**Basic and Specialized Visualization Tools**

Visualization tools play a crucial role in data analysis and communication. These are essential for extracting insights and presenting information in a concise manner to both technical and non-technical audiences. In this module, you will create a diverse range of plots using Matplotlib, the data visualization library. Throughout this module, you will learn about area plots, histograms, bar charts, pie charts, box plots, and scatter plots. You will also explore the process of creating these visualization tools using Matplotlib.

**Learning Objectives**

* Describe a box plot with an illustration and create it using Matplotlib
* Explore an area plot with an illustration and create it using Matplotlib
* Define a histogram with an illustration and create it using Matplotlib
* Describe a bar chart with an illustration and create it using Matplotlib
* Discover a pie chart with an illustration and create it using Matplotlib
* Discover a scatter plot with an illustration and create it using Matplotlib

# **Basic Visualization Tools**

## **Area Plots**

1. **What is an Area Plot?**
   * An area plot (or area chart) displays the magnitude and proportion of multiple variables over a continuous axis, typically representing time.
   * It is similar to a line plot but fills the area below the line with color to emphasize the cumulative magnitude of the variables.
2. **Use Cases:**
   * Area plots are effective for comparing two or more quantities.
   * Common applications include tracking stock market performance, visualizing population demographics, and displaying resource distribution.
3. **Data Preparation:**
   * The dataset consists of annual immigration data from various countries to Canada from 1980 to 2013.
   * Each row represents a country with metadata and numerical data of immigration.
4. **Creating an Area Plot:**
   * The process involves:
     + Sorting the dataframe by cumulative immigration.
     + Creating a new dataframe for the top countries.
     + Transposing the dataframe to have years as indices.
     + Using Matplotlib to generate the area plot.

Code Example

Here’s a step-by-step code example to create an area plot using Matplotlib:

import pandas as pd

import matplotlib.pyplot as plt

# Sample DataFrame creation (replace this with your actual data)

data = {

'Country': ['Afghanistan', 'Albania', 'India', 'China', 'UK', 'Philippines', 'Pakistan'],

'1980': [1000, 500, 20000, 15000, 12000, 8000, 6000],

'1981': [1100, 600, 21000, 16000, 13000, 9000, 7000],

# Add more years as needed

}

df\_canada = pd.DataFrame(data)

df\_canada.set\_index('Country', inplace=True)

# Calculate cumulative immigration

df\_canada['Total'] = df\_canada.sum(axis=1)

# Sort by cumulative immigration

df\_canada\_sorted = df\_canada.sort\_values(by='Total', ascending=False)

# Select top 5 countries

df\_top5 = df\_canada\_sorted.head(5).drop(columns='Total')

# Transpose the DataFrame

df\_top5\_transposed = df\_top5.transpose()

# Create the area plot

plt.figure(figsize=(10, 6))

df\_top5\_transposed.plot(kind='area', alpha=0.5)

plt.title('Area Plot of Immigration to Canada (Top 5 Countries)')

plt.xlabel('Year')

plt.ylabel('Number of Immigrants')

plt.legend(title='Countries')

plt.show()

Explanation of the Code

* **Data Preparation:**
  + A sample DataFrame df\_canada is created with immigration data. Replace this with your actual dataset.
  + The set\_index method is used to set the country names as the index.
* **Cumulative Immigration Calculation:**
  + The total immigration for each country is calculated using sum(axis=1) and stored in a new column called 'Total'.
* **Sorting and Selecting Top Countries:**
  + The DataFrame is sorted in descending order based on the 'Total' column.
  + The top 5 countries are selected, and the 'Total' column is dropped.
* **Transposing the DataFrame:**
  + The DataFrame is transposed so that years become the index, which is necessary for plotting.
* **Creating the Area Plot:**
  + The plot function is called with kind='area' to generate the area plot.
  + The plot is customized with titles and labels for clarity.

