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


import warnings
warnings.simplefilter('ignore')

df = pd.read_csv("/content/largest financial services companies by
revenue.csv")

df.head()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 50,\n  \"fields\": [\n
{\n      \"column\": \"Rank\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 14,\n        \"min\": 1,\n
\"max\": 50,\n        \"num_unique_values\": 50,\n        \"samples\":
[\n          14,\n          40,\n          31\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Company\",\n      \"properties\":
{\n        \"dtype\": \"string\",\n        \"num_unique_values\": 50,\
n        \"samples\": [\n          \"Bank of America\",\n
\"Goldman Sachs\",\n          \"Aviva\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Industry\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
4,\n        \"samples\": [\n          \"Insurance\",\n
\"Investment Services\",\n          \"Conglomerate\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Revenue in (USD Million)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
44689,\n        \"min\": 14592,\n        \"max\": 245510,\n
\"num_unique_values\": 50,\n        \"samples\": [\n          93753,\n
53498,\n          62579\n        ],\n        \"semantic_type\": \"\",\
n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Net Income in (USD Millions)\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 11101,\n
\"min\": 169,\n        \"max\": 45783,\n        \"num_unique_values\":
49,\n        \"samples\": [\n          17894,\n          4972,\n
766\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Total Assest in (USD Millions)\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1282,\n        \"min\": 13,\n
\"max\": 5110,\n        \"num_unique_values\": 50,\n
\"samples\": [\n          2819,\n          1163,\n          655\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Headquarters\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 11,\n        \"samples\": [\n
