**Introduction to Data Visualization Tools**

Data visualization is a way of presenting complex data in a form that is graphical and easy to understand. When analyzing large volumes of data and making data-driven decisions, data visualization is crucial. In this module, you will learn about data visualization and some key best practices to follow when creating plots and visuals. You will discover the history and the architecture of Matplotlib. Furthermore, you will learn about basic plotting with Matplotlib and explore the dataset on Canadian immigration, which you will use during the course. Lastly, you will analyze data in a data frame and generate line plots using Matplotlib.

**Learning Objectives**

* Use Matplotlib to create line plots
* Use Matplotlib to create plots employing Jupyter notebook
* Explore the dataset on immigration to Canada
* Discuss data visualization and its importance
* Discover the history of Matplotlib and its architecture
* Identify the steps to analyze data in Pandas data frame

# **Welcome to the Course**

1. **Introduction to Data Visualization**:
   * Understanding the importance of data visualization in data analysis.
   * Overview of different types of visualizations and their applications.
2. **Matplotlib Basics**:
   * Introduction to Matplotlib library for creating static, interactive, and animated visualizations in Python.
   * Basic plotting techniques including line plots, scatter plots, and bar charts.
3. **Advanced Plotting with Matplotlib**:
   * Customizing plots with titles, labels, legends, and annotations.
   * Working with subplots and customizing axes.
4. **Seaborn for Statistical Plots**:
   * Introduction to Seaborn library for statistical data visualization.
   * Creating plots like histograms, box plots, and violin plots.
5. **Interactive Visualizations with Plotly**:
   * Using Plotly for creating interactive plots.
   * Exploring different types of interactive charts such as 3D plots and geographic maps.
6. **Data Visualization Best Practices**:
   * Understanding the principles of effective data visualization.
   * Tips for choosing the right type of visualization for your data.

These topics provide a comprehensive foundation for creating effective and informative visualizations using Python.

**How to Make the Most of this Course**

Welcome to **Data Visualization with Python**. This is a 5-week course, where you can expect to spend approximately 3-5 hours per module.

The content of this course can be applied to multiple Specializations and Professional Certificate Programs:

* IBM Data Science Professional Certificate
* IBM Data Analyst Professional Certificate
* Applied Data Science Specialization

Here are some tips to ensure you get the best experience and gain the required knowledge for a job interview or a certification, or for that matter, to enhance your skills.

**Tip 1: Get familiar with the course content**

Browse through the module structure to understand the topics and associated assets. Having a better understanding of the module structure will help you navigate through the course easily and also help you to understand the mandatory and optional topics in the module.

Familiarize yourself with the order of topics, which will help you connect the ideas and form a cohesive whole. The outline with the modules and lessons in the course is available later in this topic.

**Tip 2: Create your schedule to complete the course**

Set a target date when you want to finish the course and schedule time each day to study. Set a reasonable goal for yourself to complete each module.

**Tip 3: Be an active learner**

* Set some time aside for studying the course. Avoid distractions.
* When watching instructional videos, make sure to take notes. You can download transcripts and highlight parts of the narration that are important. Alternatively, you can take notes elsewhere.
* Keeping yourself engaged is important to learn more and measure your understanding.
* Read the readings thoroughly as they help connect the instructional videos and real-life scenarios.
* The course contains Expert Viewpoints, which bring out answers to Data Visualization -related questions from experts in the field. Compare what the experts say to what you learned in the instructional videos and readings.
* Make sure to review the demo videos and attempt the labs. Reviewing these course materials will help you get used to the different Data Visualization-related activities needed in a Data Visualization role.
* Attempt all practice quizzes and read the feedback for correct and incorrect responses. Retake them until you get all of the answers correct.
* Complete all of the graded quizzes. If you get a response wrong, you can revisit the topic for review. Review the topic, identify the correct response, and understand why that answer is correct. While there is a minimum passing score, you can retake graded quizzes for a higher score.
* Discussion forums are optional. However, actively participating in the discussion forums is recommended. Provide thoughtful responses to the prompts and engage by responding to others’ posts.

