

```

import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np

# Load dataset
file = "/content/Visualization Dataset.xlsx"
df = pd.read_excel(file, sheet_name="Sheet1")

# Data Cleaning
df_cleaned = df.iloc[3:].reset_index(drop=True)
df_cleaned.columns = [
    "Index", "Transportation Services", "Fiscal Year", "Revenues",
    "Expenditures", "Net Revenue", "YoY Variance $", "YoY Variance %", "Customer Satisfaction"]

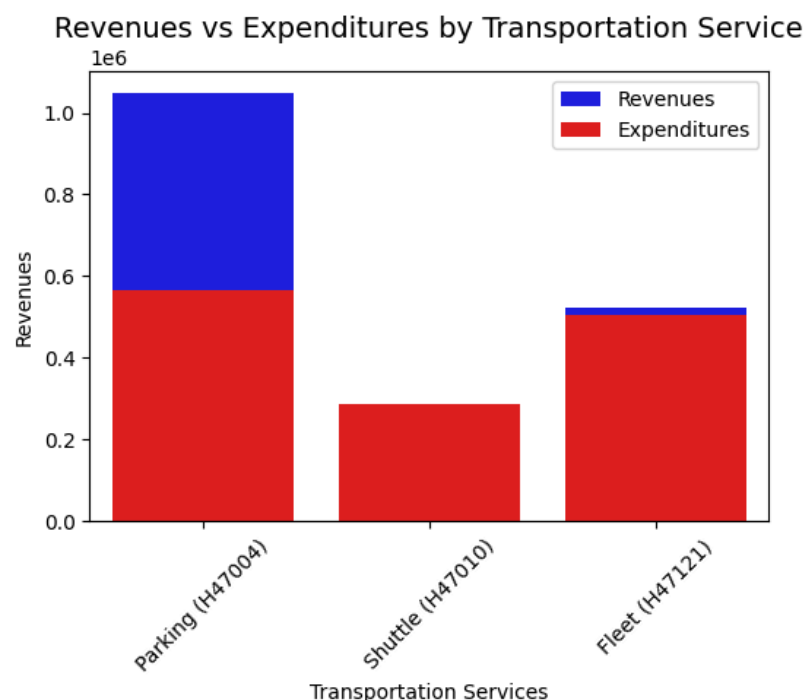
df_cleaned = df_cleaned.drop(columns=["Index"]).dropna(subset=["Transportation Services"]).reset_index(drop=True)
df_cleaned = df_cleaned.iloc[1:].reset_index(drop=True)

# Convert numerical columns to appropriate data types
numeric_columns = ["Revenues", "Expenditures", "Net Revenue", "YoY Variance $", "YoY Variance %", "Customer Satisfaction"]
for col in numeric_columns:
    df_cleaned[col] = pd.to_numeric(df_cleaned[col], errors="coerce")

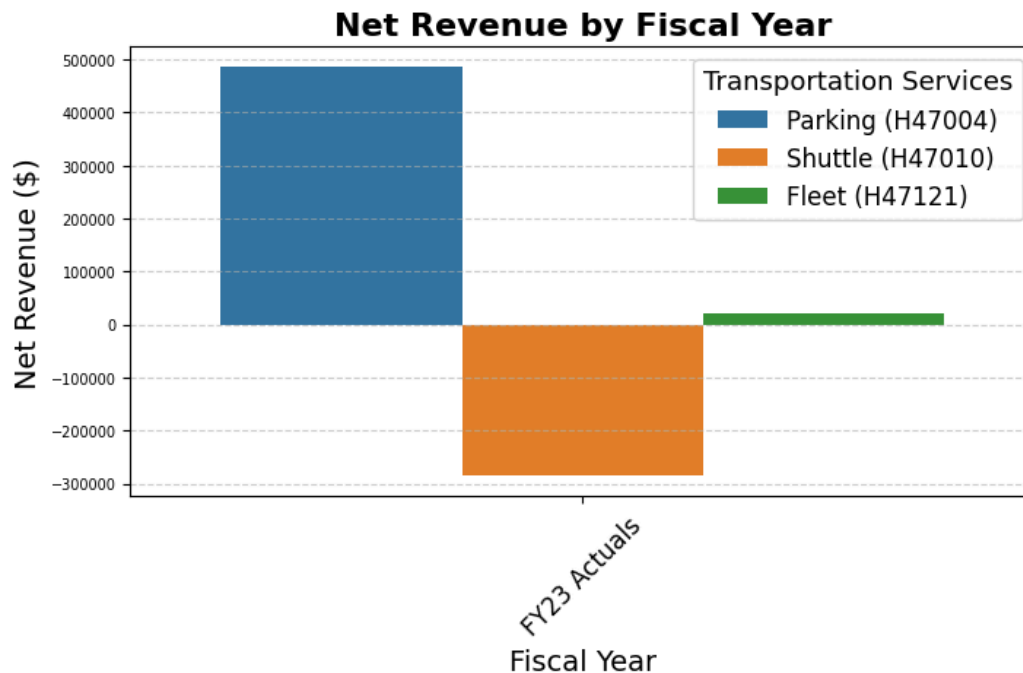
# Filter out non-data rows
df_viz = df_cleaned.dropna(subset=["Revenues", "Expenditures", "Net Revenue", "Customer Satisfaction"])

### Plot 1: Bar plot for Revenues and Expenditures ###
plt.figure(figsize=(6, 4))
sns.barplot(
    data=df_viz, x="Transportation Services", y="Revenues", color="blue", label="Revenues"
)
sns.barplot(
    data=df_viz, x="Transportation Services", y="Expenditures", color="red", label="Expenditures"
)
plt.title("Revenues vs Expenditures by Transportation Service", fontsize=14)
plt.legend()
plt.xticks(rotation=45)
plt.show()

