Geographic Information Systems

Project Report

Proposed by :

Aziza Ben Tanfous :   n° 27156

Johanna Rauberger :   n° 27492

Proposed to :

Professor Rui Figueira

2023-2024

<u>Problem 01 – Regions vulnerable to flood risk</u>

In this exercise, we used QGIS to determine the regions vulnerable to flood risk due to river overflow during the raining season. The study area extent was defined by specific coordinate values.

Data Acquisition: We downloaded the ***SRTM DEM data*** from the provided URL and loaded it into QGIS. We also loaded the ***Parcels2023.gpkg*****.**

Processing Steps: The procedure involved using the ***Raster calculator*** to generate a binary layer that represents the extent of river overflow. This binary layer was then ***converted to a polygon vector layer*** and ***clipped*** to the study area extent. After ***filtering*** for polygons with a height of 17.0m or less, the ***area of the vulnerable regions was computed*** and stored in the attribute table of the vector layer. Furthermore, a new vector layer was created to ***categorize the vulnerable regions into high risk, moderate risk, and low risk based on their heights***. Subsequently, the ***area of the vulnerable regions in each risk category was calculated*** and stored in the attribute table of the vector layer.

Results: Within the given study area[1], about **235 km²** are vulnerable to flood risk due to river overflow. Out of that area, 16704 ha classified as *high risk* (elevation <10m), 4751 ha as *moderate risk* (10m ≤ elevation <15m) and 2012 ha as *low risk* (15m ≤ elevation ≤ 17m), see Fig1.

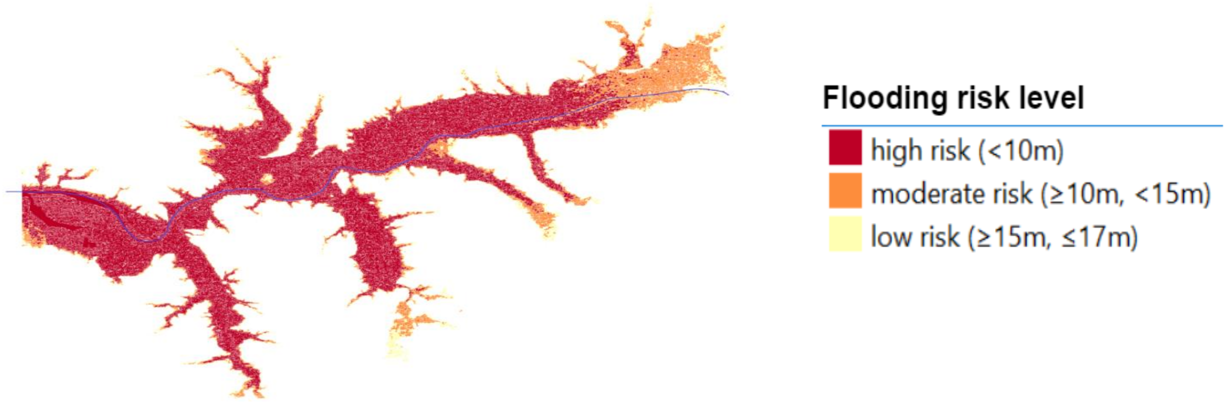


Fig1: Regions vulnerable to flood risk due to river overflow classified by risk level.

[1] The study area extent given in ETRS PT-TM06 coordinate values were converted into EPSG:4326 WGS 84 format to comply with input requirements to download the SRTM DEM data.