Conclusion

In this lecture, you learned how to create an area plot using Matplotlib, which visually represents the cumulative immigration trends of the top countries to Canada. Make sure to practice with your dataset to reinforce your understanding!

## **Histogram**

* **Definition**: A histogram is a graphical representation of the frequency distribution of a numeric data set. It partitions the data into bins and counts how many data points fall into each bin.
* **Vertical Axis**: Represents the frequency (number of data points) in each bin.
* **Horizontal Axis**: Represents the bins, which are intervals of the data range.

Steps to Create a Histogram:

1. **Partition the Data**: Divide the range of numeric values into equal-width bins.
2. **Count Data Points**: For each bin, count how many data points fall within its range.
3. **Draw Bars**: Each bin is represented by a bar whose height corresponds to the count of data points in that bin.

Example Code to Create a Histogram using Matplotlib:

import matplotlib.pyplot as plt

import pandas as pd

# Sample data: Immigration data to Canada for the year 2013

data = {

'Country': ['Afghanistan', 'Albania', 'Algeria', 'Argentina', 'Australia'],

'2013': [58639, 15699, 12345, 23456, 34567]

}

# Create a DataFrame

df\_canada = pd.DataFrame(data)

# Create a histogram for the year 2013

plt.figure(figsize=(10, 6))

plt.hist(df\_canada['2013'], bins=5, edgecolor='black')

plt.title('Distribution of Immigration to Canada in 2013')

plt.xlabel('Number of Immigrants')

plt.ylabel('Frequency')

plt.xticks(range(0, 40000, 5000)) # Set x-ticks for better readability

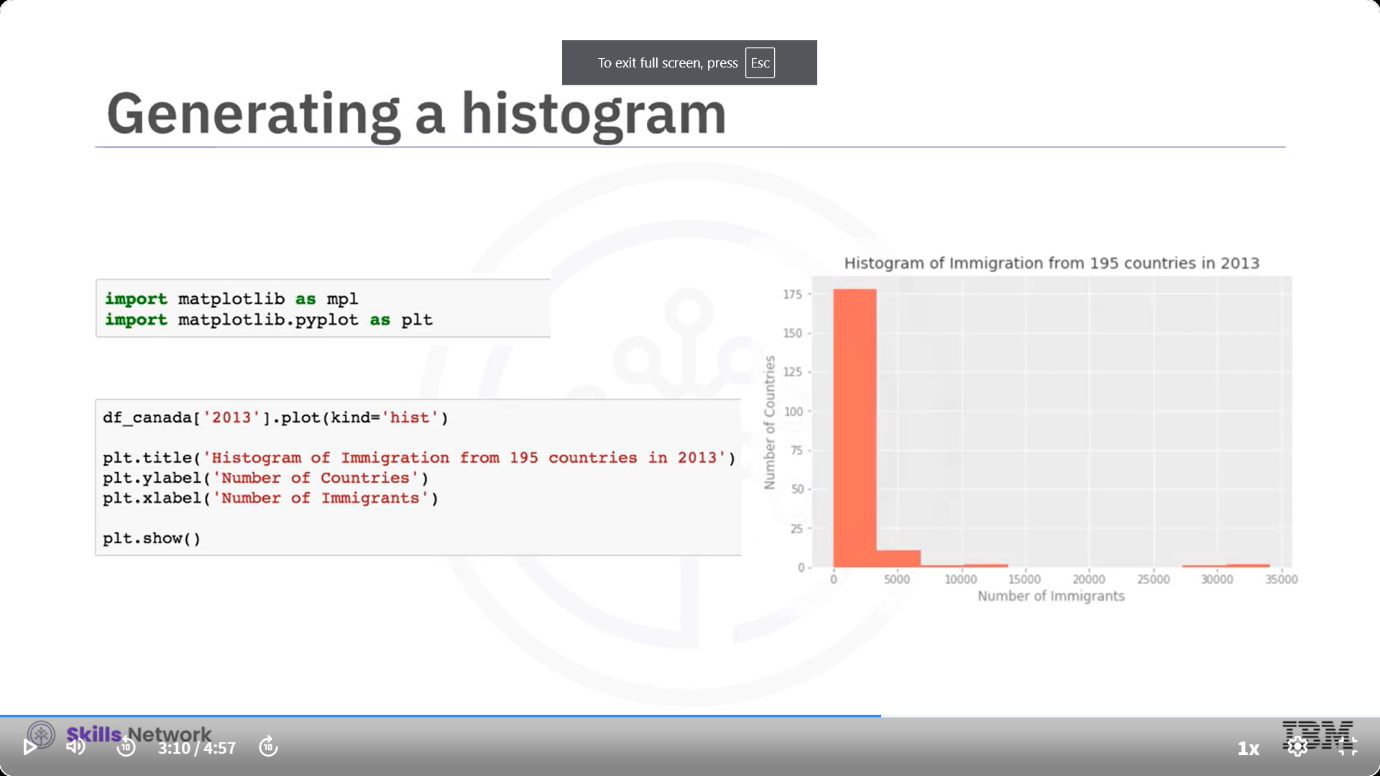
plt.show()

