Inlämningsuppgift / Assignment

The assignment is divided upp into 3 parts;

1. Numpy
2. Pandas
3. Exploration

Each section has it's own instructions to follow and questions that must be answered. Please observe that if you use any additional libraries apart from **numpy**, **pandas** or **matplotlib**, you must include an **environment.yml** file such that I can duplicate your conda environment.

Deadline for submitting this assignment is `Monday Feb 21st at 23:59`.

List of files

- *Assignment.ipynb* - which is to be renamed with your name and course town as such: **Firstname.Lastname_TOWN** where TOWN is to be replaced with **MO** for Malmö or **HMS** for Halmstad.
- *countries.csv*
- *covid-countries-data.csv*

Grading

In order to obtain a **G** you must:

> - Complete the whole *Numpy* section.
> - Complete *Part 2*, except the questions marked **Q - VG**.
> - Complete *Part 3*, except *Step 4* and the **VG** question in *Step 5*.

To obtain **VG** you must:

> - Complete all of the steps required in the **G** section.
> - Complete the **Q - VG** questions in *Part 2*.
> - Complete *Step 4* in *Part 3*.

Resources:

- Numpy official tutorial
- Matplotlib

Part 1 - Numpy

The objective of this part of the assignment is to develop a solid understanding of Numpy array operations. In this assignment you will:

> 1. Pick 5 interesting Numpy array functions by going through the documentation: https://numpy.org/doc/stable/reference/routines.html

> 2. Run and modify this Jupyter notebook to illustrate their usage (some explanation and 3 examples for each function). Use your imagination to come up with interesting and unique examples.
> 3. Do not use any of the functions mentioned on slide 11 of lecture notes *6. Datahantering och Numpy*. Choose something new!
> 4. Try to give this section an interesting title & subtitle e.g. "*5 Numpy functions you didn't know you needed*", "*Interesting ways to create Numpy arrays*" etc.

Title Here

Subtitle Here

Write a short introduction about Numpy and list the chosen functions.

- function 1
- function 2
- function 3
- function 4
- function 5

In [1]:

```
import numpy as np
```

In [ ]:

```
# List of functions explained
function1 = np.concatenate  # (change this)
function2 = ???
function3 = ???
function4 = ???
function5 = ???
```

Function 1 - np.concatenate (change this)

Add some explanation about the function in your own words

In [14]:

```
# Example 1 - working (change this)
arr1 = [[1, 2],
        [3, 4.]]

arr2 = [[5, 6, 7],
        [8, 9, 10]]

np.concatenate((arr1, arr2), axis=1)
```

Out[14]:

```
array([[ 1.,  2.,  5.,  6.,  7.],
       [ 3.,  4.,  8.,  9., 10.]])
```

Explanation about example

In [4]:

```
# Example 2 - working
???
```

Explanation about example

In [15]:

```
# Example 3 - breaking (to illustrate when it breaks)
arr1 = [[1, 2],
        [3, 4.]]

arr2 = [[5, 6, 7],
        [8, 9, 10]]

np.concatenate((arr1, arr2), axis=0)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-15-3386a3db1c34> in <module>
      6         [8, 9, 10]]
      7 
----> 8 np.concatenate((arr1, arr2), axis=0)

<__array_function__ internals> in concatenate(*args, **kwargs)

ValueError: all the input array dimensions for the concatenation axis must match
exactly, but along dimension 1, the array at index 0 has size 2 and the array at
index 1 has size 3
```

Explanation about example (why it breaks and how to fix it)

Function 2 - ???

Add some explanations

In [ ]:

```
# Example 1 - working
???
```

Explanation about example

In [ ]:

```
# Example 2 - working
???
```

Explanation about example

In [ ]:

```
# Example 3 - breaking (to illustrate when it breaks)
???
```

Explanation about example (why it breaks and how to fix it)

Some closing comments about when to use this function.

Function 3 - ???

Add some explanations

In [ ]:

```
# Example 1 - working
???
```

Explanation about example

In [ ]:

```
# Example 2 - working
???
```

Explanation about example

In [ ]:

```
# Example 3 - breaking (to illustrate when it breaks)
???
```

Explanation about example (why it breaks and how to fix it)

Some closing comments about when to use this function.