**Tip 4: Inform your family and friends about course goals**

Signing up for the course was the first step. You must make yourself accountable and commit to the course. Inform your family and friends about this commitment. If there are topics where you need a different perspective, get family and friends involved.

**Tip 5: Follow through with your plan**

You create a schedule for yourself in Tip 2. Make sure to stay motivated and achieve your set goals. Commit and execute! Be unstoppable and keep yourself motivated!

**Syllabus**

**Module 1: Introduction to Data Visualization Tools**

Lesson 1: Introduction to Data Visualization

**Module 2: Basic and Specialized Visualization Tools**

Lesson 1: Basic Visualization Tools

Lesson 2: Specialized Visualization Tools

**Module 3: Advanced Visualizations and Geospatial Data**

Lesson 1: Advanced Visualization Tools

Lesson 2: Visualizing Geospatial Data

**Module 4: Creating Dashboards with Plotly and Dash**

Lesson 1: Creating Dashboards with Plotly and Dash

**Module 5: Final Project and Exam**

Lesson 1: Final Assessment

Lesson 2: Course Wrap-Up

**Syllabus**

**Abstract**

One of the most important skills of successful data scientists and data analysts is the ability to tell a compelling story by visualizing data and findings in an approachable and stimulating way. In this course you will learn many ways to effectively visualize both small and large-scale data. You will be able to take data that at first glance has little meaning and present that data in a form that conveys insights.

This course will teach you to work with many Data Visualization tools and techniques. You will learn to create various types of basic and advanced graphs and charts like: Waffle Charts, Area Plots, Histograms, Bar Charts, Pie Charts, Scatter Plots, Word Clouds, Choropleth Maps, and many more! You will also create interactive dashboards that allow even those without any Data Science experience to better understand data and make more effective and informed decisions.

You will learn hands-on by completing numerous labs and a final project to practice and apply the many aspects and techniques of Data Visualization using Jupyter Notebooks and a Cloud-based IDE. You will use several data visualization libraries in Python, including Matplotlib, Seaborn, Folium, Plotly & Dash.

**Course Learning Objectives**

*After completing this course, a learner will be able to:*

* Implement data visualization techniques and plots using Python libraries, such as Matplotlib, Seaborn, and Folium to tell a stimulating story.
* Create different types of charts and plots such as line, area, histograms, bar, pie, box, scatter, and bubble.
* Create advanced visualizations such as waffle charts, word clouds, regression plots, maps with markers, & choropleth maps.
* Generate interactive dashboards containing scatter, line, bar, bubble, pie, and sunburst charts using the Dash framework and Plotly library.

**Module 1**

**Title: Introduction to Data Visualization Tools**

**Description**

Data visualization is a way of presenting complex data in a form that is graphical and easy to understand. When analyzing large volumes of data and making data-driven decisions, data visualization is crucial. In this module, you will learn about data visualization and some key best practices to follow when creating plots and visuals. You will discover the history and the architecture of Matplotlib. Furthermore, you will learn about basic plotting with Matplotlib and explore the dataset on Canadian immigration, which you will use during the course. Lastly, you will analyze data in a data frame and generate line plots using Matplotlib.