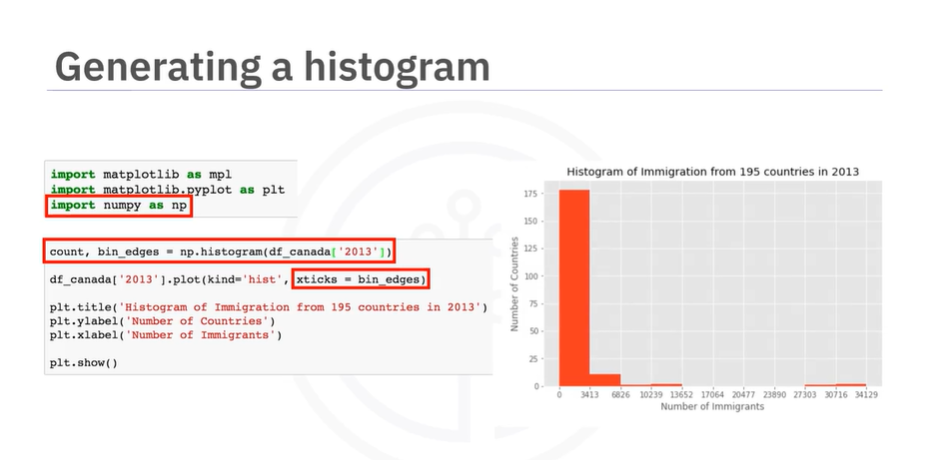
Explanation of the Code:

* **Import Libraries**:
  + matplotlib.pyplot is used for plotting.
  + pandas is used for data manipulation.
* **Sample Data**: A dictionary is created with countries and their corresponding immigration numbers for 2013.
* **Create DataFrame**: The data is converted into a pandas DataFrame for easier manipulation.
* **Plotting the Histogram**:
  + plt.figure(figsize=(10, 6)): Sets the size of the figure.
  + plt.hist(...): Creates the histogram.
    - df\_canada['2013']: Specifies the data to plot.
    - bins=5: Divides the data into 5 bins.
    - edgecolor='black': Adds a border to the bars for clarity.
* **Adding Titles and Labels**:
  + plt.title(...), plt.xlabel(...), and plt.ylabel(...) add titles and labels to the axes.
* **Setting x-ticks**: plt.xticks(...) customizes the ticks on the x-axis for better readability.
* **Display the Plot**: plt.show() renders the histogram.

Conclusion:

This code will generate a histogram that visually represents the distribution of immigration numbers to Canada in 2013. You can modify the data and parameters to explore different datasets and bin sizes.