```



```
plt.figure(figsize=(8, 4))
sns.barplot(
    data=df_viz, x="Fiscal Year", y="Net Revenue", hue="Transportation Services",
    palette="tab10"
)
plt.title("Net Revenue by Fiscal Year", fontsize=16, fontweight='bold')
plt.xlabel("Fiscal Year", fontsize=14)
plt.ylabel("Net Revenue ($)", fontsize=14)
plt.xticks(rotation=45, fontsize=12)
plt.yticks(fontsize=7)
plt.legend(title="Transportation Services", fontsize=12, title_fontsize=13)
plt.grid(axis="y", linestyle="--", alpha=0.6)
plt.show()
```



```
!pip install dash
```



```
Collecting dash
  Downloading dash-2.18.2-py3-none-any.whl.metadata (10 kB)
Collecting Flask<3.1,>=1.0.4 (from dash)
  Downloading flask-3.0.3-py3-none-any.whl.metadata (3.2 kB)
Collecting Werkzeug<3.1 (from dash)
  Downloading werkzeug-3.0.6-py3-none-any.whl.metadata (3.7 kB)
Requirement already satisfied: plotly>=5.0.0 in /usr/local/lib/python3.11/dist-packages (from dash) (5.24.1)
Collecting dash-html-components==2.0.0 (from dash)
  Downloading dash_html_components-2.0.0-py3-none-any.whl.metadata (3.8 kB)
Collecting dash-core-components==2.0.0 (from dash)
  Downloading dash_core_components-2.0.0-py3-none-any.whl.metadata (2.9 kB)
Collecting dash-table==5.0.0 (from dash)
  Downloading dash_table-5.0.0-py3-none-any.whl.metadata (2.4 kB)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.11/dist-packages (from dash) (8.6.1)
Requirement already satisfied: typing-extensions>=4.1.1 in /usr/local/lib/python3.11/dist-packages (from dash) (4)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from dash) (2.32.3)
Collecting retrying (from dash)
  Downloading retrying-1.3.4-py3-none-any.whl.metadata (6.9 kB)
Requirement already satisfied: nest-asyncio in /usr/local/lib/python3.11/dist-packages (from dash) (1.6.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from dash) (75.1.0)
Requirement already satisfied: Jinja2>=3.1.2 in /usr/local/lib/python3.11/dist-packages (from Flask<3.1,>=1.0.4->dash) (3.1.2)
Requirement already satisfied: itsdangerous>=2.1.2 in /usr/local/lib/python3.11/dist-packages (from Flask<3.1,>=1.0.4->dash) (2.1.2)
Requirement already satisfied: click>=8.1.3 in /usr/local/lib/python3.11/dist-packages (from Flask<3.1,>=1.0.4->dash) (8.1.3)
Requirement already satisfied: blinker>=1.6.2 in /usr/local/lib/python3.11/dist-packages (from Flask<3.1,>=1.0.4->dash) (1.6.2)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.11/dist-packages (from plotly>=5.0.0->dash) (6.2.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from plotly>=5.0.0->dash) (23.1)
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.11/dist-packages (from Werkzeug<3.1->dash) (2.1.1)
```

