Week 4 - Assignment

Programming for Data Science 2025

Exercises for the topics covered in the fourth lecture.

The exercise will be marked as passed if you get **at least 10/15** points.

Exercises must be handed in via **ILIAS** (Homework assignments). Deliver your submission as a compressed file (zip) containing one .py or .ipynb file with all exercises. The name of both the .zip and the .py/.ipynb file must be *SurnameName* of the two members of the group. Example: Annina Helmy + Markus Anwander = *HelmyAnnina_AnwanderMarkus.zip* .

It's important to use comments to explain your code and show that you're able to take ownership of the exercises and discuss them.

You are not expected to collaborate outside of the group on exercises and submitting other groups' code as your own will result in 0 points.

For question about the lecture content or exam, contact: *annina.helmy@students.unibe.ch* with the subject: *Programming for Data Science 2025 - Lecture XY*. For questions about the excercise/grading of excercises, contact: *thea.waldleben@students.unibe.ch* or *patricia.gribi@students.unibe.ch* with the subject: *Programming for Data Science 2025 - Excercise XY*. **Deadline: 14:00, March 20, 2025.**

Exercise 1 - Create Dataframes 5 points

Create a DataFrame *episodes_df* with the columns **ses**, **ep**, and **title**, as below:

ses	ep	title
1	1	One
1	2	Two
2	1	Three
2	2	Four

Create a DataFrame *imdb_df* with the columns **ses**, **ep**, and **score**, as below:

ses	ep	score
1	1	8.4
1	2	8.1

ses	ep	score
2	1	7.9
2	2	7.7

Merge the two DataFrames. Then, find and print the title of the episode with the higest score.

💡 **Hint**: To merge the two dataframes you have to use the *merge* method:

```
merged_df = episodes_df.merge(imdb_df, on=['ses', 'ep'])
```

1. By manipulating the dataframes, find and print the title of the episode with the higest score. (*3 points*)

In [ ]:

In [ ]:
```
###
# YOUR CODE HERE
###
```

2. Change the **score** of the entry with the title "Three" in the DataFrame you created in Task 1 and print the result. The new score should be 6. (*2 points*)

In [ ]:
```
###
# YOUR CODE HERE
###
```

Exercise 2 - Load DataFrames 6 points

1. Load the dataset `spotify_songs.csv` in the data folder. **(2 points)**

- Display the first five rows of the dataset. *(0.5 points)*
- Check the number of rows and columns *(0.5 points)*
- Display the column names of the dataset. *(0.5 points)*
- Count the number of missing values in each column and drop the rows with the missing values. *(0.5 points)*

```
spotify_df = pd.read_csv('./data/spotify_songs.csv')
```

In [ ]:
```
###
# YOUR CODE HERE
###
```

2. Analyzing Dataset **(4 points)**

Work with the cleaned dataset.

- Find the songs with the highest rank (`daily_rank`) on the 17.02.2025 (`snapshot_date`). Print the song name and the artist and the country in which the song had highest rank. *(0.5 points)*
- Find the song with the lowest daily rank on the 17.02.2025. Print the song name and the artist and the country in which the song had highest rank. *(0.5 points)*
- Map the country abbreviations to the full country names. (e.g. ZA -> South Africa) (on the whole cleaned spotify dataset). Meaning you should create a new column where we have the full country name written. *(0.5 points)*
- Plot the artists who had the most songs in this spotify list in Switzerland on the 17.02.2025. Use a bar plot (e.g. x-axis: artist name, y-axis: number of songs).
  - Python: `plt.bar(x = artists_values.index, height = artists_values.values)` . *(1 point)*
- Calculate the mean and median popularity per song on the 17.02.2025 (across the cleaned dataset). Print the results. (Hint: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.groupby.html; group by the song name). *(1 point)*
- Determine the average danceability score of songs by Billie Eilish. *(0.5 points)*

In [ ]:
```
###
# YOUR CODE HERE
###
```

Exercise 3 - DataFrames Ufuncs 4 points

Create the two dataframe *df1* and *df2* with the following code:

In [ ]:
```
import numpy as np
import pandas as pd

df1 = pd.DataFrame(
    np.arange(1, 10).reshape(3, 3),
    columns=["a", "b", "c"],
    index=["1", "2", "3"]
)

df2 = pd.DataFrame(
    np.arange(1, 10).reshape(3, 3) / 2,
    columns=["a", "b", "d"],
    index=["1", "2", "4"]
)
```

1. Add the two dataframes together, with the appropriate pandas method, and print the result. (*0.5 points*)

In [ ]:
```
###
# YOUR CODE HERE
###
```

2. Add the underlying numpy objects of the two dataframes, and print the result. (*0.5 points*)

In [ ]:
```
###
# YOUR CODE HERE
###
```

3. Compare the two results that you obtained and comment if and **why** they are different. (*3 points*)

In [ ]:
```
###
# YOUR COMMENT HERE
###
```