Plot Suggestions for Different Variable Types

Continuous and Categorical:

1. Boxplot: A boxplot can show the distribution of the continuous variable for each category of the categorical variable, allowing you to compare central tendency, spread, and identify potential outliers.
2. Violin Plot: Similar to a boxplot but with a rotated kernel density plot on each side, violin plots provide a richer visualization of the distribution of the continuous variable across categories.
3. Bar Plot with Error Bars: This plot can display the mean or median of the continuous variable for each category along with error bars to show variability, providing a clear comparison between categories.

Continuous and Continuous:

1. Scatter Plot: A classic choice for visualizing the relationship between two continuous variables, showing patterns, clusters, trends, or correlations between the variables.
2. Line Plot: Useful for displaying trends over time or sequential data, a line plot can show how one continuous variable changes in relation to another continuous variable.
3. Correlation Matrix: If you have multiple pairs of continuous variables to compare, a correlation matrix can provide a comprehensive overview of the relationships between all pairs, highlighting strong and weak correlations.

Categorical and Categorical:

1. Stacked Bar Chart: This plot can show the distribution of one categorical variable within each category of the other categorical variable, allowing for easy comparison of proportions.
2. Heatmap: A heatmap can display the frequency or proportion of each combination of categories, providing a visual representation of the relationship between the two categorical variables.
3. Clustered Bar Chart: By grouping bars based on one categorical variable and coloring them based on the other, a clustered bar chart can help compare the distribution of categories across the two variables.

Histogram, grouped bar, stacked bar, donut chart, pie chart, bar, count plot, 3D scatter plot, mosaic bar

import plotly.express as px

# Sample data

data = px.data.tips()

# Box Plot

fig = px.box(data\_frame=data, x="day", y="total\_bill", color="smoker", notched=True,

             points="all", boxmode="group", title="Box Plot")

fig.show()

# Violin Plot

fig = px.violin(data\_frame=data, x="day", y="total\_bill", color="smoker", box=True,

                points="all", title="Violin Plot")

fig.show()

# Bar Plot with Error Bars

fig = px.bar(data\_frame=data, x="day", y="total\_bill", color="sex", error\_x="stddev",

             error\_y="stddev", title="Bar Plot with Error Bars")

fig.show()

# Scatter Plot

fig = px.scatter(data\_frame=data, x="total\_bill", y="tip", color="sex", trendline="ols",

                 title="Scatter Plot")

fig.show()

# Line Plot

fig = px.line(data\_frame=data, x="total\_bill", y="tip", color="sex", title="Line Plot")

fig.show()

# Correlation Matrix

correlation\_matrix = data.corr()

fig = px.imshow(correlation\_matrix, color\_continuous\_scale='Viridis',

                title="Correlation Matrix")

fig.show()

# Stacked Bar Chart

df = px.data.tips()

fig = px.bar(data\_frame=df, x="day", y="total\_bill", color="sex", barmode="stack",

             title="Stacked Bar Chart")

fig.show()

# Heatmap

fig = px.imshow(correlation\_matrix, color\_continuous\_scale='Viridis',

                title="Heatmap")

fig.show()

# Clustered Bar Chart

df = px.data.tips()

fig = px.bar(data\_frame=df, x="day", y="total\_bill", color="sex", barmode="group",

             title="Clustered Bar Chart")

fig.show()

# Histogram

fig = px.histogram(data\_frame=data, x="total\_bill", color="sex", marginal="rug",

                   title="Histogram")

fig.show()

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

# Sample data

np.random.seed(0)

data = np.random.randn(100)

# Box Plot

plt.figure(figsize=(6, 4))

plt.boxplot(data, notch=True, vert=False, patch\_artist=True)

plt.title("Box Plot")

plt.show()

# Violin Plot

plt.figure(figsize=(6, 4))

sns.violinplot(data=data, inner="point")

plt.title("Violin Plot")

plt.show()

# Bar Plot with Error Bars

x = np.arange(5)

y = np.random.rand(5)

errors = np.random.rand(5) / 2

plt.bar(x, y, yerr=errors, capsize=5)

plt.title("Bar Plot with Error Bars")

plt.show()

# Scatter Plot

x = np.random.randn(100)

y = np.random.randn(100)

plt.scatter(x, y, alpha=0.5)

plt.title("Scatter Plot")

plt.show()

