

PERSONAL PORTFOLIO

Dashboards and visualizations created using Python or MS power BI, IBM Analytics, Google Analytics

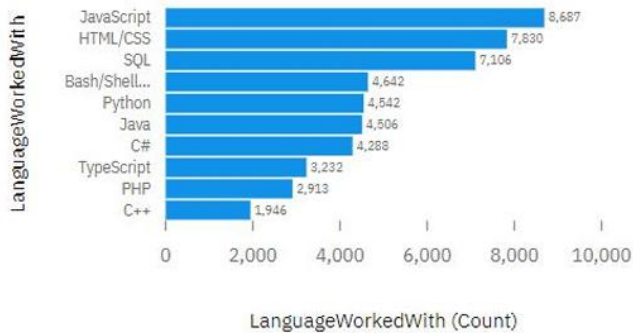
Darshit Chauhan

darshitchauhan2001@gmail.com

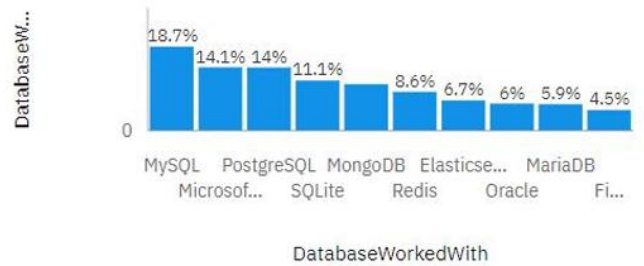
+91 8155890565

