 Use your function to create a list that contains all primes from 2 to 100

[ ]:

1.1.8 Write a function that simulates dice rolls (6-sided dices)

- Write a function that has the number of dice rolls as the input parameter
- The output is a map with the counts how often each result of a roll (1-6) occurred
- You can use the randomint() function from the random package

[ ]:

```
# load the random package
import random
```

[ ]:

1.1.9 Combine two dictionaries by adding values for common keys.

d1 = {'a': 100, 'b': 200, 'c':300}

d2 = {'a': 300, 'b': 200, 'd':400}

Sample output: {'a': 400, 'b': 400, 'd': 400, 'c': 300}

[ ]:

1.1.10 Create a list of lists. Then, write a list comprehension that creates a list with the lengths of each (sub)list in the primary list

[ ]:

1.1.11 Exceptions

Write your own function *my_division* that gets two variables a and b and returns a divided by b. Catch the ZeroDivisionError that occurs if b equals 0 and return 1 in this case.

How could the same behavior be achieved without catching an exception?

[ ]:

1.1.12 File handling

Write a function that gets a letter and the path to a (text-)file on your harddrive as input parameters and computes a list as its output. The list should contain one number for each line. The i-th line shows, how often the given letter occurs in line number i in the text file. Test your function on “ftest.txt”, which was provided in this exercise!

```
[ ]:
```

1.2 Numpy

In this last set of tasks, you will have to work with numpy, the the fundamental package for scientific computing with Python. Make sure to execute the cell below to import the package, we will usually use the abbreviation np for numpy to obtain shorter code.

```
[23]: import numpy as np
```

1.2.1 Create a numpy array with 20 rows and 10 columns, where all values are set to zero

```
[ ]:
```

1.2.2 Create an array with the same shape containing all natural numbers 0-199

```
[ ]:
```

1.2.3 Create a 4-dimensional array with exactly 16 elements and set all values to a random integer smaller than 5

```
[ ]:
```

1.2.4 What is the mean of all values contained in this array?

```
[ ]:
```

1.2.5 Select a sub-array containing exactly two elements of that array!

```
[ ]:
```

1.2.6 Given a numpy array, return a new numpy array that contains only the elements that are greater than 100

```
[ ]:
```

1.3 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.(source)

We will do a couple of simple things here. For more material see: * 10 minutes to pandas (this is probably more than ten minutes ;)) * Pandas tutorial by Kaggle

[8]:
```
import pandas as pd
```

1.3.1 Read the CSV file `exercise_01_titanic.csv` using pandas.

- see `read_csv` documentation for an example
- make sure the `index` is set to the `PassengerId`

[ ]:

1.3.2 Display the first/last five rows the dataset you loaded

[ ]:

1.3.3 Use the `describe` method to look at the features in the dataset.

[ ]:

1.3.4 Select the `Survived` variable and store it in `y`

[ ]:

1.3.5 Select all variables of type `numeric` except `Survived` and store it in `X`

Hint: Use panda's `select_dtypes`.

[ ]:

1.3.6 Calculate the number of NAs (not defined values) for each variables in `X`

[ ]:

Also, which variable has the most NAs?

1.3.7 Calculate the mean, standard deviation, size and the number of defined values for the `Age` for each `Pclass`

Hint: Use `groupby`

[ ]:

1.4 Plotting

The `matplotlib` library is one of the main libraries for visualziations in Python. `seaborn` builds on `matplotlib` and provides easier access to common plotting scenarios.

1.4.1 Matplotlib

1) Follow the pyplot tutorial including *“Plotting with categorical variables”*.
2) Apply at least one of the plots to the titanic dataset from above.

Note: It makes sense to eventually learn the “object oriented” `matplotlib` API (also see here).

[ ]:

```
```

1.4.2 Seaborn

1) Follow the `seaborn` introduction until *“Plots for categorical data”*.
2) Apply at least one of the plots to the titanic dataset from above.

[ ]:

```
```

1.5 Bonus

- Fit a model that predicts whether a passenger `Survived` using Scikit-Learn. See the supervised machine learning tutorial for reference.
- Evaluate the performance of your model.
- Try to optimize your model.

[ ]:

```
```