Visualize your Data

Data Visualization Libraries

- MatPLotLib - General-purpose plotting
- Seaborn - Statistical graphics and beautiful themes
- Plotly - Interactive and web-based plots
- Geopandas - Geospatial data visualization

In [ ]:
```
#load pandas

import pandas as pd
```

MatPlotLib

In [ ]:
```
# import the library

import matplotlib.pyplot as plt

# load dataframe

scotus = pd.read_csv("scotus.csv")

# set datatime

scotus["date"] = pd.to_datetime(scotus["date"])
scotus
```

In [ ]:
```
# filter pollster to YouGov

scotus = scotus[scotus["pollster"] == "YouGov"]
scotus
```

In [ ]:
```
# create a line graph of scotus approval rating

#use plot to call the line plot shape

scotus.plot()
```

In [ ]:
```
# set date and yes appoval

scotus.plot(x="date", y="yes")


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

In [ ]:
```
# stylize the graph

scotus.plot(x="date", y="yes",
```

```
            color = "red",
            linewidth=.75,
             )
```

In [ ]:
```
# add title and removove y label and legend label

scotus.plot(x="date", y="yes",
                  color = "red",
                  linewidth=.75,
            xlabel="date",
            ylabel="",
            title="Scotus Approval 2024",
            legend = ''
                  )
```

Adjust Font Size

In [ ]:
```
import matplotlib.pyplot as plt
import matplotlib.dates as mdates

# Sample plot command with customizations
scotus.plot(x="date", y="yes",
            color="red",
            linewidth=0.75,
            xlabel="Date",
            ylabel="",
            title="Scotus Approval 2024",
            legend='')

# Adjust font sizes
plt.xlabel("Date", fontsize=10)
plt.title("Scotus Approval 2024", fontsize=24, fontweight="bold")

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

Adjust Date Format and Breaks

Date Formats

In [ ]:
```
import matplotlib.pyplot as plt
import matplotlib.dates as mdates

# Sample plot command with customizations
scotus.plot(x="date", y="yes",
            color="red",
            linewidth=0.75,
            xlabel="Date",
            ylabel="",
            title="Scotus Approval 2024",
            legend='')

# Adjust font sizes
plt.xlabel("Date", fontsize=10)
plt.title("Scotus Approval 2024", fontsize=24, fontweight="bold")
```

```
# Customize date format on x-axis
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%b-%y'))  # Adjust date format as needed
plt.gca().xaxis.set_major_locator(mdates.MonthLocator(interval=2))  # Every other month

# Optional: Rotate date labels for better readability
plt.gcf().autofmt_xdate()

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

Add Addtional Variable

In [ ]:
```
import matplotlib.pyplot as plt
import matplotlib.dates as mdates

# Create a figure and axis
fig, ax = plt.subplots()

# Plot the 'yes' variable
scotus.plot(x="date", y="yes",
            color="red",
            linewidth=0.75,
            ax=ax,  # Specify the axis to plot on
            label="Yes")

# Plot the 'no' variable
scotus.plot(x="date", y="no",
            color="blue",
            linewidth=0.75,
            ax=ax,  # Specify the same axis to plot on
            label="No")

# Customize the labels and title
plt.xlabel("Date", fontsize=10)
ax.set_title("Scotus Approval 2024", fontsize=24, fontweight="bold")

# Customize date format on x-axis
ax.xaxis.set_major_formatter(mdates.DateFormatter('%b-%y'))  # Adjust date format as needed
ax.xaxis.set_major_locator(mdates.MonthLocator(interval=2))  # Every other month

# Optional: Rotate date labels for better readability
fig.autofmt_xdate()

# Add legend
ax.legend()

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

Export Plot

In [ ]:
```
import matplotlib.pyplot as plt
import matplotlib.dates as mdates

# Create a figure and axis
```

```
fig, ax = plt.subplots()

# Plot the 'yes' variable
scotus.plot(x="date", y="yes",
            color="red",
            linewidth=0.75,
            ax=ax,  # Specify the axis to plot on
            label="Yes")

# Plot the 'no' variable
scotus.plot(x="date", y="no",
            color="blue",
            linewidth=0.75,
            ax=ax,  # Specify the same axis to plot on
            label="No")

# Customize the labels and title
plt.xlabel("Date", fontsize=10)
ax.set_title("Scotus Approval 2024", fontsize=24, fontweight="bold")

# Customize date format on x-axis
ax.xaxis.set_major_formatter(mdates.DateFormatter('%b-%y'))  # Adjust date format as needed
ax.xaxis.set_major_locator(mdates.MonthLocator(interval=2))  # Every other month

# Optional: Rotate date labels for better readability
fig.autofmt_xdate()

# Add legend
ax.legend()

# Ensure all elements fit in the figure
plt.tight_layout()

# exporting
plt.savefig("scotus_line.png")

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

Seaborn

https://seaborn.pydata.org/

In [ ]:
```
# load the libraries