# Line Plot

x = np.linspace(0, 10, 100)

y = np.sin(x)

plt.plot(x, y)

plt.title("Line Plot")

plt.show()

# Correlation Matrix

data = np.random.rand(10, 10)

corr = np.corrcoef(data)

plt.imshow(corr, cmap='viridis', interpolation='nearest')

plt.colorbar()

plt.title("Correlation Matrix")

plt.show()

# Stacked Bar Chart

N = 5

menMeans = (20, 35, 30, 35, 27)

womenMeans = (25, 32, 34, 20, 25)

ind = np.arange(N)

width = 0.35

plt.bar(ind, menMeans, width, label='Men')

plt.bar(ind, womenMeans, width, bottom=menMeans, label='Women')

plt.ylabel('Scores')

plt.title('Scores by group and gender')

plt.legend()

plt.show()

# Heatmap

data = np.random.rand(10, 12)

plt.imshow(data, cmap='hot', interpolation='nearest')

plt.colorbar()

plt.title("Heatmap")

plt.show()

# Clustered Bar Chart

labels = ['G1', 'G2', 'G3', 'G4', 'G5']

men\_means = [20, 35, 30, 35, 27]

women\_means = [25, 32, 34, 20, 25]

x = np.arange(len(labels))

width = 0.35

fig, ax = plt.subplots()

rects1 = ax.bar(x - width/2, men\_means, width, label='Men')

rects2 = ax.bar(x + width/2, women\_means, width, label='Women')

ax.set\_ylabel('Scores')

ax.set\_title('Scores by group and gender')

ax.set\_xticks(x)

ax.set\_xticklabels(labels)

ax.legend()

plt.show()

# Histogram

data = np.random.randn(1000)

plt.hist(data, bins=30, edgecolor='black')

plt.title("Histogram")

plt.show()

import seaborn as sns

import matplotlib.pyplot as plt

import numpy as np

# Sample data

np.random.seed(0)

data = np.random.randn(100)

# Box Plot

sns.boxplot(data=data, notch=True, orient='v')

plt.title("Box Plot")

plt.show()

# Violin Plot

sns.violinplot(data=data, inner="point")

plt.title("Violin Plot")

plt.show()

# Bar Plot with Error Bars

x = np.arange(5)

y = np.random.rand(5)

errors = np.random.rand(5) / 2

sns.barplot(x=x, y=y, yerr=errors, capsize=0.1)

plt.title("Bar Plot with Error Bars")

plt.show()

# Scatter Plot

x = np.random.randn(100)

y = np.random.randn(100)

sns.scatterplot(x=x, y=y)

plt.title("Scatter Plot")

plt.show()

# Line Plot

x = np.linspace(0, 10, 100)

y = np.sin(x)

sns.lineplot(x=x, y=y)

plt.title("Line Plot")

plt.show()

# Correlation Matrix

data = np.random.rand(10, 10)

corr = np.corrcoef(data)

sns.heatmap(data=corr, cmap='viridis')

plt.title("Correlation Matrix")

plt.show()

# Stacked Bar Chart

N = 5

menMeans = (20, 35, 30, 35, 27)

womenMeans = (25, 32, 34, 20, 25)

ind = np.arange(N)

width = 0.35

plt.bar(ind, menMeans, width, label='Men')

plt.bar(ind, womenMeans, width, bottom=menMeans, label='Women')

plt.ylabel('Scores')

plt.title('Scores by group and gender')

plt.legend()

plt.show()

# Heatmap

data = np.random.rand(10, 12)

sns.heatmap(data, cmap='hot')

plt.title("Heatmap")

plt.show()

# Clustered Bar Chart

labels = ['G1', 'G2', 'G3', 'G4', 'G5']

men\_means = [20, 35, 30, 35, 27]

women\_means = [25, 32, 34, 20, 25]

x = np.arange(len(labels))

width = 0.35

fig, ax = plt.subplots()

rects1 = ax.bar(x - width/2, men\_means, width, label='Men')

rects2 = ax.bar(x + width/2, women\_means, width, label='Women')

ax.set\_ylabel('Scores')

ax.set\_title('Scores by group and gender')

ax.set\_xticks(x)

ax.set\_xticklabels(labels)

ax.legend()

plt.show()

# Histogram

data = np.random.randn(1000)

sns.histplot(data, bins=30, kde=True)

plt.title("Histogram")

plt.show()

import matplotlib.pyplot as plt

import seaborn as sns

import plotly.express as px

# Sample data

x\_data = [1, 2, 3, 4, 5]

y\_data = [2, 3, 5, 7, 11]