1. IBM Data Analytics course: Visualizations using Excel and IBM Cognos Analytics:

Top 10 Language Worked With



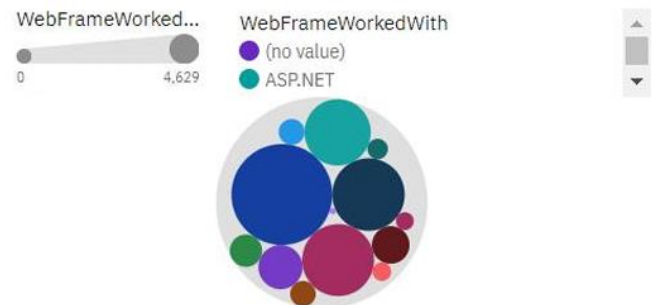
Top 10 Database Worked With



Platform Worked With



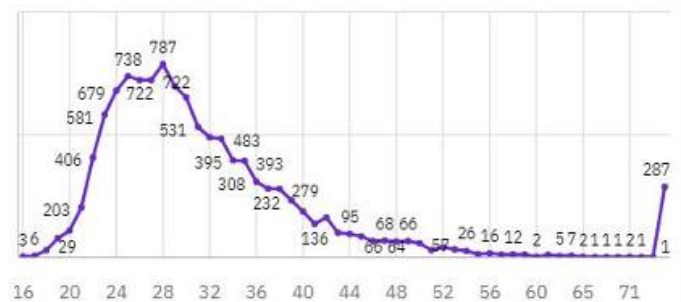
Top 10 Webframe Worked with



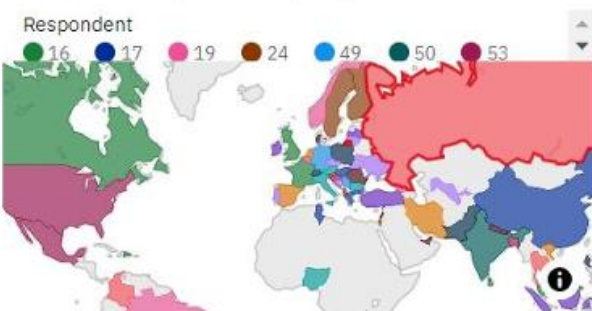