**Objectives**

*By the end of this week, you will be able to:*

* Discuss data visualization and its importance
* Discover the history of Matplotlib and its architecture
* Use Matplotlib to create plots employing Jupyter notebook
* Explore the dataset on immigration to Canada
* Identify the steps to analyze data in Pandas data frame
* Use Matplotlib to create line plots

**Activities**

Lesson 0: Welcome to the Course

* Welcome to the Course
* How to Make the Most of this Course
* Syllabus

Lesson 1: Introduction to Data Visualization

* Overview of Data Visualization
* Types of Plots
* Plot Libraries
* Introduction to Matplotlib
* Basic Plotting with Matplotlib
* Dataset on Immigration to Canada
* Line Plots
* Hands-on Lab: Exploring and Pre-processing a Dataset using Pandas
* Hands-on Lab: Introduction to Matplotlib and Line Plots
* Practice Quiz: Introduction to Data Visualization
* Module 1 Summary: Introduction to Data Visualization Tools
* Module 1 Cheat Sheet
* Module 1 Graded Quiz: Introduction to Data Visualization Tools

**Module 2**

**Title: Basic and Specialized Visualization Tools**

**Description**

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.

**Objectives**

*By the end of this week, you will be able to:*

* 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
* Describe a box plot with an illustration and create it using Matplotlib
* Discover a scatter plot with an illustration and create it using Matplotlib

**Activities**

Lesson 1: Basic Visualization Tools

* Area Plots
* Histograms
* Bar Charts
* Hands-on Lab: Area Plots, Histograms, and Bar Charts
* Practice Quiz: Basic Visualization Tools

Lesson 2: Specialized Visualization Tools

* Pie Charts
* Box Plots
* Scatter Plots
* Hands-on Lab: Pie Charts, Box Plots, Scatter Plots, and Bubble Plots
* Plotting Directly with Matplotlib
* Hands-on Lab: Plotting Directly with Matplotlib
* Practice Quiz: Specialized Visualization Tools
* Module 2 Summary: Basic and Specialized Visualization Tools
* Module 2 Cheat Sheet
* Module 2 Graded Quiz: Basic and Specialized Visualization Tools

**Module 3**

**Title: Advanced Visualizations and Geospatial Data**

**Description**

Advanced visualization tools are sophisticated platforms that provide a wide range of advanced features and capabilities. These tools provide an extensive set of options that help create visually appealing and interactive visualizations. In this module, you will learn about waffle charts and word cloud including their application. You will explore Seaborn, a new visualization library in Python, and learn how to create regression plots using it. In addition, you will learn about folium, a data visualization library that visualizes geospatial data. Furthermore, you will explore the process of creating maps using Folium and superimposing them with markers to make them interesting. Finally, you will learn how to create a Choropleth map using Folium.

**Objectives**

*By the end of this week, you will be able to:*

* Explore waffle charts and word cloud along with their application
* Describe Seaborn and explore the process of generating attractive regression plots
* Describe Folium and explore the process of creating maps
* Explore the process of superimposing markers on maps using Foilum
* Describe Choropleth maps with the help of an illustration
* Explore the process of creating a Choropleth map using Folium

**Activities**

Lesson 1: Advanced Visualizations and Geospatial Data

* Waffle Charts & Word Cloud
* Seaborn and Regression Plots
* Hands-on Lab: Waffle Charts, Word Clouds, and Regression Plots
* Practice Quiz: Advanced Visualization Tools

Lesson 2: Visualizing Geospatial Data

* Introduction to Folium
* Maps with Markers
* Choropleth Maps
* Hands-on Lab: Creating Maps and Visualizing Geospatial Data
* Practice Quiz: Visualizing Geospatial Data
* Module 3 Summary: Advanced Visualizations and Geospatial Data
* Module 3 Cheat Sheet
* Module 3 Graded Quiz: Advanced Visualizations and Geospatial Data

**Module 4**

**Title: Creating Dashboards with Plotly and Dash**

**Description**

Dashboards and interactive data applications are crucial tools for data visualization and analysis because they provide a consolidated view of key data and metrics in a visually appealing and understandable format. In this module, you will explore the benefits of dashboards and identify the different web-based dashboarding tools in Python. You will learn about Plotly and discover how to use Plotly graph objects and Plotly express to create charts. You will gain insight into Dash, an open-source user interface Python library, and its two components. Finally, you will gain a clear understanding of the callback function and determine how to connect core and HTML components using callback.