import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
# load the dataframe

import pandas as pd

reviews = pd.read_csv("cleaned_reviews.csv")
reviews.dtypes
```

In [ ]:
```
# create the bar plot

sns.countplot(data = reviews, x = "Department_Name")
```

In [ ]:
```
# create the bar plot

sns.countplot(data = reviews, x = "Department_Name")

# Add title and labels
plt.title('Count of Department Purchases')
plt.xlabel('Department')
plt.ylabel('')

# Show plot
plt.show()
```

In [ ]:
```
# Set Seaborn style
sns.set_style("darkgrid")

sns.countplot(data = reviews, x = "Department_Name")

# Add title and labels
plt.title('Count of Department Purchases')
plt.xlabel('Department')
plt.ylabel('')

# Show plot
plt.show()
```

Color Brewer Palettes

https://colorbrewer2.org/

In [ ]:
```
# Set Seaborn style
sns.set_style("darkgrid")

# Define Color Brewer palette
brewer_palette = sns.color_palette("YlOrRd")

# Create countplot with Color Brewer palette
sns.countplot(data=reviews, x="Department_Name", palette=brewer_palette)

# Add title and labels
plt.title('Count of Department Purchases')
plt.xlabel('Department')
plt.ylabel('')

# Show plot
plt.show()
```

Plotly

https://plotly.com/python/

In [ ]:
```python
import plotly.graph_objects as go

# Assuming reviews is a DataFrame containing data
# Create a Plotly histogram figure
fig = go.Figure(data=[go.Histogram(x=reviews["Age"])])

# Display the histogram
fig.show()
```

Add Bins

In [ ]:
```python
import plotly.graph_objects as go

# Create a Plotly histogram figure with additional options
fig = go.Figure(data=[go.Histogram(x=reviews["Age"],
                                   # Set number of bins
                                   nbinsx=20,
                                   )])

# Update layout for better appearance
fig.update_layout(title="Histogram of Age",
                  xaxis_title="Age",
                  yaxis_title="Frequency",

                 )

# Display the histogram
fig.show()
```

Change Theme

In [ ]:
```
import plotly.graph_objects as go

# Assuming reviews is a DataFrame containing data

# Create a Plotly histogram figure with additional options
fig = go.Figure(data=[go.Histogram(x=reviews["Age"],
                                   # Set number of bins
                                   nbinsx=20,
                                   opacity=0.7,
                                   # Set fill and line colors
                                   marker=dict(color='#ffbf00',  # Fill color
                                               line=dict(color='#f08080', width=3)))])

# Update layout for better appearance
fig.update_layout(title="Histogram of Age",
                  xaxis_title="Age",
                  yaxis_title="Frequency",
                  bargap=0.1,  # Set gap between bars
                  plot_bgcolor='#f0f0f0',  # Set background color
                  paper_bgcolor='#ffffff'  # Set paper color
                 )

# Display the histogram
fig.show()
```

Change Size

In [ ]:
```
import plotly.graph_objects as go

# Assuming reviews is a DataFrame containing data

# Create a Plotly histogram figure with additional options
fig = go.Figure(data=[go.Histogram(x=reviews["Age"],
                                   # Set number of bins
                                   nbinsx=20,
                                   opacity=0.7,
                                   # Set fill and line colors
                                   marker=dict(color='#ffbf00',  # Fill color
                                               line=dict(color='#f08080', width=3)))])

# Update layout for better appearance
fig.update_layout(title="Histogram of Age",
                  xaxis_title="Age",
                  yaxis_title="Frequency",
                  bargap=0.1,  # Set gap between bars
                  plot_bgcolor='#f0f0f0',  # Set background color
                  paper_bgcolor='#ffffff',  # Set paper color
                  height=750  # Set graph height
                 )

# Display the histogram
fig.show()
```