\"Italy\",\n          \"United States\",\n          \"Canada\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 7 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Rank                            50 non-null     int64
 1   Company                         50 non-null     object
 2   Industry                        50 non-null     object
 3   Revenue in (USD Million)        50 non-null     int64
 4   Net Income in (USD Millions)    50 non-null     int64
 5   Total Assest in (USD Millions)  50 non-null     int64
 6   Headquarters                    50 non-null     object
dtypes: int64(4), object(3)
memory usage: 2.9+ KB
```

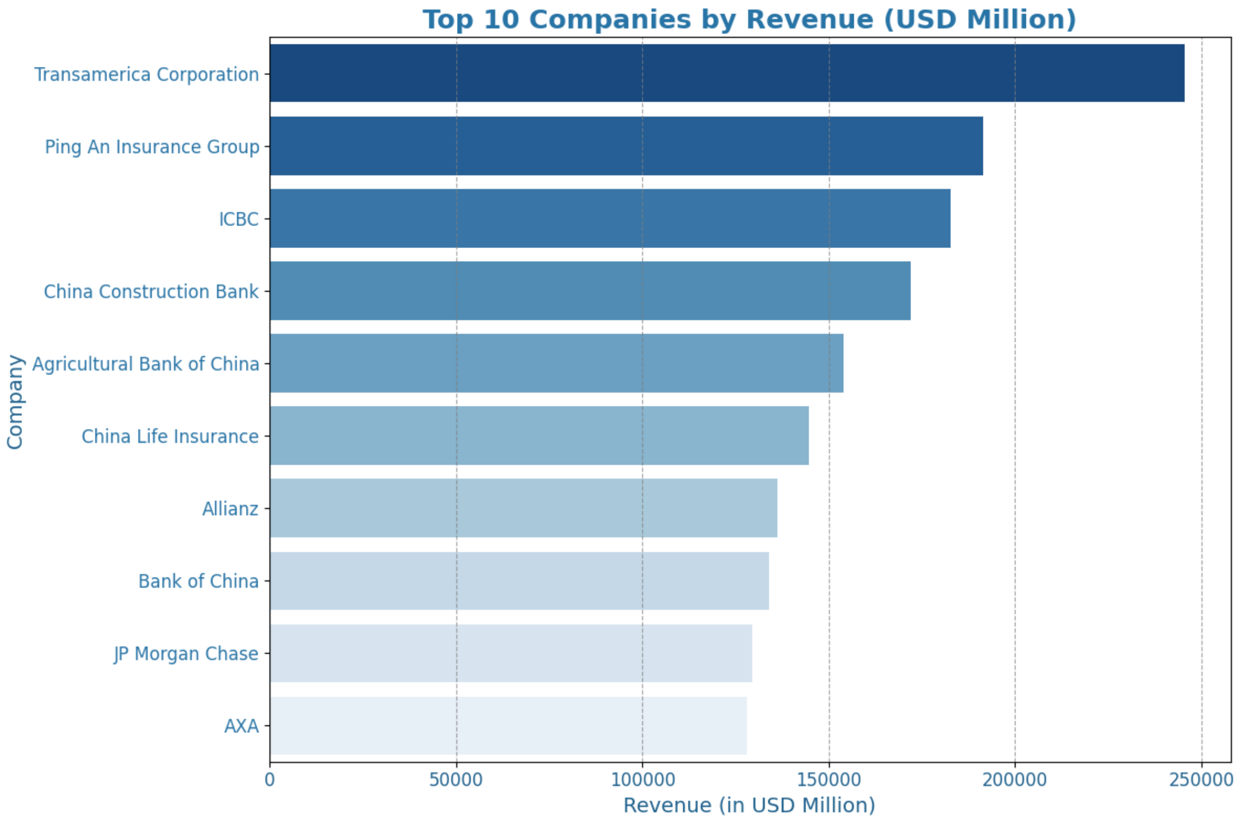
```
df.describe()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n
{\n      \"column\": \"Rank\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 17.716559962530223,\n
\"min\": 1.0,\n        \"max\": 50.0,\n        \"num_unique_values\":
6,\n        \"samples\": [\n          50.0,\n          25.5,\n