**Objectives**

*By the end of this week, you will be able to:*

* Identify different web-based dashboarding tools available in Python
* Explore Plotly and its two sub-modules
* Use Plotly graph objects and Plotly express to create charts
* Discover Dash and its two components
* Describe the callback function
* Determine the process of connecting core and HTML components using callback

**Activities**

Lesson 1: Creating Dashboards with Plotly and Dash

* Dashboarding Overview
* Additional Resources for Dashboards
* Introduction to Plotly
* Additional Resources for Plotly
* Plotly Basics: Scatter, Line, Bar, Bubble, Histogram, Pie, Sunburst
* Practice Quiz: Creating Dashboards with Plotly

Lesson 2: Working with Dash

* Introduction to Dash
* Overview of Cloud IDE lab environment
* Dash Basics: HTML and Core Components
* Additional Resources for Dash
* Make Dashboards Interactive
* Additional Resources for Interactive Dashboards
* Add Interactivity: User Inputs and Callbacks
* Understanding the Lab Environment
* Flight Delay Time Statistics Dashboard
* Practice Quiz: Working with Dash
* Module 4 Summary: Creating Dashboards with Plotly and Dash
* Module 4 Cheat Sheet
* Module 4 Graded Quiz: Creating Dashboards with Plotly and Dash

**Module 5**

**Title: Final Project and Exam**

**Description**

The primary focus of this module is to practice the skills gained earlier in the course and then demonstrate those skills in your final assignment. For the final assignment you will analyze historical automobile sales data covering periods of recession and non-recession. You will bring your analysis to life using visualization techniques and then display the plots and graphs on dashboards. Finally, you will submit your assignment for peer review and you will review an assignment from one of your peers. To wrap up the course you will take a final exam in the form of a timed quiz.

**Objectives**

*By the end of this week, you will be able to:*

* Practice visualization skills
* Practice creating a dashboard
* Create various visualizations using a number of plot libraries
* Create a dashboard and add interactivity
* Review and grade an assignment submitted by peers

**Activities**

Lesson 1: Practice Project

* Practice Project Overview
* Practice Assignment: Part 1 - Analyzing wildfire data in Australia
* Practice Assignment: Part 2 - Creating Dashboards

Lesson 2: Final Project

* Final Project Overview
* Final Assignment: Part 1 - Create Visualizations using Matplotlib, Seaborn & Folium
* Final Assignment: Part 2 - Create Dashboard with Plotly and Dash
* Final Assignment: Part 3 - Submission and Grading
* Final Exam: Data Visualization with Python - Timed Quiz

Lesson 3: Course Wrap Up

* Course Summary
* Congratulations and Next Steps
* Thanks from the Course Team

# **Introduction to Data Visualization**

## **Overview of Data Visualization**

* **Definition**: Data visualization is the graphical representation of data and information, helping to create visual representations of complex datasets.
* **Forms of Data Visualization**:
  + Basic charts and graphs (e.g., line graphs, bar charts)
  + Complex visualizations (e.g., interactive dashboards, maps, infographics)
* **Importance of Data Visualization**:
  + Simplifies understanding of complex datasets.
  + Highlights patterns, trends, and relationships.
  + Communicates insights effectively to stakeholders.
  + Aids in decision-making by identifying opportunities and potential problems.
* **Use Cases**:
  + News publications (e.g., The New York Times) for COVID-19 data.
  + Companies (e.g., Airbnb) for pricing strategies.
  + Music platforms (e.g., Spotify) for user listening habits.
  + Entertainment (e.g., Netflix) for operational insights.
* **Best Practices for Effective Visualization**:
  + Choose the appropriate visualization type for the data.
  + Keep visualizations simple and easy to read.
  + Clearly label axes and provide context.
  + Focus on the main message and avoid unnecessary clutter.
  + Tailor visualizations to the audience's needs.
* **Examples of Good and Bad Visualization**:
  + Good: Clear, focused visualizations that convey the main message.
  + Bad: Overly complex charts that confuse the audience.