Problem 02 – Soil use

For this problem, our assigned area was Region 4 from the ***EditingZones*** layer, which represents a specific polygon. Our objective was to identify and delineate parcels within ***Region 4*** using Google Satellite imagery. Through careful analysis of land use patterns, boundaries, and visible features, we aimed to accurately represent the soil use within the area. To store the vector data and attribute information, we created a new GeoPackage file named ***Probl2_27156.gpkg*** as per the assignment requirements. We ensured consistent spatial reference by setting the GeoPackage file to use the ***ETRS-TM06 CRS (EPSG: 3763).*** Within the GeoPackage file, we established a dedicated vector layer named ***Parcels4*** to represent the parcels in ***Region 4***. Digitizing approximately ***10 parcels*** based on our interpretation of the satellite imagery, we assigned the corresponding soil use codes and descriptions by referring to the provided "TableSoilUse.xls" file using the joint operation. Additionally, we assigned owners to each parcel, obtained from the "***Owners.xls***" file using the joint operation. All vector layers, including the ***Parcels4*** layer, were stored within the GeoPackage file to ensure complete coverage of the assigned area in Region 4.

Problem 03 - all parcels in the region delimited by Cadastre and those created in Problem 02

Data Preparation: We gathered the necessary input datasets, including the ***Cadastre GDS***, ***Parcels_27156*** from Problem 02, and ***SoilUse GDS***. Additionally, modify the ***TableSoilUse.xls*** to include a new class ***UseCode: 50*** with the description ***Other***.

Processing Steps: We began by opening the required datasets in QGIS, including ***Cadastre***, ***Parcels_27156***, ***SoilUse***, and ***TableSoilUse.xls***. We created a new Geopackage layer named ***ParcelsNew_27156*** and saved it in the ***DataOut*** folder. The attribute table of ***ParcelsNew_27156*** was updated to include fields for soil use, Parcels_27156, and owner. A legend based on soil use was created using the "***Categorized" symbolization method***. We used the ***field calculator*** to compute the area of each parcel and converted it to hectares. The area owned by each owner and the total area for each soil use class were calculated using the ***groupstat*** function and saved in a ***CSV files.*** Lastly, we used the ***Extract by Attribute tool*** to create a new layer named ***Use_27156*** that represents only agriculture and forest regions from the ***ParcelsNew_27156*** dataset, excluding other classifications.

Problem 04 - the relative index of susceptibility to groundwater pollution of the region

Data Preparation: We gathered the necessary input datasets, including the ***ParcelsNew_27156*** and the ***SoilType*** GDS.

Processing Steps: The analysis commenced by accessing the necessary datasets, including ***ParcelsNew_27156*** and ***SoilType GDS***. Subsequently, a new attribute, namely $IIuu$ was introduced to the ***ParcelsNew_27156*** layer utilizing the field calculator. Similarly, the ***SoilType GDS*** was augmented with a new attribute called $IIuu$ through the same procedure. Through a data join operation, the information from the ***SoilType GDS*** was merged with the ***ParcelsNew_27156*** layer, resulting in the creation of the ***GwPollIndex*** layer. The calculation of the IIp value was carried out by employing the formula $IIpp = 0.25IIuu \times IIt$. To visualize the ***GwPollIndex layer***, symbology techniques were employed, utilizing the classification provided in the assignment document. Subsequently, the select by expression functionality was utilized to identify and isolate attributes where $IIpp$ exceeded 3. Finally, the selected features were exported as a new layer denoted as ***HvHGwPollIndex***.