37.75\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Revenue in (USD Million)\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 75753.58568568908,\n
\"min\": 50.0,\n        \"max\": 245510.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
85435.12,\n          70736.0,\n          50.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Net Income in (USD Millions)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
14874.641706176511,\n        \"min\": 50.0,\n        \"max\":
45783.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
9369.32,\n          4963.0,\n          50.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Total Assest in (USD Millions)\",\
n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 1643.5904621405693,\n        \"min\": 13.0,\n        \"max\":
5110.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
1480.46,\n          1024.5,\n          50.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```

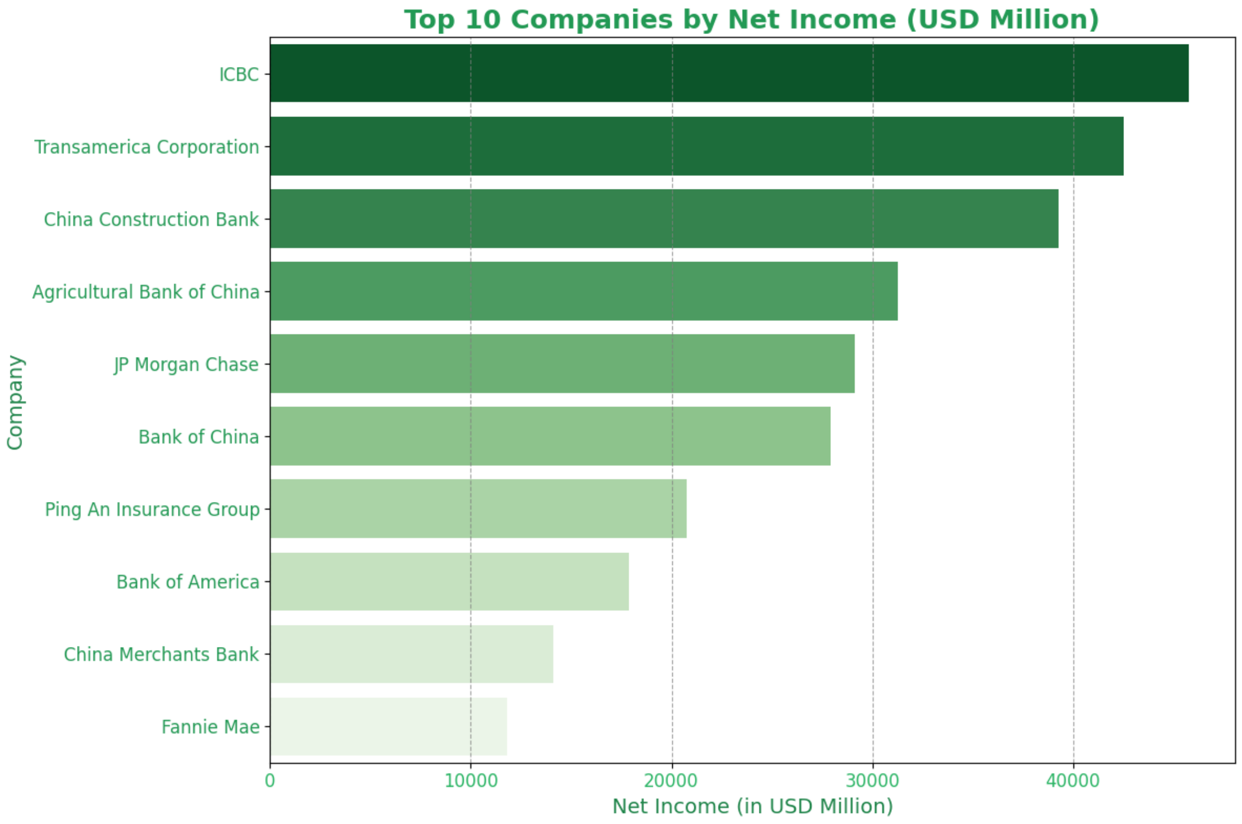
```
df.isna().sum()