## **Types of Plots**

1. **Line Plot**
   * Displays data points connected by straight lines.
   * **Use Cases**: Trends over time (e.g., stock prices, temperature changes).
   * **Caution**: Can be misleading if scales are not chosen carefully.
2. **Bar Plot**
   * Uses rectangular bars to represent data magnitude.
   * **Use Cases**: Comparing different categories (e.g., sales by product).
   * **Caution**: Inaccurate axis scales can distort the data representation.
3. **Scatter Plot**
   * Shows values for two variables using Cartesian coordinates.
   * **Use Cases**: Examining relationships between variables (e.g., house prices vs. size).
   * **Caution**: Outliers can significantly affect interpretation.
4. **Box Plot**
   * Displays data distribution and key statistical measures (median, quartiles).
   * **Use Cases**: Comparing distributions across categories (e.g., salaries across departments).
   * **Caution**: Ignoring outliers can lead to misinterpretation.
5. **Histogram**
   * Represents the distribution of a dataset using bars for frequency within intervals.
   * **Use Cases**: Understanding data distribution, skewness, and variability.
   * **Caution**: Choice of bins affects data representation; too few or too many can mislead.

Summary

* Different plots serve various purposes in data visualization.
* Proper handling of scales and outliers is crucial for accurate representation.

## **Plot Libraries**

* **Importance of Data Visualization**: It helps in gaining insights and effectively communicating complex information.
* **Popular Plot Libraries**:
  + **Matplotlib**:
    - General-purpose plotting library.
    - Supports various plots (line, scatter, bar, histogram, pie, box, heat maps).
    - Highly customizable and integrates well with other libraries.
  + **Pandas**:
    - Primarily for data manipulation but offers plotting capabilities.
    - Built on Matplotlib, making it easy to create plots directly from data frames.
  + **Seaborn**:
    - Built on Matplotlib, focused on statistical visualizations.
    - Offers stylish plots and integrates well with Pandas.
  + **Folium**:
    - Excellent for geospatial data visualization.
    - Allows creation of interactive maps (choropleth, point maps, heat maps).
  + **Plotly**:
    - Provides highly interactive plots and dashboards.
    - Web-based, making it easy to share and collaborate on visualizations.
  + **PyWaffle**:
    - Simple library for creating waffle charts and other categorical visualizations.

These libraries each have unique strengths and can be combined to create sophisticated visualizations.

## **Introduction to Matplotlib**

* **What is Matplotlib?**
  + A widely used data visualization library in Python.
  + Created by John Hunter to replace proprietary software for EEG and ECoG visualization.
* **Architecture of Matplotlib:**
  + Composed of three main layers:
    - **Backend Layer:** Handles rendering and user input.
    - **Artist Layer:** Contains objects (artists) that represent visual elements (e.g., lines, text).
    - **Scripting Layer (Pyplot):** Simplifies common tasks for quick graphics generation.
* **Types of Artist Objects:**
  + **Primitive Artists:** Basic shapes like Line2D, Rectangle, Circle, and Text.
  + **Composite Artists:** More complex structures like Axis, Tick, Axes, and Figure.
* **Anatomy of a Plot:**
  + Components include:
    - **Canvas:** The area where the plot is drawn.
    - **Axis:** Represents individual plots within a figure.
    - **Title, Labels, Legend, Grid, Annotations:** Enhance understanding of the data.
* **Example of Creating a Histogram:**
  + Use the Pyplot interface to generate a histogram of random numbers.
* **Resources:**
  + Reference to the official Matplotlib website for further reading on the anatomy of figures.