Function 4 - ???

Add some explanations

In [ ]:

```
# Example 1 - working
???
```

Explanation about example

In [ ]:

```
# Example 2 - working
???
```

Explanation about example

In [ ]:

```
# Example 3 - breaking (to illustrate when it breaks)
???
```

Explanation about example (why it breaks and how to fix it)

Some closing comments about when to use this function.

Function 5 - ???

Add some explanations

In [ ]:

```
# Example 1 - working
???
```

Explanation about example

In [ ]:

```
# Example 2 - working
???
```

Explanation about example

In [ ]:

```
# Example 3 - breaking (to illustrate when it breaks)
???
```

Explanation about example (why it breaks and how to fix it)

Some closing comments about when to use this function.

Conclusion

Summarize what was covered in *Part 1*, and where to go next.

Reference Links

Provide links to your references and other interesting articles about Numpy arrays:

- ...

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Part 2 - Pandas

As you go through *Part 2*, you will find a **???** in certain places. To complete this part of the assignment, you must replace all the **???** with appropriate values, expressions or statements to ensure that the notebook runs properly end-to-end.

Some things to keep in mind:

- Make sure to run all the code cells, otherwise you may get errors like `NameError` for undefined variables.
- Do not change variable names, delete cells or disturb other existing code. It may cause problems during evaluation.
- In some cases, you may need to add some code cells or new statements before or after the line of code containing the **???**.
- Questions marked **Q - VG** are for **VG level**.

In [ ]:

```
import pandas as pd
```

Load the data from the supplied CSV file into a Pandas data frame.

In [ ]:

```
countries_df = pd.read_csv('countries.csv')
```

In [ ]:

```
countries_df
```

Q1: How many countries does the dataframe contain? (Show which function/s you use to find this out.)

In [ ]:

```
num_countries = ???
```

In [ ]:

```
print('There are {} countries in the dataset'.format(num_countries))
```

Q2: Retrieve a list of continents from the dataframe?

In [ ]:

```
continents = ???
```

In [ ]:

```
continents
```

Q3: What is the total population of all the countries listed in this dataset?

In [ ]:

```
total_population = ???
```

In [ ]:

```
print('The total population is {}.'.format(int(total_population)))
```

Q4: Create a dataframe containing 10 countries with the highest population.

In [ ]:

```
most_populous_df = ???
```

In [ ]:

```
most_populous_df
```

Q5: Add a new column in `countries_df` to record the overall GDP per country (product of population & per capita GDP).

In [ ]:

```
countries_df['gdp'] = ???
```

In [ ]:

```
countries_df
```

Q - VG: Create a dataframe containing 10 countries with the lowest GDP per capita, among the countries with a population greater than 100 million.

In [ ]:

```
 
```

In [ ]:

```
 
```

Q6: Create a DataFrame that counts the number countries on each continent?

Hint: `groupby`.

In [ ]:

```
country_counts_df = ???
```

In [ ]:

```
country_counts_df
```

Q7: Create a data frame showing the total population of each continent.

In [ ]:
```
continent_populations_df = ???
```

In [ ]:
```
continent_populations_df
```

Next, use the CSV file containing overall Covid-19 stats for various countires, and read the data into another Pandas data frame.

In [ ]:
```
covid_data_df = pd.read_csv('covid-countries-data.csv')
```

In [ ]:
```
covid_data_df
```

Q8: Count the number of countries for which the `total_tests` data is missing.

In [ ]:
```
total_tests_missing = ???
```

In [ ]:
```
print("The data for total tests is missing for {} countries.".format(int(total_
```

Let's merge the two data frames, and compute some more metrics.

Q9: Merge `countries_df` with `covid_data_df` on the `location` column.

In [ ]:
```
combined_df = ???
```

In [ ]:
```
combined_df
```

Q10: Add columns `tests_per_million` , `cases_per_million` and `deaths_per_million` into `combined_df` .

In [ ]:
```
combined_df['tests_per_million'] = combined_df['total_tests'] * 1e6 / combined_
```

In [ ]:
```
combined_df['cases_per_million'] = ???
```

In [ ]:
```
combined_df['deaths_per_million'] = ???
```

In [ ]:
```
combined_df
```

Q11: Create a dataframe with 10 countires that have highest number of tests per million people.