```

```
Rank                              0
Company                           0
Industry                          0
Revenue in (USD Million)          0
Net Income in (USD Millions)      0
Total Assest in (USD Millions)    0
Headquarters                      0
dtype: int64
```

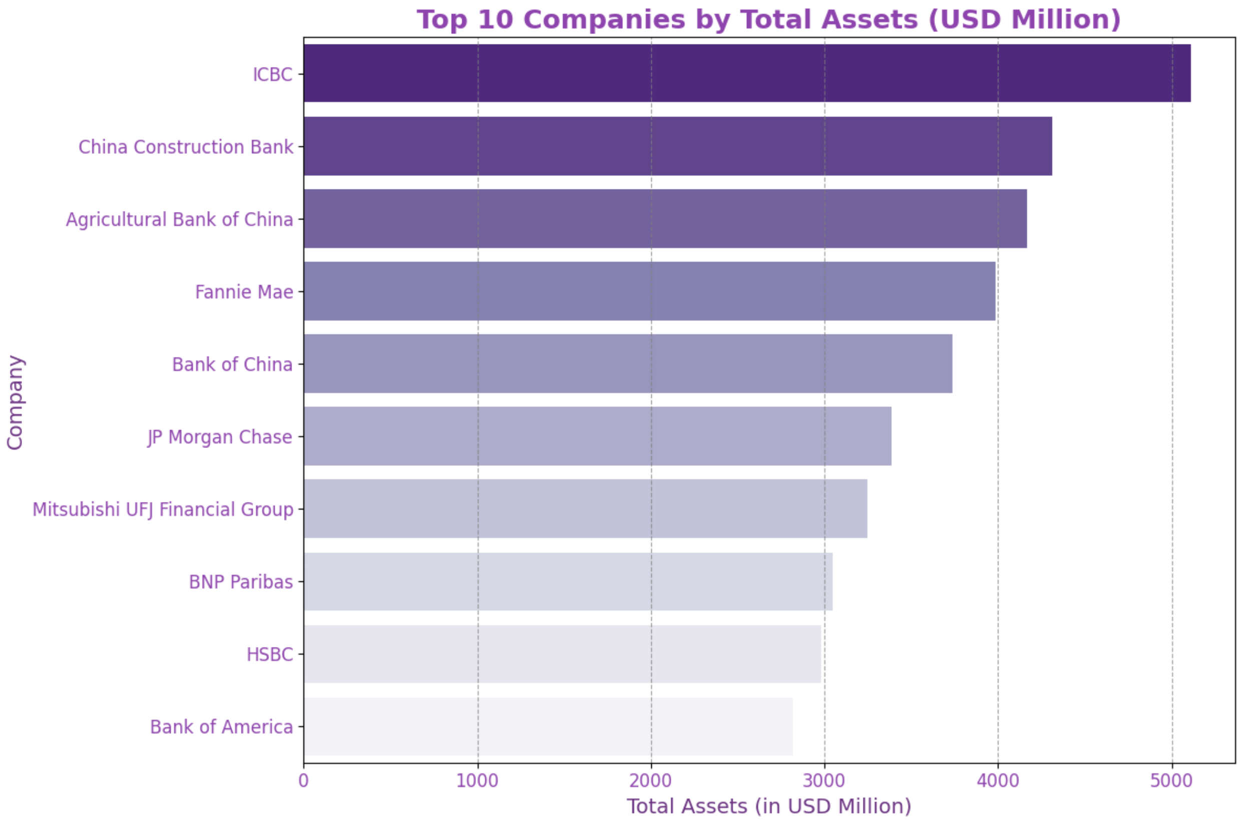
```
top_revenue = df.nlargest(10, 'Revenue in (USD Million)')
plt.figure(figsize=(12, 8))
sns.barplot(
    x=top_revenue['Revenue in (USD Million)'],
    y=top_revenue['Company'],
    palette='Blues_r'
)
plt.title('Top 10 Companies by Revenue (USD Million)', fontsize=18,
fontweight='bold', color='#2874A6')
plt.xlabel('Revenue (in USD Million)', fontsize=14, color='#1F618D')
plt.ylabel('Company', fontsize=14, color='#1F618D')
plt.xticks(fontsize=12, color='#1F618D')
plt.yticks(fontsize=12, color='#2874A6')
plt.grid(axis='x', linestyle='--', alpha=0.7, color='gray')
plt.tight_layout()
plt.show()
```



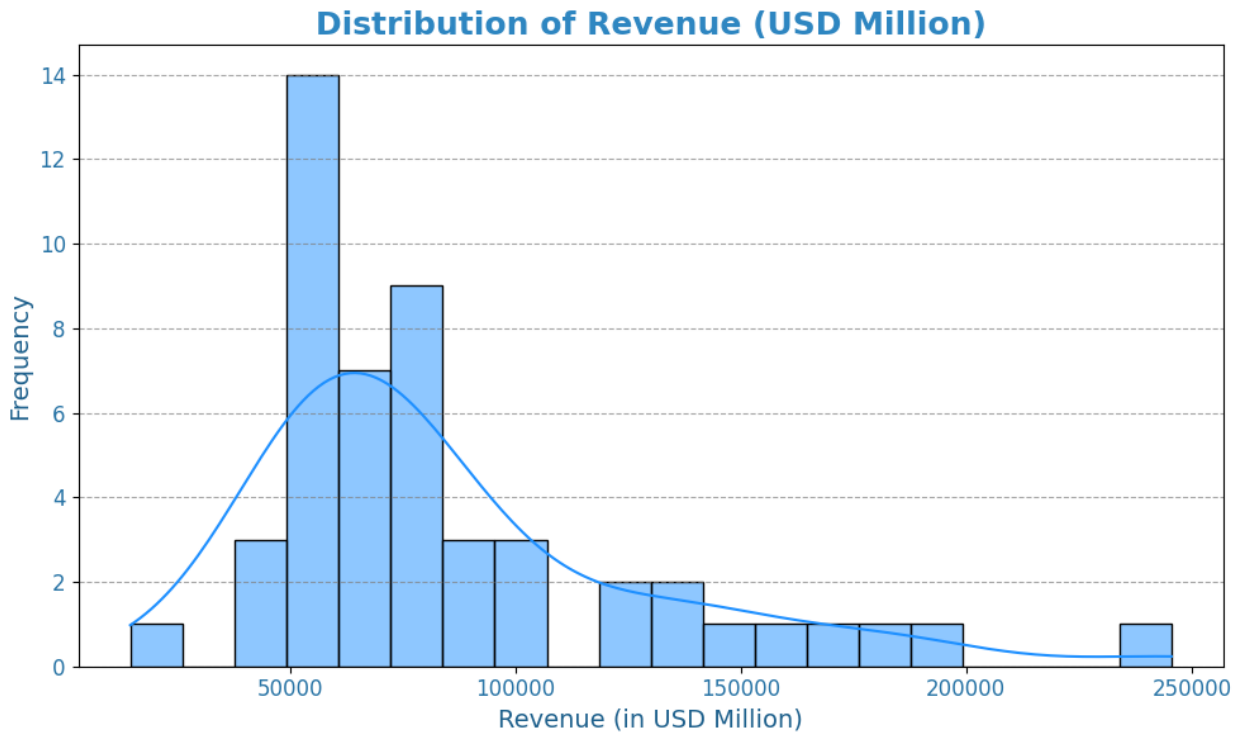
```