Requirement already satisfied: zipp>=3.20 in /usr/local/lib/python3.11/dist-packages (from importlib-metadata->dash)
 Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->dash)
 Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->dash) (3.1)
 Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->dash)
 Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->dash)
 Requirement already satisfied: six>=1.7.0 in /usr/local/lib/python3.11/dist-packages (from retrying->dash) (1.17).
 Downloading dash-2.18.2-py3-none-any.whl (7.8 MB)

7.8/7.8 MB 52.3 MB/s eta 0:00:00

Downloading dash_core_components-2.0.0-py3-none-any.whl (3.8 kB)

Downloading dash_html_components-2.0.0-py3-none-any.whl (4.1 kB)

Downloading dash_table-5.0.0-py3-none-any.whl (3.9 kB)

Downloading flask-3.0.3-py3-none-any.whl (101 kB)

101.7/101.7 kB 8.7 MB/s eta 0:00:00

Downloading werkzeug-3.0.6-py3-none-any.whl (227 kB)

228.0/228.0 kB 16.1 MB/s eta 0:00:00

Downloading retrying-1.3.4-py3-none-any.whl (11 kB)

Installing collected packages: dash-table, dash-html-components, dash-core-components, Werkzeug, retrying, Flask,

Attempting uninstall: Werkzeug

Found existing installation: Werkzeug 3.1.3

Uninstalling Werkzeug-3.1.3:

Successfully uninstalled Werkzeug-3.1.3

Attempting uninstall: Flask

Found existing installation: Flask 3.1.0

Uninstalling Flask-3.1.0:

Successfully uninstalled Flask-3.1.0

Successfully installed Flask-3.0.3 Werkzeug-3.0.6 dash-2.18.2 dash-core-components-2.0.0 dash-html-components-2.0

```
import dash
from dash import dcc, html
import plotly.express as px
import pandas as pd
import numpy as np

# Sample data
np.random.seed(42)
df_viz = pd.DataFrame({
    "Date": pd.date_range(start="2024-01-01", periods=10, freq='D'),
    "Net Revenue": np.random.randint(-300000, 500000, 10),
    "Expenditures": np.random.randint(100000, 400000, 10),
    "Customer Satisfaction": np.random.uniform(7.0, 9.0, 10),
    "Department": np.random.choice(["Parking", "Shuttle", "Fleet"], 10)
})

# Ensure Date is datetime format
df_viz["Date"] = pd.to_datetime(df_viz["Date"])

# Net Revenue Comparison Across Departments
fig_bar = px.bar(
    df_viz, x="Department", y="Net Revenue", color="Department",
    title="Net Revenue Comparison Across Departments"
)

# Revenue and Expenditures by Department
fig_grouped_bar = px.bar(
    df_viz, x="Department", y=["Net Revenue", "Expenditures"], barmode='group',
    title="Revenue and Expenditures by Department"
)

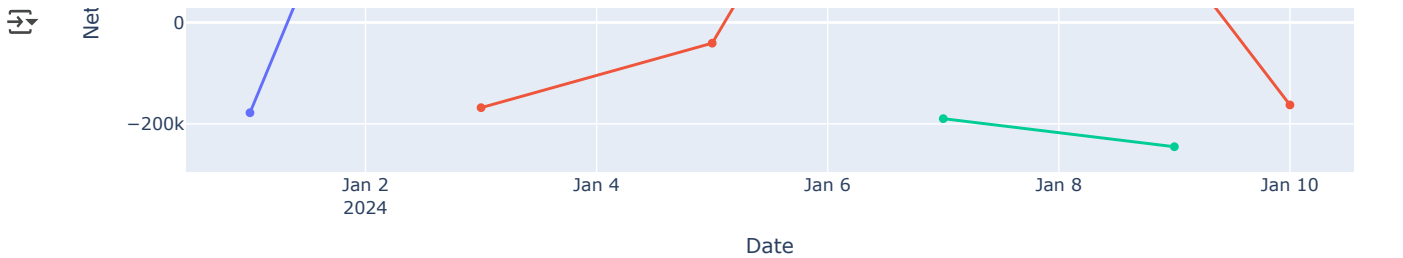
# Line Chart: Year-over-Year Revenue Growth Trend
fig_line = px.line(
    df_viz, x="Date", y="Net Revenue", color="Department",
    markers=True, title="Revenue Growth Trend Over Time"
)

# Scatter Plot: Customer Satisfaction vs Net Revenue
fig_scatter = px.scatter(
    df_viz, x="Customer Satisfaction", y="Net Revenue",
    color="Department", size=df_viz["Net Revenue"].abs(),
    title="Customer Satisfaction vs Net Revenue"
)

# Dash App
```

```
app = dash.Dash(__name__)
app.layout = html.Div([
    html.H1("Business Insights Dashboard", style={'textAlign': 'center'}),
    dcc.Graph(figure=fig_bar),
    dcc.Graph(figure=fig_grouped_bar),
    dcc.Graph(figure=fig_line),
    dcc.Graph(figure=fig_scatter)
])

if __name__ == '__main__':
    app.run_server(debug=True)
```