Gender by Gender



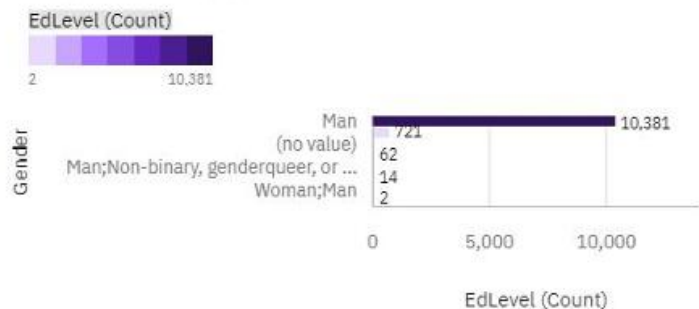
Respondent by Age



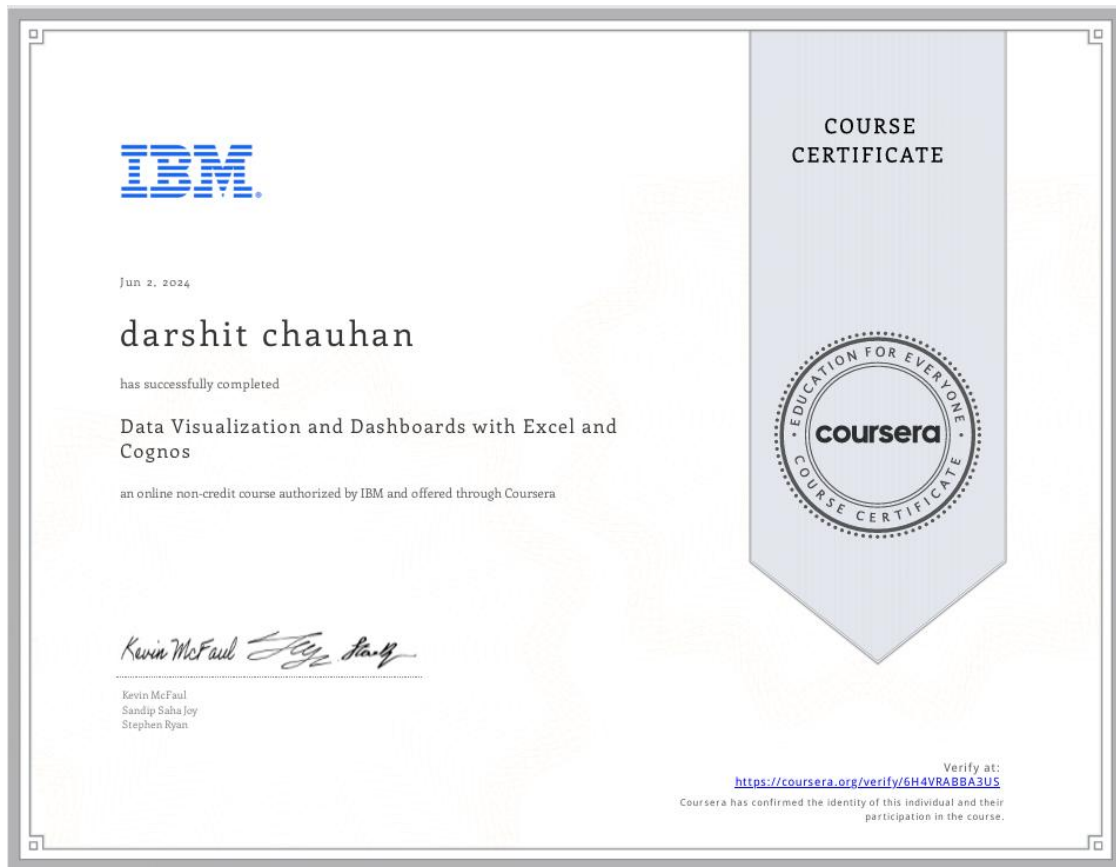
Respondent by Country regions



Respondents by gender stacked with education level

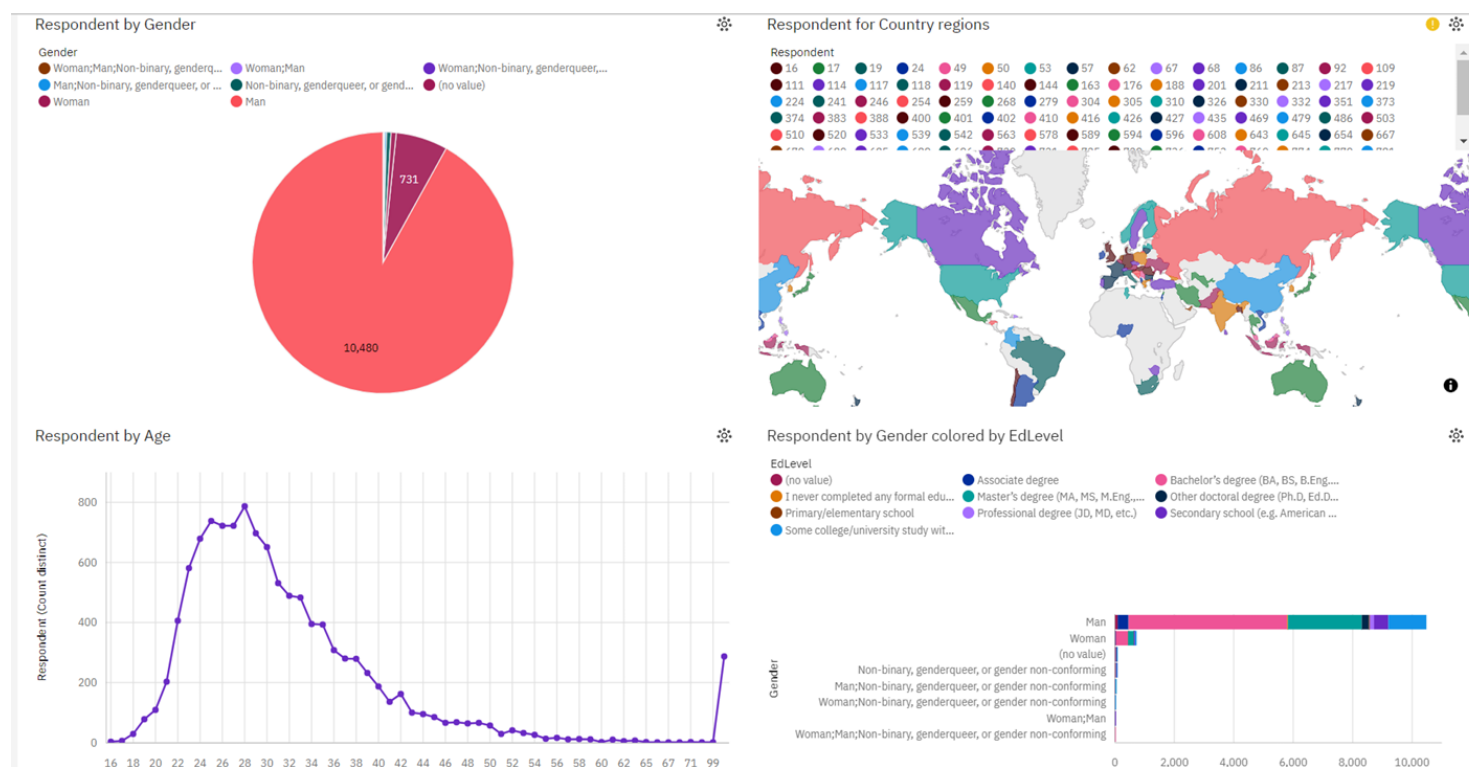


CERTIFICATION EARNED:

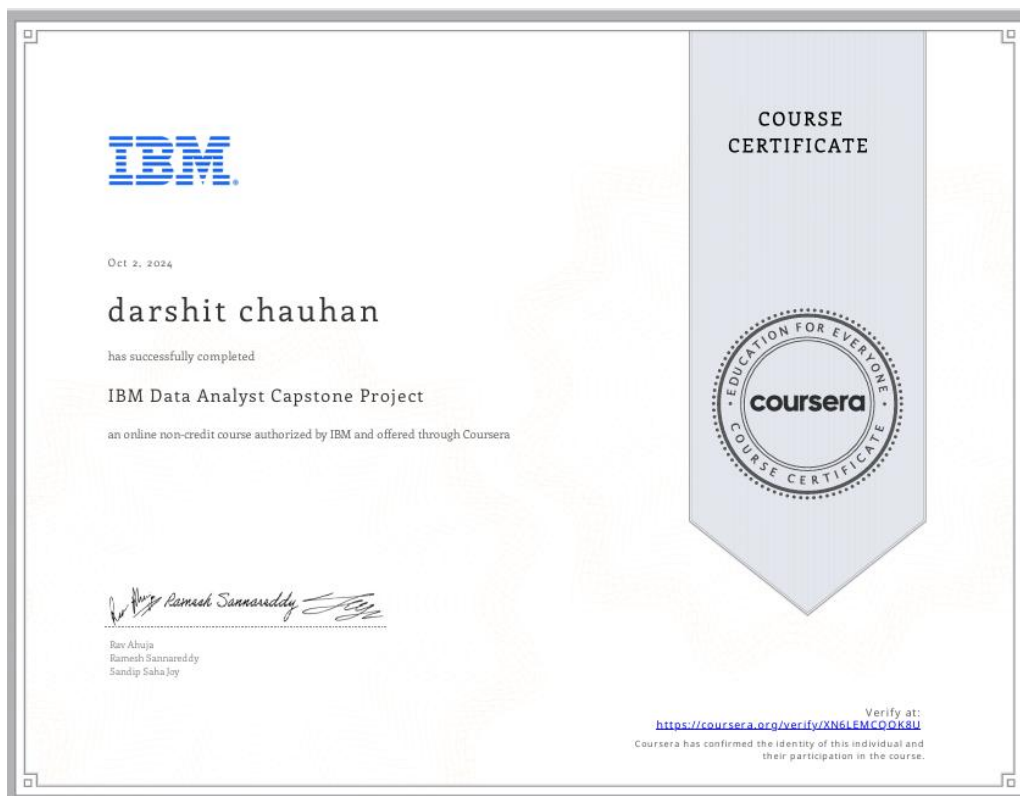


2. IBM Data Analytics course: CAPSTONE PROJECT.

DEMOGRAPHICS DASHBOARD:



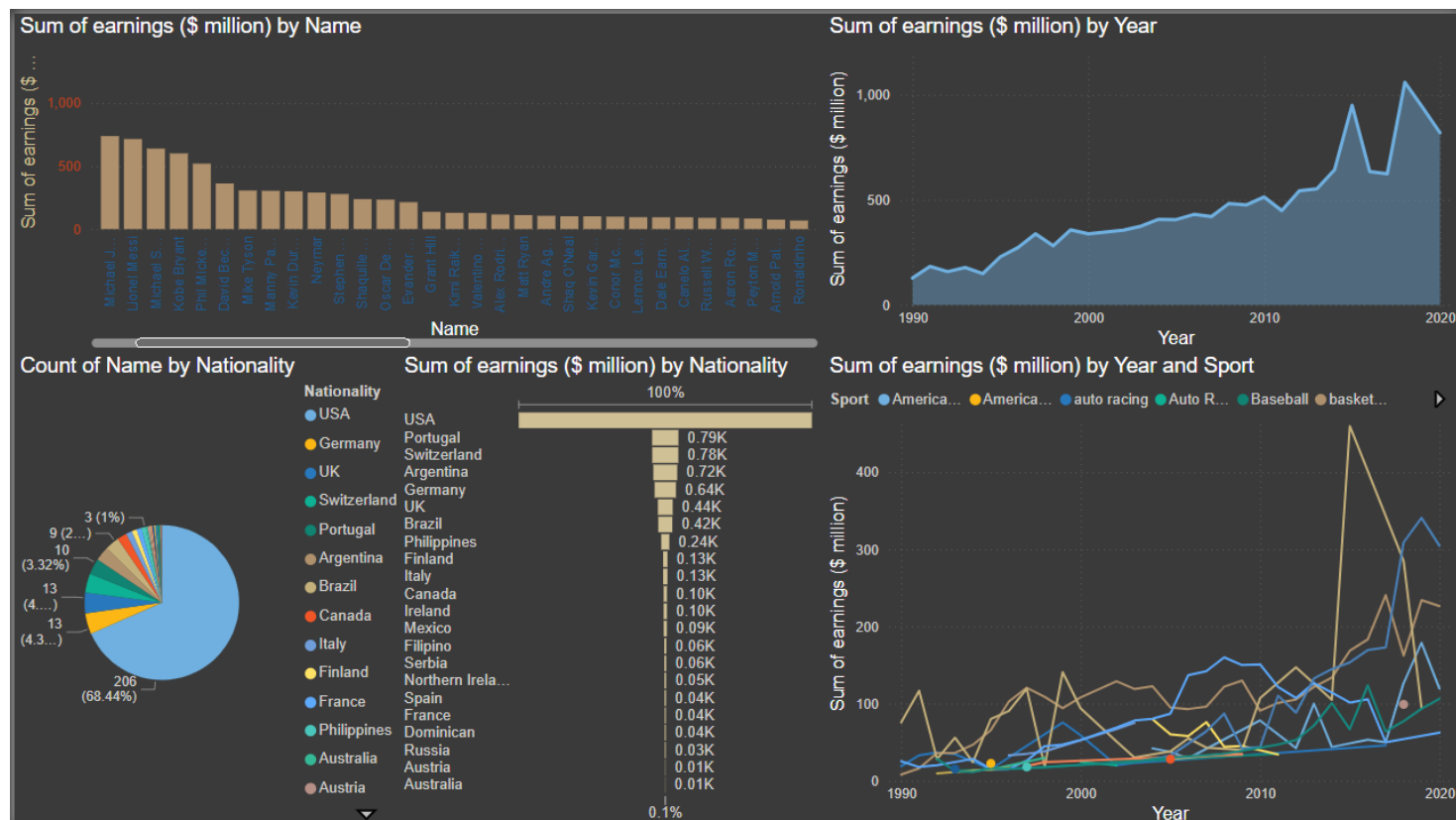
CERTIFICATION EARNED:



3. Forbes Highest Paid Athletes (1990-2020).

Dataset source: <https://www.kaggle.com/datasets/parulpandey/forbes-highest-paid-athletes-19902019>

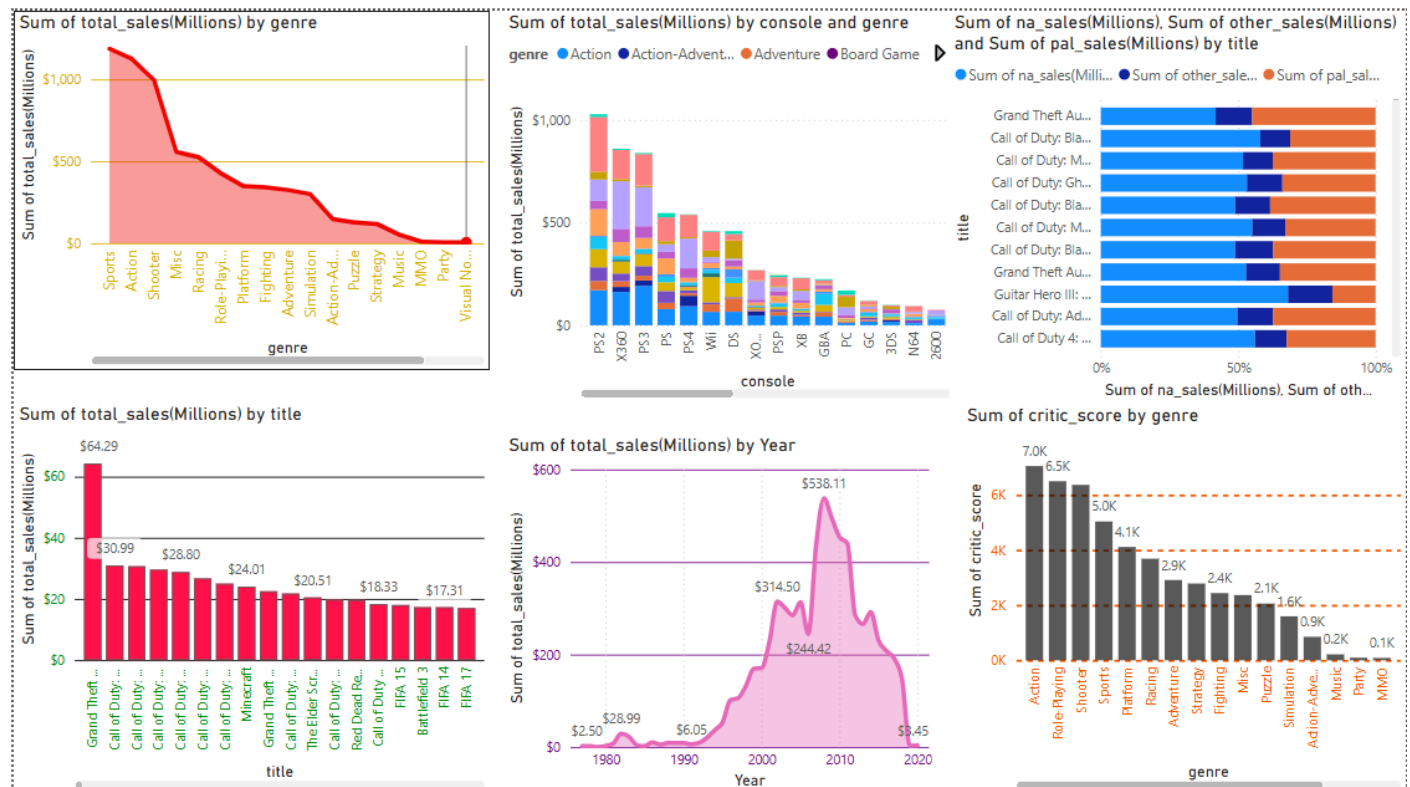
Tool used: MS Excel, MS Power BI



4. Video games sales.

Dataset source: https://mavenanalytics.io/dataplayground?dataStructure=Single%20table&order=date_added%2Cdesc

Tool used: MS Excel, MS Power BI



5. Stock Market simulation (2022-2024)

Dataset source: <https://www.kaggle.com/datasets/samayashar/stock-market-simulation-dataset>

Tool used: Python

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import matplotlib.ticker as ticker

# Load the dataset
df = pd.read_excel("synthetic_stock_data.xlsx")

# ----- 1. Trend Distribution (Improved) -----
plt.figure(figsize=(6, 4))
ax = sns.countplot(data=df, x='Trend', palette='Set2')

# Increase the number of ticks for accuracy
ax.yaxis.set_major_locator(ticker.MultipleLocator(50))
plt.title("Market Trend Distribution")
plt.xlabel("Trend")
plt.ylabel("Count")
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()

# ----- 2. Sector-wise Metrics -----
sector_summary = df.groupby('Sector').agg({
    'Close': 'mean',
    'PE_Ratio': 'mean',
    'Dividend_Yield': 'mean'
}).round(2).reset_index()

# Close Price
fig1 = px.bar(sector_summary, x='Sector', y='Close', title='Average Close Price by Sector',
              color='Sector', text='Close')
fig1.update_layout(yaxis=dict(tickmode="linear", dtick=5)) # Increase y-axis tick intervals
fig1.show()

# PE Ratio
fig2 = px.bar(sector_summary, x='Sector', y='PE_Ratio', title='Average PE Ratio by Sector',
              color='Sector', text='PE_Ratio')
fig2.update_layout(yaxis=dict(tickmode="linear", dtick=2)) # Increase y-axis tick intervals
fig2.show()

# Dividend Yield
fig3 = px.bar(sector_summary, x='Sector', y='Dividend_Yield', title='Average Dividend Yield by Sector',
              color='Sector', text='Dividend_Yield')
fig3.update_layout(yaxis=dict(tickmode="linear", dtick=0.5)) # Increase y-axis tick intervals
fig3.show()
```

```

# ----- 3. Sentiment Score by Sector -----
sentiment_summary = df.groupby('Sector')['Sentiment_Score'].mean().reset_index().sort_values(by='Sentiment_Score', ascending=False)

plt.figure(figsize=(8, 4))
ax = sns.barplot(data=sentiment_summary, x='Sentiment_Score', y='Sector', palette='coolwarm')

# Increase the number of ticks for better accuracy
ax.xaxis.set_major_locator(ticker.MultipleLocator(0.1))
plt.title("Average Sentiment Score by Sector")
plt.xlabel("Avg Sentiment Score")
plt.ylabel("Sector")
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()

# ----- 4. Sentiment vs Close Price -----
plt.figure(figsize=(6, 4))
ax = sns.scatterplot(data=df, x='Sentiment_Score', y='Close', hue='Sector', alpha=0.6)

# More ticks for better accuracy
ax.xaxis.set_major_locator(ticker.MultipleLocator(0.2))
ax.yaxis.set_major_locator(ticker.MultipleLocator(5))
plt.title("Sentiment Score vs Close Price")
plt.xlabel("Sentiment Score")
plt.ylabel("Close Price")
plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()

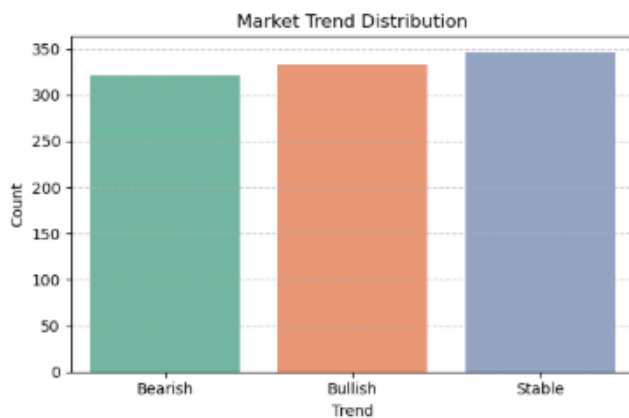
# ----- 5. Time Series Close Price (Sample Company) -----
company_name = df['Company'].unique()[0]
company_df = df[df['Company'] == company_name]

plt.figure(figsize=(10, 5))
ax = plt.plot(company_df['Date'], company_df['Close'], marker='o')

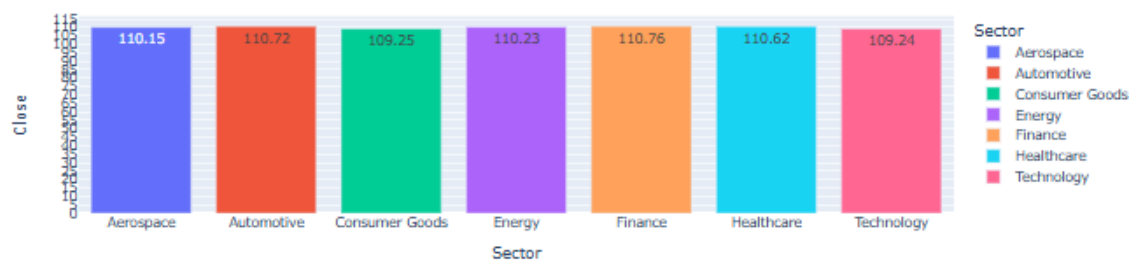
# Increase x-axis intervals for better readability
plt.xticks(rotation=45)
plt.locator_params(axis='x', nbins=10) # Increase the number of x-axis ticks

plt.title(f"Time Series Close Price: {company_name}")
plt.xlabel("Date")
plt.ylabel("Close Price")
plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()