top_net_income = df.nlargest(10, 'Net Income in (USD Millions)')
plt.figure(figsize=(12, 8))
sns.barplot(
    x=top_net_income['Net Income in (USD Millions)'],
    y=top_net_income['Company'],
    palette='Greens_r'
)
plt.title('Top 10 Companies by Net Income (USD Million)', fontsize=18,
fontweight='bold', color='#229954')
plt.xlabel('Net Income (in USD Million)', fontsize=14,
color='#1E8449')
plt.ylabel('Company', fontsize=14, color='#1E8449')
plt.xticks(fontsize=12, color='#28B463')
plt.yticks(fontsize=12, color='#229954')
plt.grid(axis='x', linestyle='--', alpha=0.7, color='gray')
plt.tight_layout()
plt.show()
```



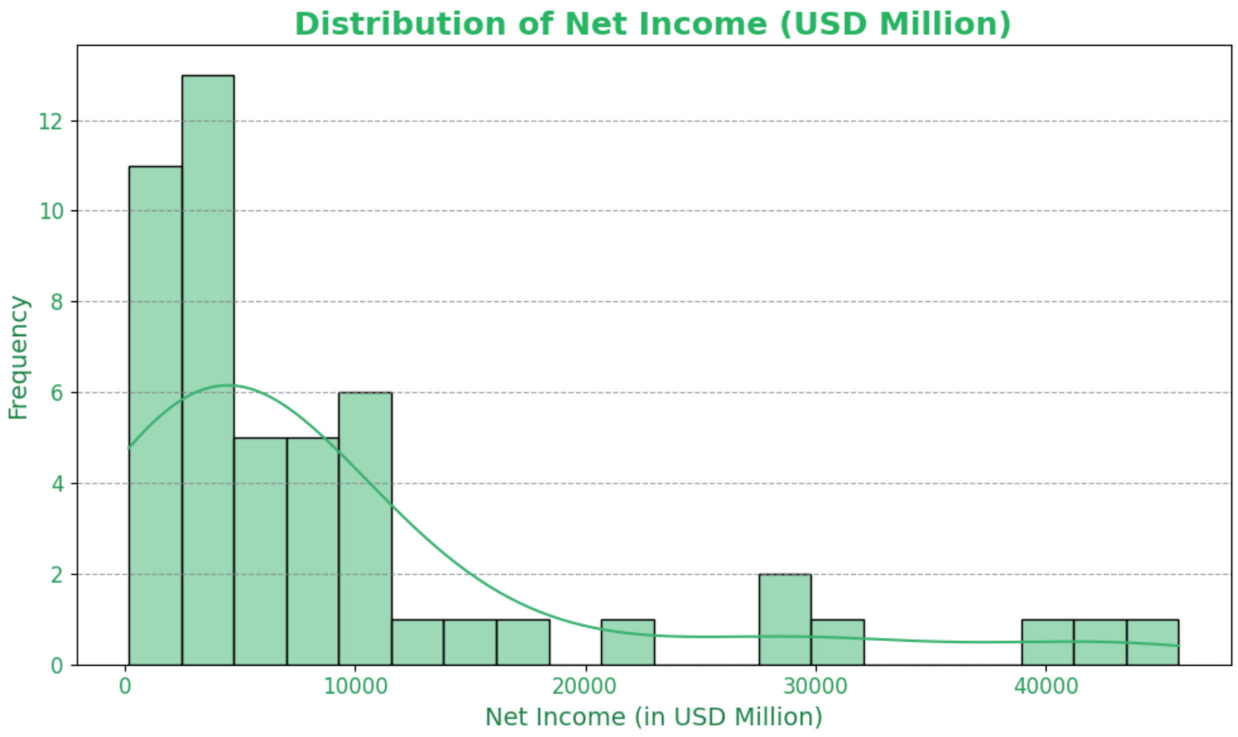
```
top_assets = df.nlargest(10, 'Total Assest in (USD Millions)')
plt.figure(figsize=(12, 8))
sns.barplot(
    x=top_assets['Total Assest in (USD Millions)'],
    y=top_assets['Company'],
    palette='Purples_r'
)
plt.title('Top 10 Companies by Total Assets (USD Million)',
fontsize=18, fontweight='bold', color='#8E44AD')
plt.xlabel('Total Assets (in USD Million)', fontsize=14,
