Visualization and Analysis of International Football Results (1872-2024)

This repository contains code and analytical documents focused on the visualization and analysis of international football results spanning from 1872 to 2024, using a dataset sourced from Kaggle.

Overview

For the purpose of data visualization, two prominent Python libraries—Matplotlib and Plotly—have been utilized. Most visualizations in this repository are created using Matplotlib, while some are developed using Plotly to leverage its interactive capabilities.

Introduction to Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is widely used for plotting data and generating a broad range of chart types, from simple line plots to complex heatmaps and 3D plots.

Key Features

- **Versatility**: Matplotlib supports a variety of plots, including line charts, bar charts, scatter plots, histograms, and more.
- **Customization**: Users have control over every aspect of a plot, such as line styles, font properties, and axes properties.
- **Interactivity**: Offers features like zooming, panning, and dynamic updates.
- **Integration**: Works seamlessly with other Python libraries like NumPy, pandas, and SciPy.

Basic Usage

To use Matplotlib, you typically start by importing the matplotlib.pyplot module:

```
import matplotlib.pyplot as plt
```

Example: Creating a Simple Line Plot

```
import matplotlib.pyplot as plt


# Data

years = [2000, 2005, 2010, 2015, 2020]

population = [6.1, 6.5, 6.9, 7.3, 7.8]


# Create a line plot

plt.plot(years, population)
```

```
# Add title and labels
plt.title("World Population Over Time")
plt.xlabel("Year")
plt.ylabel("Population (billions)")

# Show the plot
plt.show()
```

This snippet creates a simple line plot depicting world population growth over time.

Conclusion

Matplotlib is a powerful tool for data visualization, offering extensive customization and integration capabilities. It is widely used by scientists, engineers, and data analysts.

Introduction to Plotly

Plotly is a versatile library for creating interactive, web-based visualizations in Python. It is particularly noted for its ability to produce dynamic and complex visualizations effortlessly.

Key Features

- **Interactivity**: Plotly excels in creating interactive visualizations that support features like hovering, zooming, and panning.
- **Versatility**: Supports a broad range of chart types, including line plots, bar charts, scatter plots, heatmaps, and 3D charts.
- **Integration**: Works well with popular data analysis libraries like pandas and NumPy, and integrates seamlessly with Jupyter Notebooks.
- **Customization**: Offers extensive options for customizing plots, including layout, colors, and annotations.
- **Accessibility**: Visualizations are rendered in web browsers, making them easily shareable and accessible on various platforms.

Basic Usage

To use Plotly, you typically start by importing the plotly.express module:

```
import plotly.express as px
```

Example: Creating a Simple Line Plot

```
import plotly.express as px
```

```
# Data
years = [2000, 2005, 2010, 2015, 2020]
population = [6.1, 6.5, 6.9, 7.3, 7.8]

# Create a line plot
fig = px.line(x=years, y=population, title="World Population Over Time")
fig.update_layout(xaxis_title="Year", yaxis_title="Population (billions)")

# Show the plot
fig.show()
```

This snippet creates an interactive line plot illustrating world population growth over time.

Conclusion

Plotly is an excellent choice for creating interactive and visually appealing data visualizations. Its ease of use, combined with powerful features, makes it a favorite among data scientists, analysts, and developers.

Introduction to pandas

Pandas is a powerful and widely-used open-source data manipulation and analysis library for Python. It provides data structures and functions needed to work with structured data seamlessly.

Key Features

- **Data Structures**: Offers Series (one-dimensional) and DataFrame (two-dimensional) structures for intuitive and flexible data manipulation.
- **Data Manipulation**: Provides a variety of functions for data cleaning, transformation, and analysis, including filtering, grouping, merging, and pivoting tables.
- **Handling Missing Data**: Built-in functionality to handle missing or null data effectively.
- **Integration**: Integrates well with other libraries like NumPy and Matplotlib.
- **Input/Output Operations**: Supports reading and writing data from various file formats such as CSV, Excel, SQL databases, and more.

Basic Usage

To use pandas, you typically start by importing the library:

```
import pandas as pd
```

Example: Creating a Simple DataFrame

```
import pandas as pd


# Data
data = {
    'Year': [2000, 2005, 2010, 2015, 2020],
    'Population (billions)': [6.1, 6.5, 6.9, 7.3, 7.8]
}


# Create a DataFrame
df = pd.DataFrame(data)


# Display the DataFrame
print(df)
```

This snippet creates a simple DataFrame displaying world population data over time.

Conclusion

Pandas is an essential library for data manipulation and analysis in Python. Its ability to handle large amounts of data efficiently, combined with its versatile functions, makes it a fundamental tool for data scientists, analysts, and developers.

Repository Structure

All folder names in this repository are self-explanatory and indicate their content. Each code file is accompanied by documentation and an analysis report to facilitate understanding.

Happy Coding!