Customer Satisfaction vs Net Revenue

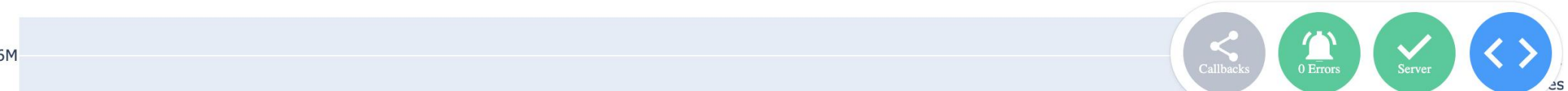


Start coding or [generate](#) with AI.

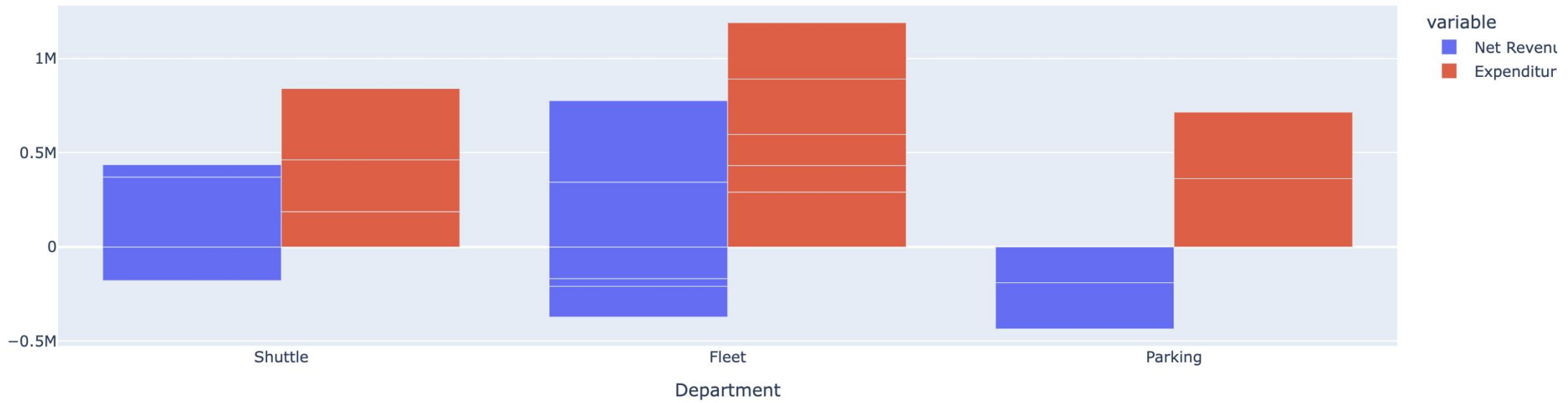
Net Revenue Comparison Across Departments



Revenue and Expenditures by Department



Revenue and Expenditures by Department



Revenue Growth Trend Over Time