# Load sample dataset

tips = sns.load\_dataset("tips")

# Sample data

x = [1, 2, 3, 4, 5]

y = [2, 3, 5, 7, 11]

# Plotting

plt.figure(figsize=(8, 6)) # Optional: Adjust figure size

# Box plot

plt.boxplot(data, vert=False, patch\_artist=True, showmeans=True)

sns.boxplot(x="day", y="total\_bill", data=tips)

# Violin plot

sns.violinplot(x="day", y="total\_bill", data=tips)

# Scatter plot

plt.scatter(x\_data, y\_data, color='red', label='Scatter Plot')

sns.scatterplot(x="total\_bill", y="tip", data=tips)

scatter\_trace = go.Scatter(x=x, y=y, mode='markers', name='Scatter Plot')

scatter\_layout = go.Layout(title='Sample Scatter Plot', xaxis=dict(title='X-axis'), yaxis=dict(title='Y-axis'))

scatter\_figure = go.Figure(data=[scatter\_trace], layout=scatter\_layout)

scatter\_figure.show()

# Plotly Express

scatter\_express = px.scatter(x=x, y=y, title='Sample Scatter Plot (Express)', labels={'x': 'X-axis', 'y': 'Y-axis'})

scatter\_express.show()

# Line plot

plt.plot(x\_data, y\_data, marker='o', linestyle='-', label='Line Plot')

sns.lineplot(x="total\_bill", y="tip", data=tips)

line\_trace = go.Scatter(x=x, y=y, mode='lines', name='Line Plot')

line\_layout = go.Layout(title='Sample Line Plot', xaxis=dict(title='X-axis'), yaxis=dict(title='Y-axis'))

line\_figure = go.Figure(data=[line\_trace], layout=line\_layout)

line\_figure.show()

# Heatmap

data = np.random.rand(10, 10)

plt.subplot(2, 2, 4)

plt.imshow(data, cmap='viridis', interpolation='nearest')

plt.colorbar(label='Color Scale')

correlation\_matrix = tips.corr()

sns.heatmap(correlation\_matrix, annot=True, cmap="coolwarm")

z = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

heatmap\_trace = go.Heatmap(z=z)

heatmap\_layout = go.Layout(title='Sample Heatmap')

heatmap\_figure = go.Figure(data=[heatmap\_trace], layout=heatmap\_layout)

heatmap\_figure.show()

# Histogram

data = [1, 1, 2, 2, 2, 3, 3, 3, 3, 4, 4, 5, 5, 5, 5]

plt.hist(data, bins=5, color='blue', alpha=0.7, label='Histogram')

sns.histplot(tips["total\_bill"], bins=10, kde=True)

# Customize plot

plt.title('Sample Plots')

plt.xlabel('X-axis Label')

plt.ylabel('Y-axis Label')

plt.legend()

plt.suptitle('Additional Sample Plots') # Optional: Add main title

plt.tight\_layout() # Optional: Adjust layout

# Show plot

plt.grid(True) # Optional: Add grid

plt.show()

# Bar plot

plt.bar(x\_data, y\_data, color='green', label='Bar Plot')

sns.barplot(x="day", y="total\_bill", data=tips)

bar\_trace = go.Bar(x=x, y=y, name='Bar Plot')

bar\_layout = go.Layout(title='Sample Bar Plot', xaxis=dict(title='X-axis'), yaxis=dict(title='Y-axis'))

bar\_figure = go.Figure(data=[bar\_trace], layout=bar\_layout)

bar\_figure.show()

# Pie chart

labels = ['A', 'B', 'C', 'D']

sizes = [15, 30, 45, 10]

plt.subplot(2, 2, 3)

plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)

sns.set\_style("whitegrid") # Optional: Set style

sns.pieplot(sizes, labels=labels, colors=sns.color\_palette("pastel"), startangle=90, autopct='%1.1f%%')

labels = ['A', 'B', 'C', 'D']

values = [15, 30, 45, 10]

pie\_trace = go.Pie(labels=labels, values=values)

pie\_layout = go.Layout(title='Sample Pie Chart')

pie\_figure = go.Figure(data=[pie\_trace], layout=pie\_layout)

pie\_figure.show()

# Pairplot

sns.pairplot(tips)