Problem 05 – Highest and lowest producers of olive oil in the world

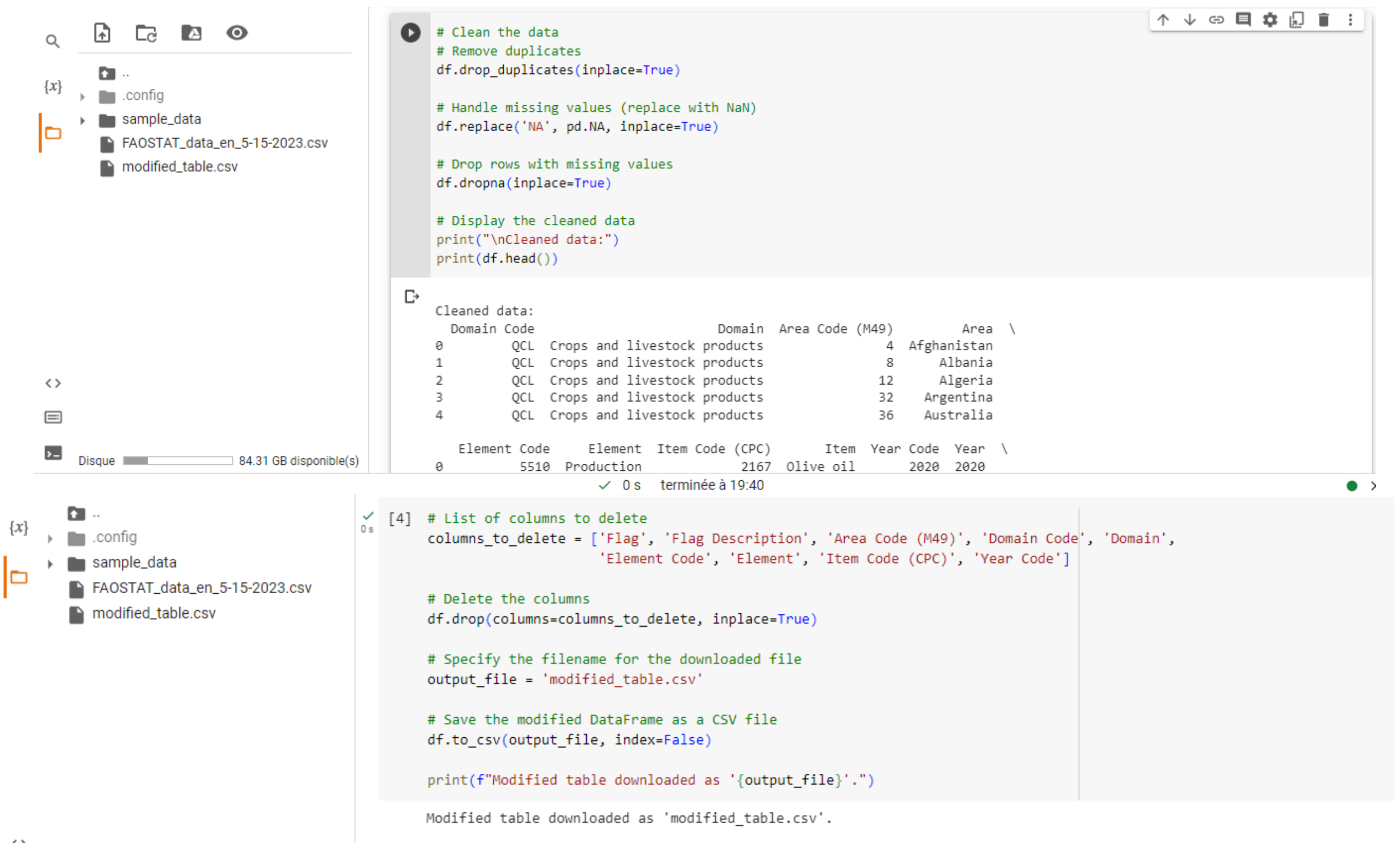
Question: What countries produce olive oil, and which country had the highest and lowest production of olive oil in 2020?

Data Acquisition: We downloaded the data from the FAOSTAT database and the world map from open data soft.

Data Management:

In order to ensure the quality and reliability of the data, we conducted a thorough data **cleaning** process. The steps involved removing duplicate entries with ***df.drop_duplicates(),*** handling missing values by replacing 'NA' with ***NaN*** using ***df.replace('NA', pd.NA),*** dropping rows with missing values using ***df.dropna().*** The resulting clean data was then displayed using ***print(df.head()),*** demonstrating the removal of duplicates, handling of missing values, and any necessary data type corrections. These measures contributed to improved data integrity and enhanced the suitability of the dataset for subsequent analysis.