## **Bar Charts**

* **Bar Chart**: A bar chart is a type of plot where the length of each bar is proportional to the value it represents. It's used to compare values of a variable at a specific point in time.
* **Matplotlib**: A popular Python library for creating static, animated, and interactive visualizations.

Steps to Create a Bar Chart:

1. **Import Matplotlib**: Start by importing the necessary library.
2. **Prepare Data**: Create a DataFrame with the data you want to visualize.
3. **Plot the Bar Chart**: Use the plot function with kind='bar'.
4. **Customize the Chart**: Add titles and labels to the axes.
5. **Display the Chart**: Use the show function to display the figure.

Example Code:

import matplotlib.pyplot as plt

import pandas as pd

# Sample data: Immigration from Iceland to Canada (1980-2013)

data = {

'Year': [1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013],

'Immigration': [100, 150, 120, 130, 140, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260, 270, 280, 290, 300, 310, 320, 330, 340, 350, 360, 370, 380, 390, 400, 410, 420, 430, 440]

}

# Create a DataFrame

df\_iceland = pd.DataFrame(data)

# Plotting the bar chart

plt.figure(figsize=(10, 6))

plt.bar(df\_iceland['Year'], df\_iceland['Immigration'], color='blue')

# Adding title and labels

plt.title('Immigration from Iceland to Canada (1980-2013)')

plt.xlabel('Year')

plt.ylabel('Number of Immigrants')

# Display the chart

plt.show()

Explanation of the Code:

* **Importing Libraries**: matplotlib.pyplot is imported for plotting, and pandas is used for data manipulation.
* **Data Preparation**: A dictionary is created with years and corresponding immigration numbers, which is then converted into a DataFrame.
* **Creating the Bar Chart**: The bar function is used to create the bar chart, where the x-axis represents the years and the y-axis represents the number of immigrants.
* **Customization**: The chart is customized with a title and axis labels for clarity.
* **Displaying the Chart**: Finally, plt.show() is called to render the chart.

## **Pie Charts**

* **Definition**: A pie chart is a circular statistical graphic divided into segments to illustrate numerical proportions.
* **Example**: A pie chart can represent the party-wise percentage of seats won in the Canadian federal election.

Creating a Pie Chart with Matplotlib:

1. **Data Preparation**:
   * Use **pandas** to group data by a specific category (e.g., continent) and sum the values.
   * Example DataFrame: DF\_continents with rows for each continent and columns for years.
2. **Importing Libraries**:
3. import pandas as pd

import matplotlib.pyplot as plt

1. **Creating the Pie Chart**:
   * Use the plot function on the total column of the DataFrame.
   * Set kind='pie' to generate a pie chart.
   * Add a title and display the chart.
2. # Assuming DF\_continents is already created
3. plt.figure(figsize=(8, 8)) # Optional: Set figure size
4. DF\_continents['total'].plot(kind='pie', title='Immigration by Continent')

plt.show()

1. **Exploding Slices**:
   * Use the explode parameter to offset slices for emphasis.
2. explode = [0.1 if x < 0.1 else 0 for x in DF\_continents['total']]
3. DF\_continents['total'].plot(kind='pie', explode=explode, title='Immigration by Continent')

plt.show()

Criticism of Pie Charts:

* Some critics argue that pie charts do not consistently display accurate data.
* Bar charts are often preferred for better data representation.

Conclusion:

* Pie charts are useful for visualizing proportions but should be used judiciously.

## **Box Plots**

* **Definition**: A box plot is a statistical representation of data distribution through five main dimensions:
  + **Minimum**: The lowest value, calculated as the first quartile minus 1.5 times the interquartile range (IQR).
  + **First Quartile (Q1)**: The value at the 25th percentile of the sorted data.
  + **Median (Q2)**: The middle value of the sorted data.
  + **Third Quartile (Q3)**: The value at the 75th percentile of the sorted data.
  + **Maximum**: The highest value, calculated as the third quartile plus 1.5 times the IQR.
* **Outliers**: Displayed as individual dots outside the minimum and maximum values.