## **Basic Plotting with Matplotlib**

* **Matplotlib** is a well-established data visualization library in Python.
* It can be integrated into various environments, including Jupyter Notebooks.
* The **Plot function** is central for creating various types of visualizations.
* **Backends** in Matplotlib determine how plots are displayed:
  + **Inline Backend**: Displays plots within the browser.
  + **Notebook Backend**: Allows modifications to the plot after rendering.

Code Example:

import matplotlib.pyplot as plt

# Create a simple plot

plt.plot(5, 5, 'o') # 'o' indicates a circular marker at position (5, 5)

plt.title("Simple Plot") # Adding a title

plt.xlabel("X-axis") # Label for the x-axis

plt.ylabel("Y-axis") # Label for the y-axis

plt.show() # Display the plot

Explanation:

* **Importing Matplotlib**: The code starts by importing the matplotlib.pyplot module as plt, which is a common convention.
* **Creating a Plot**: The plt.plot(5, 5, 'o') function creates a plot with a circular marker at the coordinates (5, 5).
* **Adding Titles and Labels**: The plt.title(), plt.xlabel(), and plt.ylabel() functions are used to add a title and labels to the axes.
* **Displaying the Plot**: Finally, plt.show() renders the plot in the output.

## **Dataset on Immigration to Canada**

* **Topic**: Dataset on Immigration to Canada
* **Objective**: Understand the dataset for data visualization and learn how to import and process data using pandas.

Key Points

1. **Data Source**:
   * The dataset is compiled by the **Population Division of the United Nations**.
   * It includes immigration data from **45 countries**.
2. **Data Structure**:
   * The dataset contains:
     + **Textual data** about the UN department and other information (first 20 rows).
     + **Column labels** (Row 21).
     + **Country-specific data** including:
       - Country name
       - Continent
       - Region (developed or developing)
       - Total number of immigrants from that country (years 1980 to 2013).
3. **Focus**:
   * The course will primarily focus on immigration data to **Canada**.
4. **Data Importing**:
   * To analyze the data, it needs to be imported into a **pandas DataFrame**.
   * Required libraries:
     + **pandas**: For data manipulation.
     + **openpyxl**: To extract data from Excel files.
   * Use the function read\_excel to read the data, skipping the first 20 rows to focus on relevant data.
5. **Data Processing**:
   * After importing, the country name should be set as the **index** of each row for easier querying.
   * An additional column should be added to represent the **total immigration** for each country from 1980 to 2013.
6. **DataFrame Naming**:
   * The DataFrame is named df\_canada.

Conclusion

* The lecture emphasizes the importance of understanding the dataset and preparing it for visualization.
* Throughout the course, pandas will be used for data analysis before creating various visualization tools.

## **Line Plots**

Line Plots Overview

* **Definition**: A line plot displays information as a series of data points connected by straight lines. It is a basic type of chart commonly used in various fields, including data science.

When to Use a Line Plot

* **Visualizing Trends**: Line plots are particularly useful for visualizing trends and changes over time.
* **Time-Series Data**: Commonly used for time-series data, such as:
  + Changes in stock prices
  + Website traffic
  + Temperature fluctuations
* **Comparing Variables**: They can also show relationships between two variables and compare multiple data series on one chart.
* **Highlighting Changes**: Effective in highlighting sudden changes or anomalies in data.

Example

* **Immigration from Haiti to Canada**:
  + A line plot can illustrate the trend of immigrants from Haiti to Canada from 1980 to 2013.
  + Notable spike in 2010 due to the earthquake in Haiti.