We *removed* also certain columns that were deemed unnecessary or redundant. These columns included Flag, Flag Description, Area Code (M49), Domain Code, Domain, Element Code, Element, Item Code (CPC), and Year Code. Removing these columns helped streamline the dataset and focus on the relevant information for analysis. Finally we downloaded the *modified_table.csv* (Fig 2) . The *modified_table* has been meticulously normalized to meet the requirements of the 3rd normal form, guaranteeing data consistency, eliminating redundancy, and streamlining data management.



```
# Clean the data
# Remove duplicates
df.drop_duplicates(inplace=True)

# Handle missing values (replace with NaN)
df.replace('NA', pd.NA, inplace=True)

# Drop rows with missing values
df.dropna(inplace=True)

# Display the cleaned data
print("\nCleaned data:")
print(df.head())
```

```
Cleaned data:
  Domain Code                        Domain  Area Code (M49)         Area  \
0         QCL  Crops and livestock products                4  Afghanistan
1         QCL  Crops and livestock products                8      Albania
2         QCL  Crops and livestock products               12      Algeria
3         QCL  Crops and livestock products               32    Argentina
4         QCL  Crops and livestock products               36    Australia

   Element Code     Element  Item Code (CPC)       Item  Year Code  Year  \
0          5510  Production             2167  Olive oil       2020  2020
```

✓ 0 s terminée à 19:40

```
[4] # List of columns to delete
    columns_to_delete = ['Flag', 'Flag Description', 'Area Code (M49)', 'Domain Code', 'Domain',
                         'Element Code', 'Element', 'Item Code (CPC)', 'Year Code']

    # Delete the columns
    df.drop(columns=columns_to_delete, inplace=True)

    # Specify the filename for the downloaded file
    output_file = 'modified_table.csv'

    # Save the modified DataFrame as a CSV file
    df.to_csv(output_file, index=False)

    print(f"Modified table downloaded as '{output_file}'.")
```

```
Modified table downloaded as 'modified_table.csv'.
```

Fig 2: Cleaning Code using python

Processing Steps:

In the GIS analysis workflow, we commenced by opening the dataset comprising the *olive oil production data* and the *world map* in QGIS. After confirming that both datasets share the *same Coordinate Reference System (CRS) of WGS 84_EPSG:4326*, we proceeded to create a new layer named "*OliveOil_producers*" by performing a join operation between the olive oil production data and the spatial data layer containing the country boundaries. To streamline the layer, we selectively retained only the name and continent attributes from the world map layer, discarding unnecessary features. Next, utilizing the Rule-Based symbology option in QGIS, we developed a thematic map based on the olive oil production values. The color blue was assigned to represent the olive oil producers on the map, ensuring clarity and visual distinction. Additionally, to standardize the production values, we introduced a new feature named "***production***" that ensured all values were expressed in decimal format with three digits after the decimal point. To identify the highest and lowest producers, we employed the "***select by expression***" functionality, enabling us to select the relevant features based on their production values. Subsequently, we extracted the corresponding layer containing the selected features for further analysis. In order to present our findings effectively, we utilized the new print layout feature in QGIS to create a well-composed map. The map layout incorporated essential elements such as a title, scale bar, north arrow, and a legend, providing a comprehensive and visually appealing representation of the olive oil production analysis. By following this professional workflow, we were able to analyze and visualize the olive oil production data, facilitating a clear understanding of the highest producers and conveying the information in an organized and visually engaging manner.

Results:

The production of olive oil is a significant industry worldwide, involving several countries renowned for their unique flavors and characteristics in olive oil products. Spain, Italy, Greece, Turkey, Tunisia, Morocco, Portugal, and United States are among the major contributors to the global olive oil market.Spain led global olive oil production in 2020 with 1,356,411,000 tonnes. In contrast, Ukraine faced challenges due to its continental climate and agricultural focus, resulting in negligible olive oil production of 0 tonnes. The country's limited demand for olive oil further hindered investment in its cultivation and production (Fig3).

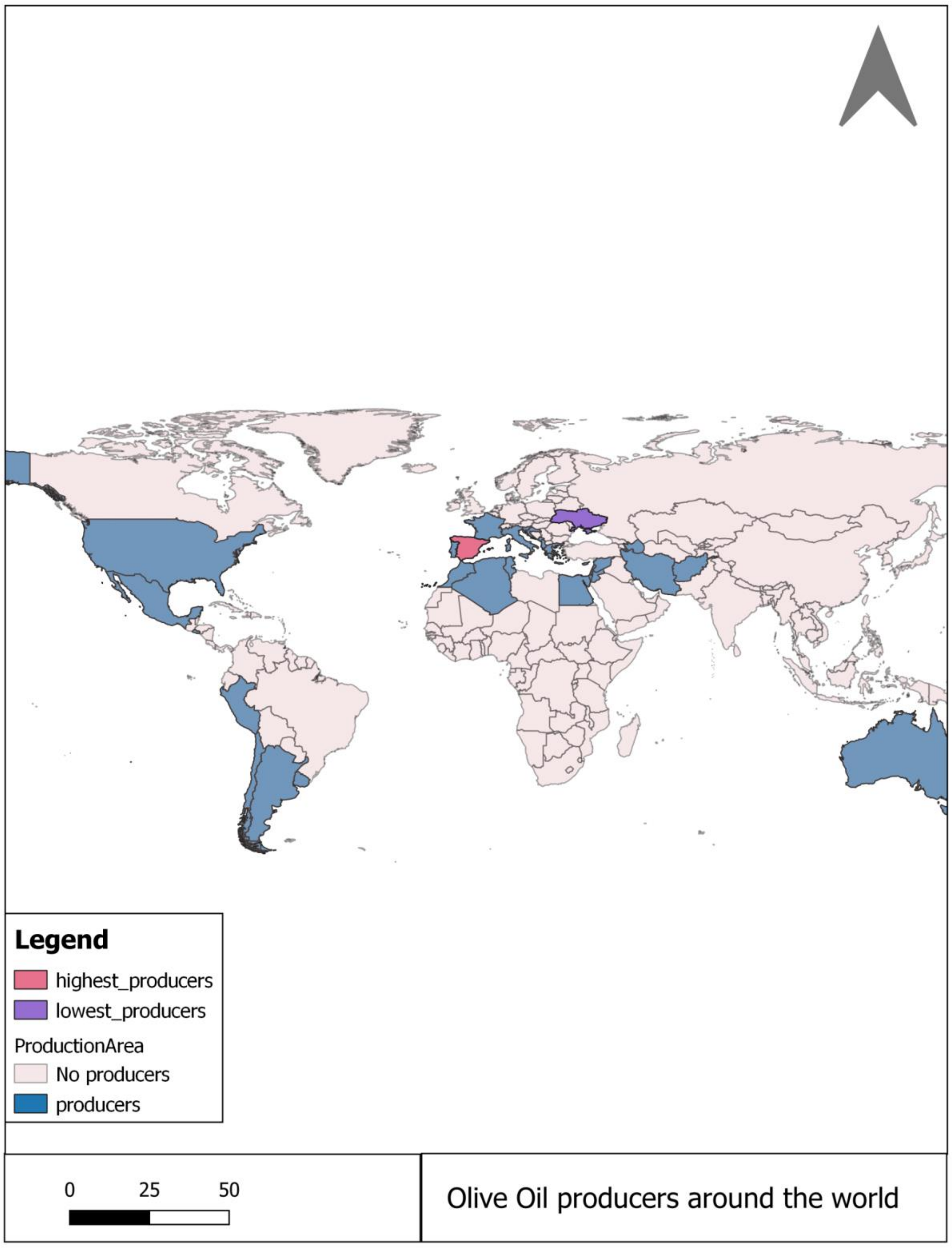


Fig 3: Production of olive oil