Creating a Box Plot with Matplotlib:

To create a box plot using Matplotlib, follow these steps:

1. **Import Libraries**:
2. import pandas as pd

import matplotlib.pyplot as plt

1. **Prepare the Data**: Assume you have a DataFrame df\_canada with immigration data. You will filter it for Japan:
2. # Example DataFrame creation (replace with your actual data)
3. data = {
4. 'Year': [1980, 1981, 1982, 1983, 1984],
5. 'Immigration': [1000, 1500, 1200, 1300, 1100]
6. }
7. df\_canada = pd.DataFrame(data)
8. # Filter for Japan

df\_japan = df\_canada[['Year', 'Immigration']]

1. **Create the Box Plot**:
2. plt.figure(figsize=(10, 6))
3. plt.boxplot(df\_japan['Immigration'], vert=False)
4. plt.title('Box Plot of Japanese Immigration to Canada (1980-2013)')
5. plt.xlabel('Number of Immigrants')

plt.show()

Explanation of the Code:

* **Importing Libraries**: You need pandas for data manipulation and matplotlib.pyplot for plotting.
* **Data Preparation**: Create a DataFrame with immigration data and filter it for Japan.
* **Box Plot Creation**: Use plt.boxplot() to create the box plot. The vert=False parameter makes the plot horizontal. Finally, add titles and labels for clarity.

Conclusion:

Box plots are useful for visualizing the distribution of data and identifying outliers. By following the steps above, you can create a box plot to analyze immigration data effectively.

## **Scatter Plots**

* **Definition**: A **scatter plot** is a type of plot that displays values for typically two variables against each other. It helps to determine if a correlation exists between the two variables.
* **Example**: The video mentions a scatter plot of **income versus education**. It shows that individuals with more years of education tend to earn a higher income.
* **Creating a Scatter Plot with Matplotlib**:
  1. **Data Preparation**: You need a DataFrame (e.g., df\_canada) that contains the data you want to visualize.
  2. **New DataFrame**: Create a new DataFrame (df\_total) that shows each year and the corresponding total number of immigrants.
  3. **Importing Libraries**: Import Matplotlib and its pyplot interface.
  4. **Plotting**: Use the plot function with kind='scatter' to generate the scatter plot.
  5. **Setting Parameters**: Specify the x and y parameters for the horizontal and vertical axes, respectively.
  6. **Final Touches**: Add a title and label the axes, then display the plot.

Example Code

Here’s a code snippet to create a scatter plot using Matplotlib:

import matplotlib.pyplot as plt

import pandas as pd

# Sample data creation

data = {

'year': [1980, 1981, 1982, 1983, 1984],

'total': [1000, 1500, 2000, 2500, 3000]

}

df\_total = pd.DataFrame(data)

# Creating a scatter plot

plt.figure(figsize=(10, 6))

plt.scatter(df\_total['year'], df\_total['total'], color='darkblue', s=100) # s is the size of markers

plt.title('Total Immigration to Canada (1980-1984)')

plt.xlabel('Year')

plt.ylabel('Total Immigrants')

plt.grid(True)

plt.show()

Explanation of the Code

* **Importing Libraries**: The code starts by importing the necessary libraries: matplotlib.pyplot for plotting and pandas for data manipulation.
* **Sample Data**: A sample dataset is created using a dictionary, which is then converted into a DataFrame.
* **Creating the Plot**:
  + plt.figure(figsize=(10, 6)): Sets the size of the plot.
  + plt.scatter(...): Creates the scatter plot with specified x and y values, color, and size of markers.
  + plt.title(...), plt.xlabel(...), plt.ylabel(...): Adds a title and labels to the axes.
  + plt.grid(True): Adds a grid for better readability.
  + plt.show(): Displays the plot.