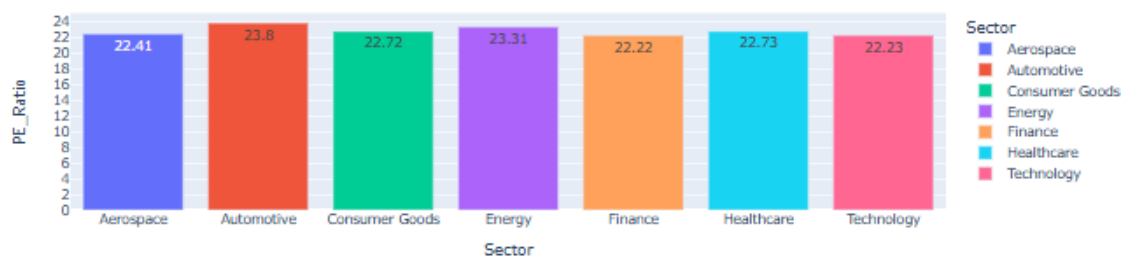
```



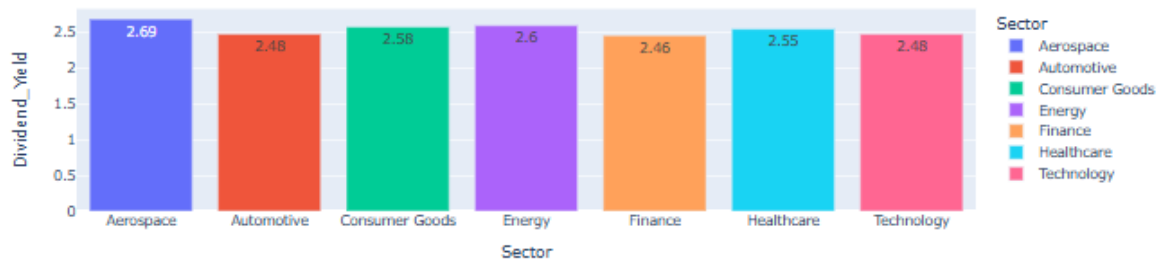
Average Close Price by Sector



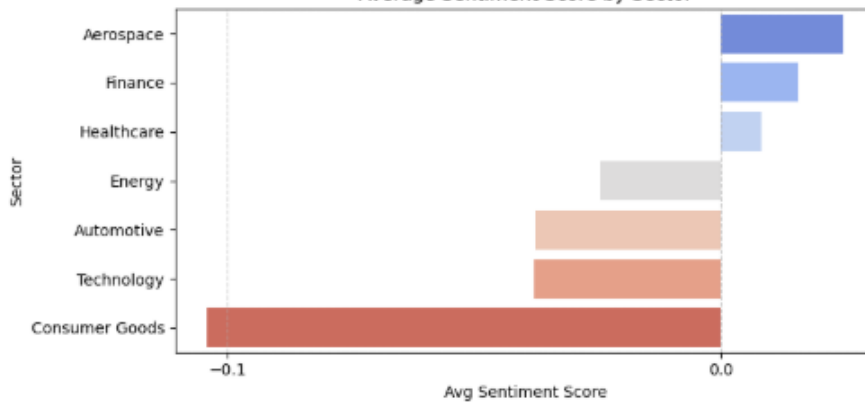
Average PE Ratio by Sector



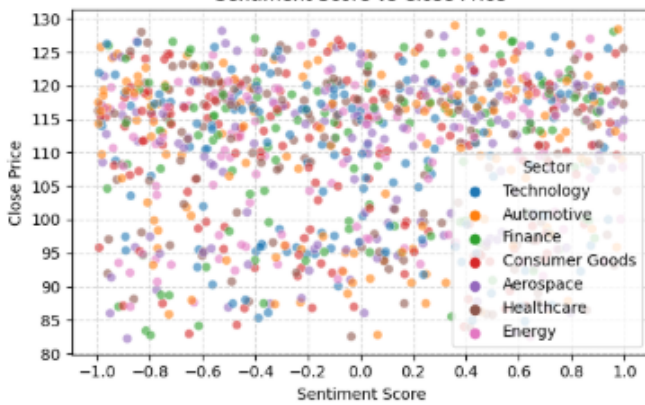
Average Dividend Yield by Sector



Average Sentiment Score by Sector



Sentiment Score vs Close Price



Time Series Close Price: Uber