Generating a Line Plot

1. **Data Preparation**:
   * Each row in the dataset represents a country with annual immigration data to Canada from 1980-2013.
   * The country name is set as the index for easier querying.
   * An additional column for total immigration from each country is added.
2. **Using Matplotlib**:
   * Import the necessary libraries:
   * import matplotlib.pyplot as plt

import pandas as pd

* + **Load the Data**: Assuming you have a DataFrame named df\_canada containing the immigration data.
  + # Example DataFrame creation (replace with actual data loading)
  + data = {
  + 'Country': ['Haiti', 'Afghanistan', 'Albania'],
  + '1980': [1000, 500, 200],
  + '2010': [5000, 2000, 1000],
  + '2013': [6000, 2500, 1500]
  + }
  + df\_canada = pd.DataFrame(data)

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

* + **Generate the Line Plot**:
  + # Plotting the immigration data for Haiti
  + plt.plot(df\_canada.loc['Haiti'], kind='line', marker='o')
  + plt.title('Immigration from Haiti to Canada (1980-2013)')
  + plt.xlabel('Year')
  + plt.ylabel('Number of Immigrants')
  + plt.grid()

plt.show()

Code Explanation

* **Importing Libraries**:
  + import matplotlib.pyplot as plt: Imports the Matplotlib library for plotting.
  + import pandas as pd: Imports the Pandas library for data manipulation.
* **Loading Data**:
  + A sample DataFrame df\_canada is created with immigration data. In practice, you would load your dataset from a file or other source.
* **Setting Index**:
  + df\_canada.set\_index('Country', inplace=True): Sets the 'Country' column as the index for easier data access.
* **Plotting**:
  + plt.plot(df\_canada.loc['Haiti'], kind='line', marker='o'): Plots the data for Haiti. The marker='o' adds circular markers at each data point.
  + plt.title(...): Sets the title of the plot.
  + plt.xlabel(...): Labels the x-axis.
  + plt.ylabel(...): Labels the y-axis.
  + plt.grid(): Adds a grid to the plot for better readability.
  + plt.show(): Displays the plot.

Key Takeaways

* A line plot is a simple yet powerful tool for visualizing data trends.
* It can be generated easily using Python libraries like Matplotlib.

Summary: Introduction to Data Visualization Tools

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

* Data visualization is the process of presenting data in a visual format, such as charts, graphs, and maps, to help people understand and analyze data easily.
* Data visualization has diverse use cases, such as in business, science, healthcare, and finance.
* It is important to follow best practices, such as selecting appropriate visualizations for the data being presented, choosing colors and fonts that are easy to read and interpret, and minimizing clutter.
* There are various types of plots commonly used in data visualization.
* Line plots capture trends and changes over time, allowing us to see patterns and fluctuations.
* Bar plots compare categories or groups, providing a visual comparison of their values.
* Scatter plots explore relationships between variables, helping us identify correlations or trends.
* Box plots display the distribution of data, showcasing the median, quartiles, and outliers.
* Histograms illustrate the distribution of data within specific intervals, allowing us to understand its shape and concentration.
* Matplotlib is a plotting library that offers a wide range of plotting capabilities.
* Pandas is a plotting library that provides Integrated plotting functionalities for data analysis.
* Seaborn is a specialized library for statistical visualizations, offering attractive default aesthetics and color palettes.
* Folium is a Python library that allows you to create interactive and customizable maps.
* Plotly is an interactive and dynamic library for data visualization that supports a wide range of plot types and interactive features.
* PyWaffle enables you to visualize proportional representation using squares or rectangles.
* Matplotlib is one of the most widely used data visualization libraries in Python.
* Matplotlib was initially developed as an EEG/ECoG visualization tool.
* Matplotlib’s architecture is composed of three main layers: Backend layer, Artist layer, and the Scripting layer.
* The anatomy of a plot refers to the different components and elements that make up a visual representation of data.
* Matplotlib is a well-established data visualization library that can be integrated in different environments.
* Jupyter Notebook is an open-source web application that allows you to create and share documents.
* Matplotlib has a number of different backends available.
* You can easily include the label and title to your plot with plt.
* In order to start creating different types of plots of the data, you will need to import the data into a Pandas DataFrame.
* A line plot is a plot in the form of a series of data points connected by straight line segments.
* Line plot is one of the most basic type of chart and is common in many fields.
* You can generate a line plot by assigning "line" to 'Kind' parameter in the plot() function.