color='#6C3483')
plt.ylabel('Company', fontsize=14, color='#6C3483')
plt.xticks(fontsize=12, color='#8E44AD')
plt.yticks(fontsize=12, color='#8E44AD')
plt.grid(axis='x', linestyle='--', alpha=0.7, color='gray')
plt.tight_layout()
plt.show()
```



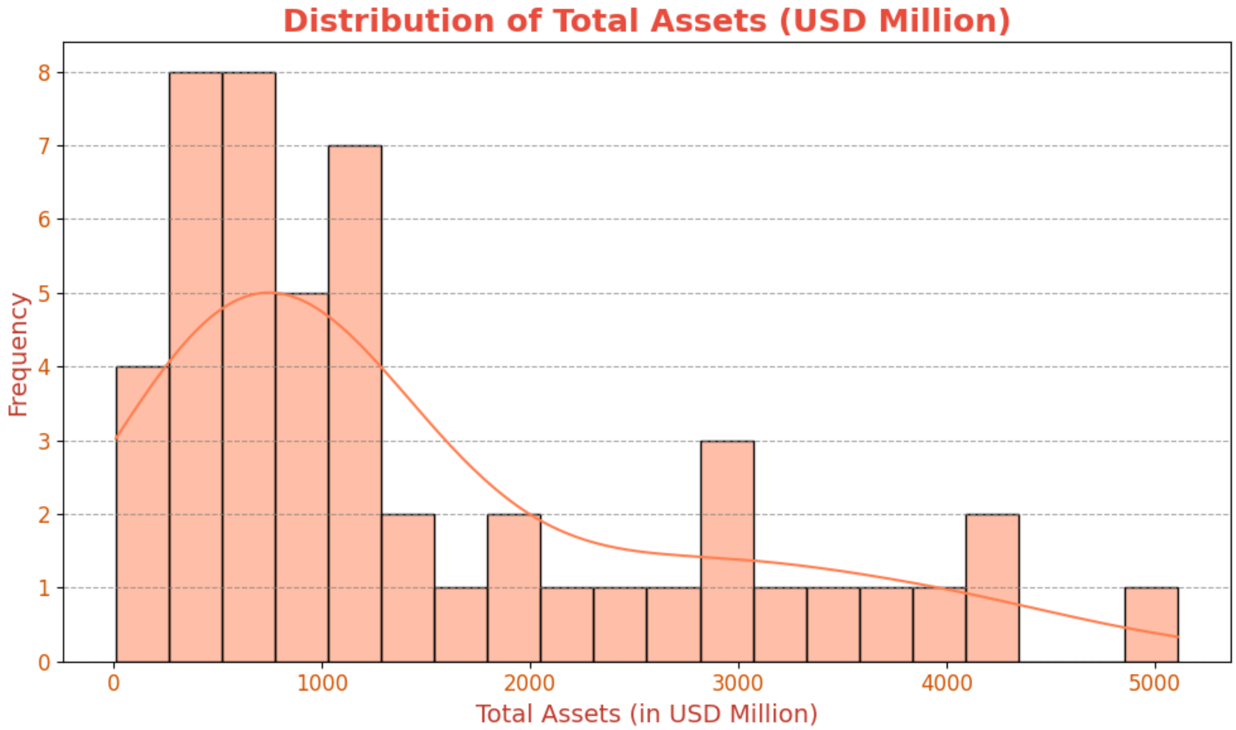
```
plt.figure(figsize=(10, 6))
sns.histplot(
    df['Revenue in (USD Million)'],
    bins=20,
    kde=True,
    color='dodgerblue'
)
plt.title('Distribution of Revenue (USD Million)', fontsize=18,
fontweight='bold', color='#2E86C1')
plt.xlabel('Revenue (in USD Million)', fontsize=14, color='#1F618D')
plt.ylabel('Frequency', fontsize=14, color='#1F618D')
plt.xticks(fontsize=12, color='#2874A6')
plt.yticks(fontsize=12, color='#2874A6')
plt.grid(axis='y', linestyle='--', alpha=0.7, color='gray')
plt.tight_layout()
plt.show()
```



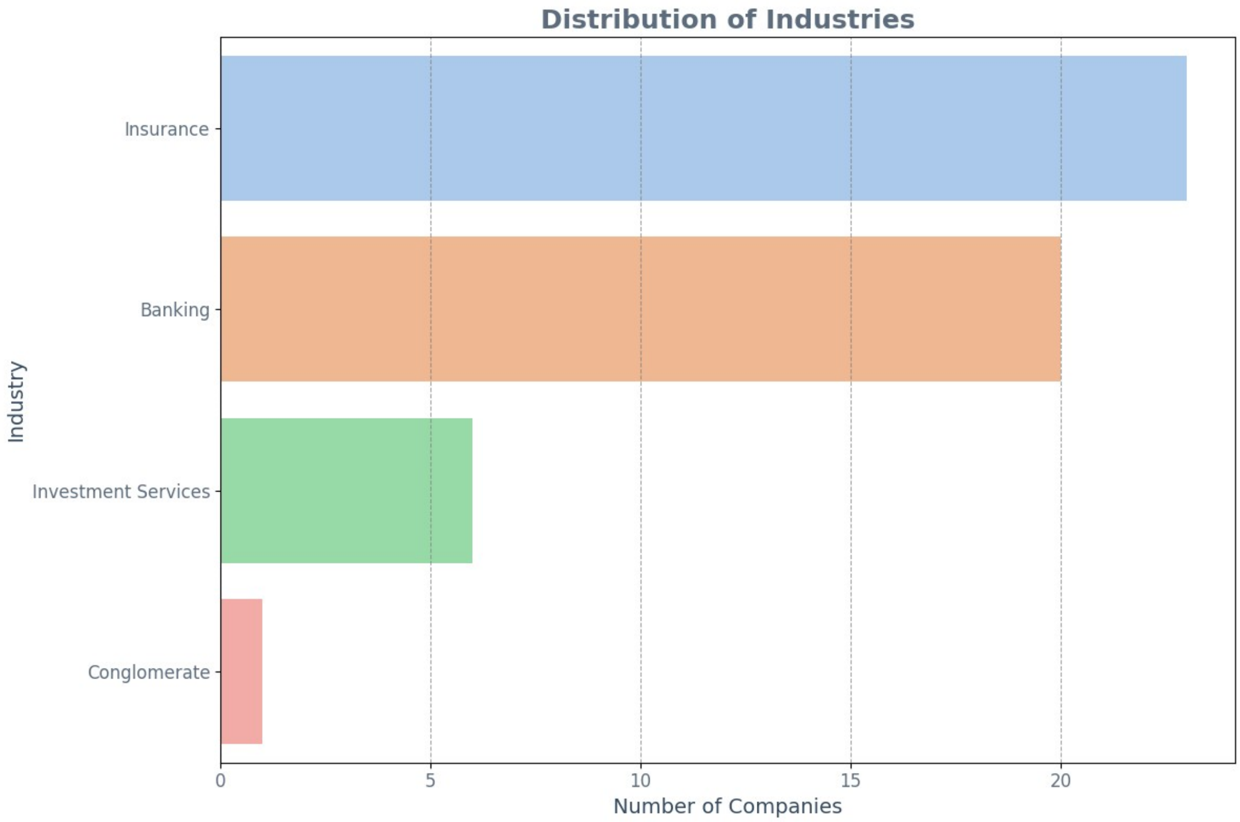
```
plt.figure(figsize=(10, 6))
sns.histplot(
    df['Net Income in (USD Millions)'],
    bins=20,
    kde=True,
    color='mediumseagreen'