In [ ]:
```
highest_tests_df = ???
```

In [ ]:
```
highest_tests_df
```

Q12: Create a dataframe with 10 countires that have highest number of positive cases per million people.

```
In [ ]: highest_cases_df = ???
```

```
In [ ]: highest_cases_df
```

Q13: Create a dataframe with 10 countires that have highest number of deaths cases per million people?

```
In [ ]: highest_deaths_df = ???
```

```
In [ ]: highest_deaths_df
```

Q - VG: Count number of countries that feature in both the lists of "highest number of tests per million" and "highest number of cases per million".

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

Q - VG: Count number of countries that feature in both the lists "20 countries with lowest GDP per capita" and "20 countries with the lowest number of hospital beds per thousand population". Only consider countries with a population higher than 10 million while creating the list.

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

Part 3 - Exploration

The object of *Part 3* is for you to reflect upon what kind of data is interesting to you and using an example dataset, examine and explain what the data is like and what kind of things you could find out using it.

Pick a real-world dataset of your choice and perform an exploratory data analysis. Focus on documentation and presentation - this Jupyter notebook will also serve as a project report, so make sure to include detailed explanations wherever possible using Markdown cells.

Evaluation Criteria

Your submission will be evaluated using the following criteria:

> - Dataset must contain at least 5 columns and 500 rows of data
> - You must ask and answer at least 4 questions about the dataset
> - Your submission must include at least 4 visualizations (graphs) with axes, title and any other annotations necessary to understand the graph.
> - Your submission must include explanations using markdown cells, apart from the code.
> - Your work must not be plagiarized i.e. copy-pasted for somewhere else.

Dataset repositories:

- UCI repository
- Public datasets
- Google dataset search
- Kaggle datasets

Example datasets:

- https://www.kaggle.com/datasnaek/youtube-new
- https://www.kaggle.com/imdevskp/corona-virus-report
- https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data

Example Projects

Refer to these projects for inspiration:

- Analyzing your browser history using Pandas & Seaborn by Kartik Godawat

- 2019 State of Javscript Survey Results

- 2020 Stack Overflow Developer Survey Results

Follow this step-by-step guide to work on your project.

Step 1: Select a real-world dataset

> - Find an interesting dataset at any of the recommended repositories below.
> - The data should be in CSV format, and should contain at least 5 columns and 500 rows
> - Download the dataset using pandas read_csv function and an url. See example below. (Please note that when downloading from kaggle, you will have to amend this code.) Alternatively, supply the exact link from which you got the dataset and any necessary instructions to download, unpack, load into dataframe etc. in order to get it to work.

```
import pandas as pd
```

```
url =
'https://raw.githubusercontent.com/cs109/2014_data/master/countries.csv'

c = pd.read_csv(url)
```

Step 2: Perform data preparation & cleaning

> - Load the dataset into a data frame using Pandas.
> - Explore the number of rows & columns, ranges of values etc.
> - Handle missing, incorrect and invalid data.
> - Perform any additional steps (parsing dates, creating additional columns, merging multiple dataset etc.).
> - Give a summary of the dataset as it is now, e.g. size, type of categories (qualitative vs. quantitative), quality, distribution etc..

Step 3: Perform exploratory analysis & visualization

> - Compute the mean, sum, range and other interesting statistics for numeric columns.
> - Explore distributions of numeric columns using histograms etc.
> - Explore relationship between columns using scatter plots, bar charts etc.
> - Make a note of interesting insights from the exploratory analysis.

Step 4: Ask & answer questions about the data - VG

> - Ask at least 4 interesting questions about your dataset. What kind of analysis could you do on this data?
> - Answer the questions either by computing the results using Numpy/Pandas or by plotting graphs using Matplotlib.
> - Create new columns, merge multiple dataset and perform grouping/aggregation wherever necessary.
> - Wherever you're using a library function from Pandas/Numpy/Matplotlib etc. explain briefly what it does.

Step 5: Summarize your inferences & write a conclusion

> - Write a summary of what you've learned from the analysis.
> - Include interesting insights and graphs from previous sections.
> - **(VG)** Share ideas for future work on the same topic using other relevant datasets.
> - Share links to resources you found useful during your analysis.

In [ ]: