

My first step is to import all necessary libraries into Jupiter Notebook which is my location where I write this Python code.

In [17]: `!pip install plotly`

Requirement already satisfied: plotly in c:\users\nikola\anaconda3\lib\site-packages (5.9.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\nikola\anaconda3\lib\site-packages (from plotly) (8.0.1)

In [18]: `import plotly.express as px
import pandas as pd
import json
import plotly.io as pio`

Next, I have to use an appropriate command (`pd.read_csv`) in order to read Comma Separated Values (CSV) file. This step is necessary since my dataset (Register of International River Basins 2019) has been written also as a CSV file which allows data to be saved in a table structured format.

In [19]: `file = pd.read_csv(r"C:\Users\Nikola\Desktop\The Project Proposal Geocomputation I\Reg
file.head(5)`

Out[19]:

	BCODE	Basin_Name	Continent_	Area_km2
0	ADIG	Adige	EU	14478
1	AKPA	Akpa	AF	2434
2	ALKL	Alakol	AS	63501
3	ALSK	Alsek	NaN	28220
4	AMCR	Amacuro	SA	3719

Code	Description
Country Name	Name of country
BCODE	Four digit TFDD Basin code for the basin
CCODE	Three letter code for the Country name
BCCODE	Basin Country code that is related to the treaty or agreement (e.g., AMZN_BRA, AMZN is the four-letter code for the Amazon Basin and BRA is the three-letter country code for Brazil.
Alt_CCODE	In previous versions of TFDD, the CCODEs have changed, this lists the older country codes
Basin_Name	Name of the international river basin, basins with multiple major tributaries that are part of the basin are separated by hyphens, if the river is known by different names
Continent	Two letter code for the continent
Area_km2	Area of Basin Country Unit or Basin, Calculated from the World Cylindrical Equal Area Projection

In my next step I will use plotly.express graphing library in order to visualize my dataset, and represent the total number of river basins worldwide in percentages per each continent.

```
In [34]: river_basin_area = px.pie(file, values="Area_km2", names="Continent_", title="Area in  
river_basin_area.show()
```

Some of basins in my dataset belong geografically to North America (abbreviation NA). However, NA has a logical constant of length 1, which contains a missing value indicator, so in Python is categorized as a NULL value. In such a way, all basins from North America (and Central America respectively) in this dataset will be marked as NULL. In the above presented pie chart, 13.9% refers to all basins from North and Central America.

Since I have a lot of river basins that I want to represent visually, in terms of their Area in km2, I will use a bar chart from plotly.express.

```
In [53]: river_basin_area = px.bar(file, x="Area_km2", y="Basin_Name", title="Area in km2 per E  
river_basin_area.show()
```

For my next step, which will include working with pandas and geopandas, I have to import some libraries.

```
In [61]: pip install geopandas
```

```

Collecting geopandas
  Downloading geopandas-0.13.0-py3-none-any.whl (1.1 MB)
  ----- 1.1/1.1 MB 2.9 MB/s eta 0:00:00
Collecting pyproj>=3.0.1
  Downloading pyproj-3.5.0-cp310-cp310-win_amd64.whl (5.1 MB)
  ----- 5.1/5.1 MB 6.0 MB/s eta 0:00:00
Requirement already satisfied: packaging in c:\users\nikola\anaconda3\lib\site-packages (from geopandas) (22.0)
Collecting shapely>=1.7.1
  Downloading shapely-2.0.1-cp310-cp310-win_amd64.whl (1.4 MB)
  ----- 1.4/1.4 MB 8.7 MB/s eta 0:00:00
Collecting fiona>=1.8.19
  Downloading Fiona-1.9.3-cp310-cp310-win_amd64.whl (21.9 MB)
  ----- 21.9/21.9 MB 14.2 MB/s eta 0:00:00
Requirement already satisfied: pandas>=1.1.0 in c:\users\nikola\anaconda3\lib\site-packages (from geopandas) (1.5.3)
Requirement already satisfied: certifi in c:\users\nikola\anaconda3\lib\site-packages (from fiona>=1.8.19->geopandas) (2022.12.7)
Requirement already satisfied: click~=8.0 in c:\users\nikola\anaconda3\lib\site-packages (from fiona>=1.8.19->geopandas) (8.0.4)
Collecting click-plugins>=1.0
  Downloading click_plugins-1.1.1-py2.py3-none-any.whl (7.5 kB)
Collecting munch>=2.3.2
  Downloading munch-2.5.0-py2.py3-none-any.whl (10 kB)
Requirement already satisfied: attrs>=19.2.0 in c:\users\nikola\anaconda3\lib\site-packages (from munch>=2.3.2->fiona>=1.8.19->geopandas) (22.1.0)
Collecting cligj>=0.5
  Downloading cligj-0.7.2-py3-none-any.whl (7.1 kB)
Requirement already satisfied: numpy>=1.21.0 in c:\users\nikola\anaconda3\lib\site-packages (from pandas>=1.1.0->geopandas) (1.23.5)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\nikola\anaconda3\lib\site-packages (from pandas>=1.1.0->geopandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\nikola\anaconda3\lib\site-packages (from pandas>=1.1.0->geopandas) (2022.7)
Requirement already satisfied: colorama in c:\users\nikola\anaconda3\lib\site-packages (from click~=8.0->fiona>=1.8.19->geopandas) (0.4.6)
Requirement already satisfied: six in c:\users\nikola\anaconda3\lib\site-packages (from munch>=2.3.2->fiona>=1.8.19->geopandas) (1.16.0)
Installing collected packages: shapely, pyproj, munch, cligj, click-plugins, fiona, geopandas
Successfully installed click-plugins-1.1.1 cligj-0.7.2 fiona-1.9.3 geopandas-0.13.0 munch-2.5.0 pyproj-3.5.0 shapely-2.0.1
Note: you may need to restart the kernel to use updated packages.

```

For my next step, which will include working with pandas and geopandas, I have to import some libraries.

```
In [66]: pip install mapclassify
```

Collecting mapclassify

Downloading mapclassify-2.5.0-py3-none-any.whl (39 kB)
 Requirement already satisfied: numpy>=1.3 in c:\users\nikola\anaconda3\lib\site-packages (from mapclassify) (1.23.5)
 Requirement already satisfied: pandas>=1.0 in c:\users\nikola\anaconda3\lib\site-packages (from mapclassify) (1.5.3)
 Requirement already satisfied: scikit-learn in c:\users\nikola\anaconda3\lib\site-packages (from mapclassify) (1.2.1)
 Requirement already satisfied: networkx in c:\users\nikola\anaconda3\lib\site-packages (from mapclassify) (2.8.4)
 Requirement already satisfied: scipy>=1.0 in c:\users\nikola\anaconda3\lib\site-packages (from mapclassify) (1.10.0)
 Requirement already satisfied: pytz>=2020.1 in c:\users\nikola\anaconda3\lib\site-packages (from pandas>=1.0->mapclassify) (2022.7)
 Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\nikola\anaconda3\lib\site-packages (from pandas>=1.0->mapclassify) (2.8.2)
 Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\nikola\anaconda3\lib\site-packages (from scikit-learn->mapclassify) (2.2.0)
 Requirement already satisfied: joblib>=1.1.1 in c:\users\nikola\anaconda3\lib\site-packages (from scikit-learn->mapclassify) (1.1.1)
 Requirement already satisfied: six>=1.5 in c:\users\nikola\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pandas>=1.0->mapclassify) (1.16.0)
 Installing collected packages: mapclassify
 Successfully installed mapclassify-2.5.0
 Note: you may need to restart the kernel to use updated packages.

In [74]: `pip install folium`

Collecting folium

Downloading folium-0.14.0-py2.py3-none-any.whl (102 kB)
 ----- 102.3/102.3 kB 652.8 kB/s eta 0:00:00
 Requirement already satisfied: Jinja2>=2.9 in c:\users\nikola\anaconda3\lib\site-packages (from folium) (3.1.2)
 Requirement already satisfied: numpy in c:\users\nikola\anaconda3\lib\site-packages (from folium) (1.23.5)
 Collecting branca>=0.6.0
 Downloading branca-0.6.0-py3-none-any.whl (24 kB)
 Requirement already satisfied: requests in c:\users\nikola\anaconda3\lib\site-packages (from branca) (2.28.1)
 Requirement already satisfied: MarkupSafe>=2.0 in c:\users\nikola\anaconda3\lib\site-packages (from Jinja2>=2.9->folium) (2.1.1)
 Requirement already satisfied: idna<4,>=2.5 in c:\users\nikola\anaconda3\lib\site-packages (from requests->branca) (3.4)
 Requirement already satisfied: certifi>=2017.4.17 in c:\users\nikola\anaconda3\lib\site-packages (from requests->branca) (2022.12.7)
 Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\nikola\anaconda3\lib\site-packages (from requests->branca) (1.26.14)
 Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\nikola\anaconda3\lib\site-packages (from requests->branca) (2.0.4)
 Installing collected packages: branca, folium
 Successfully installed branca-0.6.0 folium-0.14.0
 Note: you may need to restart the kernel to use updated packages.

In [70]: `import geopandas as gpd
import matplotlib as mb
import mapclassify`

In [71]: `river_basins = gpd.read_file(r"C:\Users\Nikola\Desktop\The Project Proposal Geocomputa
river_basins`

Out[71]:

	OBJECTID	MRBID	RIVER_BASIN	CONTINENT	SEA	OCEAN	SUM_SUB_AREA	Shape_L
0	1	1201	ATOUI	Africa	North Atlantic	Atlantic Ocean	27695.4	11.8
1	2	1202	BANDAMA	Africa	Gulf of Guinea	Atlantic Ocean	99254.1	18.5
2	3	1203	BIA	Africa	Gulf of Guinea	Atlantic Ocean	10957.1	7.1
3	4	1204	CAVALLY	Africa	Gulf of Guinea	Atlantic Ocean	29622.9	13.0
4	5	1205	CESTOS (also NUON)	Africa	North Atlantic	Atlantic Ocean	12785.0	8.0
...
515	516	6258	VARDAR	Europe	Mediterranean Sea	Atlantic Ocean	24524.3	11.6
516	517	6259	VISTULA	Europe	Baltic Sea	Atlantic Ocean	192637.8	46.1
517	518	6260	WESER	Europe	North Sea	Atlantic Ocean	44775.6	19.7
518	519	6902	KURA	Europe	Caspian Sea	---	189819.2	34.6
519	520	6903	VOLGA	Europe	Caspian Sea	---	1424437.8	176.0

520 rows x 10 columns

In order to use command `explore()` I installed following libraries: `matplotlib`, `folium` and `mapclassify`

```
In [75]: river_basins.explore()
```

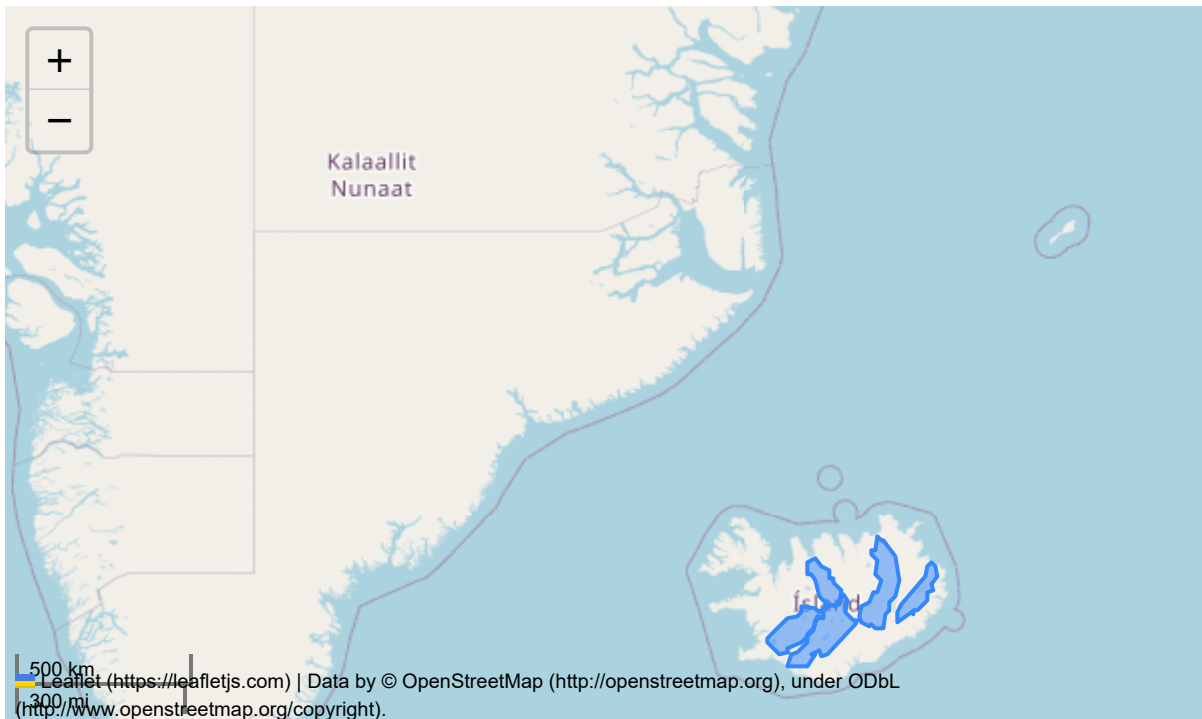
Out[75]: Make this Notebook Trusted to load map: File -> Trust Notebook



River basins in Europe are presented below.

```
In [86]: river_basins.query("CONTINENT == 'Europe'").explore()
```

Out[86]:



The river basin of Danube in Europe (runs through my country of birth - Republic of Serbia).

```
In [85]: river_basins.query("RIVER_BASIN == 'DANUBE']").explore()
```

Out[85]:

