DATA VISUALIZATION

ALI MOHAMED SAYED AHMED

20221449583

Steps:

1. load data:
   (Acquire)
2. pre-processing :
   (Parse and filter)
3. Data visualization:
   (Represent, Refine and Interact)

2. pre-processing:

```python
f wrangle(filepath1):

  #define my data set
  df=pd.read_csv(filepath1)
  df.drop(df.columns[-1], axis=1, inplace = True)
  df=df.drop(columns=["PopulationCensus (Cp)1986-11-17","PopulationCensus (Cf)1996-11-19","PopulationCensus (Cf)2006-11-11","PopulationCensus (Cf)2017-03-28","Unnamed: 10
  #reset index
  df.reset_index(drop=True, inplace=True)
  #change columns name
  new_columns = {
      'Name': 'Governorate',
      'Abbr.': 'Abbreviation',
      'Native': 'Native Name',
      'Capital': 'Capital City',
      'AreaA (km²)': 'Area (km²)',
      'PopulationEstimate (E)2023-07-01': 'Population Estimate (July 2023)'
  }

  # Rename columns based on the new_columns dictionary
  df.rename(columns=new_columns, inplace=True)

  return df
```

this function reads a dataset from a CSV file, removes unnecessary columns, resets the index, renames columns based on a dictionary of mappings, and returns the modified DataFrame.

Data after this step be:

```
df=wrangle("enviroment.csv")
df
```

	Governorate	Abbreviation	Native Name	Capital City	Area (km²)	Population Estimate (July 2023)
0	Ad-Daqahiyah [Dakahlia]	DQH	الدقهلية	Al-Manṣūrah	3,538	7,050,004
1	Al-Baḥr al-Aḥmar [Red Sea]	BAH	البحر الأحمر	Al-Ghurdaqah	120,000	403,077
2	Al-Buḥayrah [Beheira]	BHR	البحيرة	Damanhūr	9,826	6,878,289
3	Al-Fayyūm [Fayoum]	FYM	الفيوم	Al-Fayyūm	6,068	4,080,645
4	Al-Gharbiyah [Gharbia]	GHB	الغربية	Ṭanṭā	1,942	5,439,085
5	Al-Iskandariyah [Alexandria]	ISK	الإسكندرية	Al-Iskandariyah	2,300	5,546,663
6	Al-Ismāʿīliyah [Ismaïlia]	ISM	الإسماعيلية	Al-Ismāʿīliyah	5,067	1,452,743
7	Al-Jīzah [Giza]	JIZ	الجيزة	Al-Jīzah	13,184	9,514,540
8	Al-Minūfiyah [Monufia]	MNF	المنوفية	Shibīn al-Kawm	2,499	4,736,945
9	Al-Minyā	MNY	المنيا	Al-Minyā	32,279	6,337,595
10	Al-Qāhirah [Cairo]	QAH	القاهرة	Al-Qāhirah	3,085	10,248,385
11	Al-Qalyūbyah [Qalyubia]	QLY	القليوبية	Banhā	1,124	6,137,688
12	Al-Uqṣor [Luxor]	UQS	الأقصر	Al-Uqṣur	2,960	1,400,640
13	Al-Wādī al-Jadīd [New Valley]	WJD	الوادي الجديد	Al-Khārijah	440,098	266,926

3. Data visualization:

- is the graphical representation of information and data.
- By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.
- essential to analyze massive amounts of information and make data-driven decisions.

```
PImage mapImage; // Declaring a variable to store an image
int rowCount;   // Declaring a variable to store the number of rows in the data
Table data;    // Declaring a variable to store tabular data
float dataMin = MAX_FLOAT;
float dataMax = MIN_FLOAT;
String message = "";   // Declaring an empty string variable to store a message

void setup() {
  size(430, 463);
  mapImage = loadImage("egypt.jpg");  // Loading an image named "egypt.jpg"
  data = new Table("data.tsv");      // Loading tabular data from a file named "data.tsv"
  rowCount = data.getRowCount();    // Getting the number of rows in the data table

  // Find the minimum and maximum values.
  for (int row = 0; row < rowCount; row++) {
    float value = data.getFloat(row, 4);    // Getting a float value from the 4th column of the current row
    if (value > dataMax) {
      dataMax = value;
    }                                  // Updating the max and min value
    if (value < dataMin) {
      dataMin = value;
    }}
  }
```

this snippet initializes necessary variables, loads data and an image, and prepares data by finding the minimum and maximum values

```
void draw() {
  background(255);    // Set background color to white
  image(mapImage, 0, 0);
  smooth();
  for (int row = 0; row < rowCount; row++) {
    String abbrev = data.getRowName(row);       //the abbreviation from the current row
    float x = data.getFloat(abbrev, 2);    // Get the x-coordinate from the 2nd column of the current row
    float y = data.getFloat(abbrev, 3);  // Get the y-coordinate from the 3rd column of the current row
    drawData(x, y, abbrev);               // Set fill color to black
  }

  if (!message.equals("")) {      // Draw the message
    fill(0);
    textSize(14);
    textAlign(CENTER);
    text(message, width / 2, height - 20);
  }}
void drawData(float x, float y, String abbrev) {
  String governmentName = data.getString(abbrev, 1); // Get government name from column index 1
  float value = data.getFloat(abbrev, 4);    // Get data value from the 4th column
  float percent = norm(value, dataMin, dataMax);
  color between = lerpColor(#87CEEB, #800080, percent); // Sky Blue to Purple
  fill(between);     //Set fill color to interpolated color
  ellipse(x, y, 7, 7);    // Draw a circle at (x, y) with a diameter of 7 pixels
}
```

It renders visual elements on a canvas, including a map image and data points represented by colored circles. The data points' colors vary based on the normalized data value. there is a message to display, it appears at the center-bottom

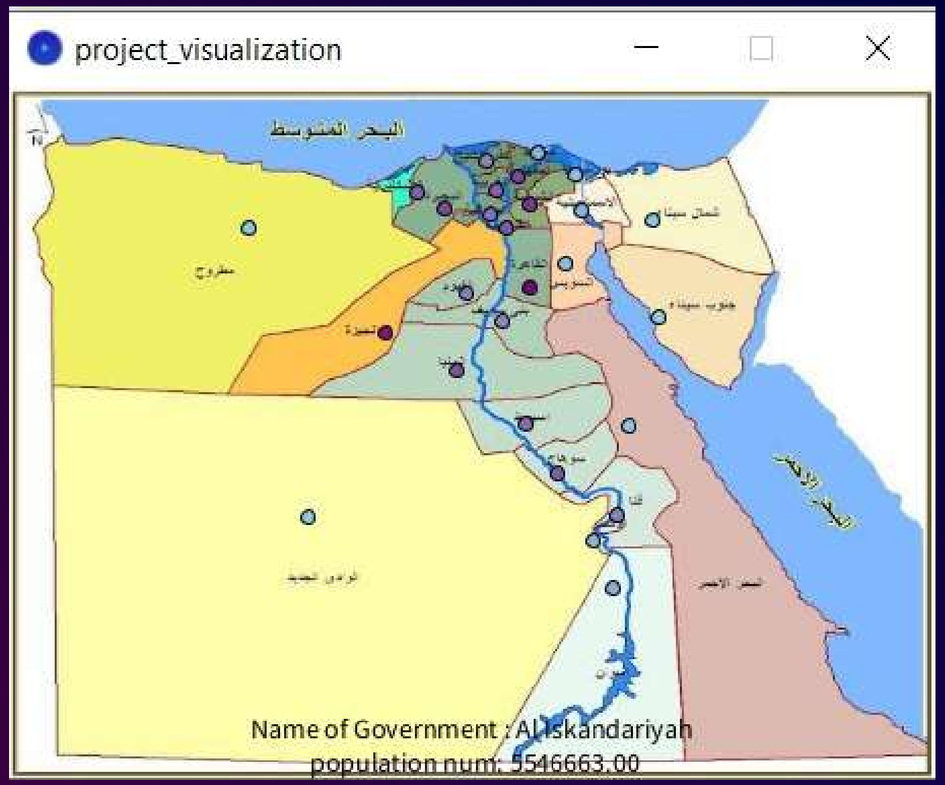
```
void mouseClicked() {
  for (int row = 0; row < rowCount; row++) {
    String abbrev = data.getRowName(row);       //Get the abbreviation from the current row
    float x = data.getFloat(abbrev, 2);         // Get the x-coordinate from the 2nd column of the current row
    float y = data.getFloat(abbrev, 3);       // Get the y-coordinate from the 3rd column of the current row


    if (dist(x, y, mouseX, mouseY) < 7 / 2) { // Check if mouse is over the point
      displayData(abbrev);                     // If mouse is over the point, display data for that point
      break;
    }
  }
}

void displayData(String abbrev) {             // Function to display data for a given abbreviation
  String governmentName = data.getString(abbrev, 1); // Assuming government names are in column index 1
  float value = data.getFloat(abbrev, 4);
   // Construct the message to display
  message = "Name of Government : " + governmentName + "\n population num: " + nf(value, 0, 2);
}
```

- When the mouse is clicked, it iterates through each row of the data table and checks if the mouse is over any of the data points. If the mouse is over a point, it displays the data associated with that point.
- this part is for interacting and display the name of gov and area

Final result of this:



where:

- the area with max of population represent by Purple color
- the area with min of population represent by blue color
- The visualization might be interactive since it responds to mouse clicks.
- When a user clicks on a data point, additional information about that point is displayed