## **Plotting Directly with matplotlib**

1. **Matplotlib Overview**
   * Matplotlib is a comprehensive plotting library for Python.
   * The pyplot module provides a convenient interface for creating and customizing plots.
2. **Data Visualization vs. Data Storytelling**
   * **Data Visualization**: Creating informative charts to explore patterns and trends.
   * **Data Storytelling**: Crafting a narrative around the data to engage the audience.

Code Examples

1. Importing Libraries

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

* **matplotlib.pyplot**: Used for creating static, animated, and interactive visualizations.
* **numpy**: Used for numerical operations and creating arrays.
* **pandas**: Used for data manipulation and analysis.

2. Creating a Line Plot

years = np.arange(1980, 2014) # Generates an array from 1980 to 2013

immigrants = np.random.randint(2000, 10000, size=34) # Random integers for immigrants

plt.plot(years, immigrants) # Create a line plot

plt.title('Immigrants between 1980 to 2013') # Title of the plot

plt.xlabel('Years') # X-axis label

plt.ylabel('Total Immigrants') # Y-axis label

plt.grid(True) # Enable grid

plt.show() # Display the plot

* **Parameters**:
  + years: X-axis data (years).
  + immigrants: Y-axis data (number of immigrants).
* **Usage**: Line plots are useful for showing trends over time.

3. Creating a Scatter Plot

plt.scatter(years, immigrants) # Create a scatter plot

plt.title('Immigrants between 1980 to 2013')

plt.xlabel('Years')

plt.ylabel('Total Immigrants')

plt.grid(True)

plt.show()

* **Usage**: Scatter plots are effective for showing the relationship between two variables.

4. Customizing Plots

plt.plot(years, immigrants, linestyle='--', marker='o', color='b') # Dashed line with circle markers

plt.title('Immigrants between 1980 to 2013')

plt.xlabel('Years')

plt.ylabel('Total Immigrants')

plt.xlim(1975, 2015) # Set x-axis limits

plt.ylim(0, 12000) # Set y-axis limits

plt.grid(True)

plt.legend(['Immigrants']) # Add legend

plt.show()

* **Customization Options**:
  + linestyle: Style of the line (e.g., solid, dashed).
  + marker: Style of the data points (e.g., 'o' for circles).
  + color: Color of the line/markers.

5. Creating a Bar Plot

plt.bar(years, immigrants) # Create a bar plot

plt.title('Number of Immigrants per Year')

plt.xlabel('Years')

plt.ylabel('Total Immigrants')

plt.show()

* **Usage**: Bar plots are useful for comparing quantities across different categories.

6. Creating a Pie Chart

plt.pie(immigrants[:5], labels=years[:5], autopct='%1.1f%%') # Pie chart for the first five years

plt.title('Immigrants from 1980 to 1984')

plt.show()

* **Parameters**:
  + labels: Labels for each slice.
  + autopct: Format for displaying percentages.

Conclusion

* Matplotlib provides a flexible interface for creating various types of plots.
* Understanding the differences between data visualization and data storytelling is crucial for effective communication of insights.

Summary: Basic and Specialized Visualization Tools

Congratulations! You have completed this module. At this point in the course, you know:

* A pie chart is a circular statistical graphic, divided into segments, to illustrate numerical proportion.
* The process of creating a pie chart involves importing Matplotlib to represent a large set of data over a period of time.
* A box plot is a way of statistically representing given data distribution through five main dimensions.
* The five main dimensions are minimum, first quartile, median, third quartile, and maximum.
* You can create a box plot using Matplotlib.
* A scatter plot displays values pertaining to typically two variables against each other.
* The process of creating a scatter plot involves importing Matplotlib to visualize a large set of data.
* Matplotlib is a versatile plotting library that offers a flexible interface for creating various types of plots.
* Matplotlib’s Pyplot module offers a convenient way to create and customize plots quickly.
* Data Storytelling is the ‘art of storytelling’ that involves creating a narrative around the data.
* Data visualization is an important aspect of data storytelling and involves creating engaging visuals.