)
plt.title('Distribution of Net Income (USD Million)', fontsize=18,
fontweight='bold', color='#28B463')
plt.xlabel('Net Income (in USD Million)', fontsize=14,
color='#1D8348')
plt.ylabel('Frequency', fontsize=14, color='#1D8348')
plt.xticks(fontsize=12, color='#239B56')
plt.yticks(fontsize=12, color='#239B56')
plt.grid(axis='y', linestyle='--', alpha=0.7, color='gray')
plt.tight_layout()
plt.show()
```



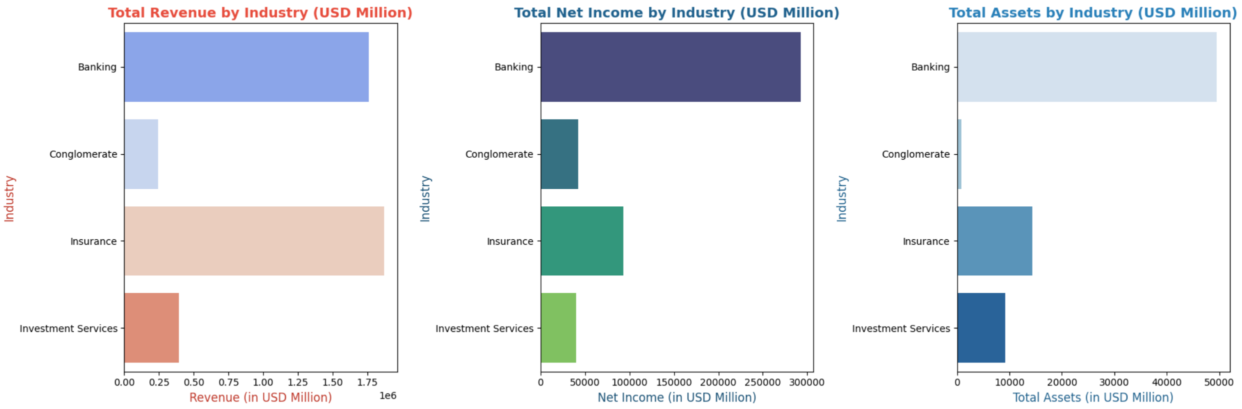
```
plt.figure(figsize=(10, 6))
sns.histplot(
    df['Total Assest in (USD Millions)'],
    bins=20,
    kde=True,
    color='coral'
)
plt.title('Distribution of Total Assets (USD Million)', fontsize=18,
fontweight='bold', color='#E74C3C')
plt.xlabel('Total Assets (in USD Million)', fontsize=14,
color='#C0392B')
plt.ylabel('Frequency', fontsize=14, color='#C0392B')
plt.xticks(fontsize=12, color='#D35400')
plt.yticks(fontsize=12, color='#D35400')
plt.grid(axis='y', linestyle='--', alpha=0.7, color='gray')
plt.tight_layout()
plt.show()
```



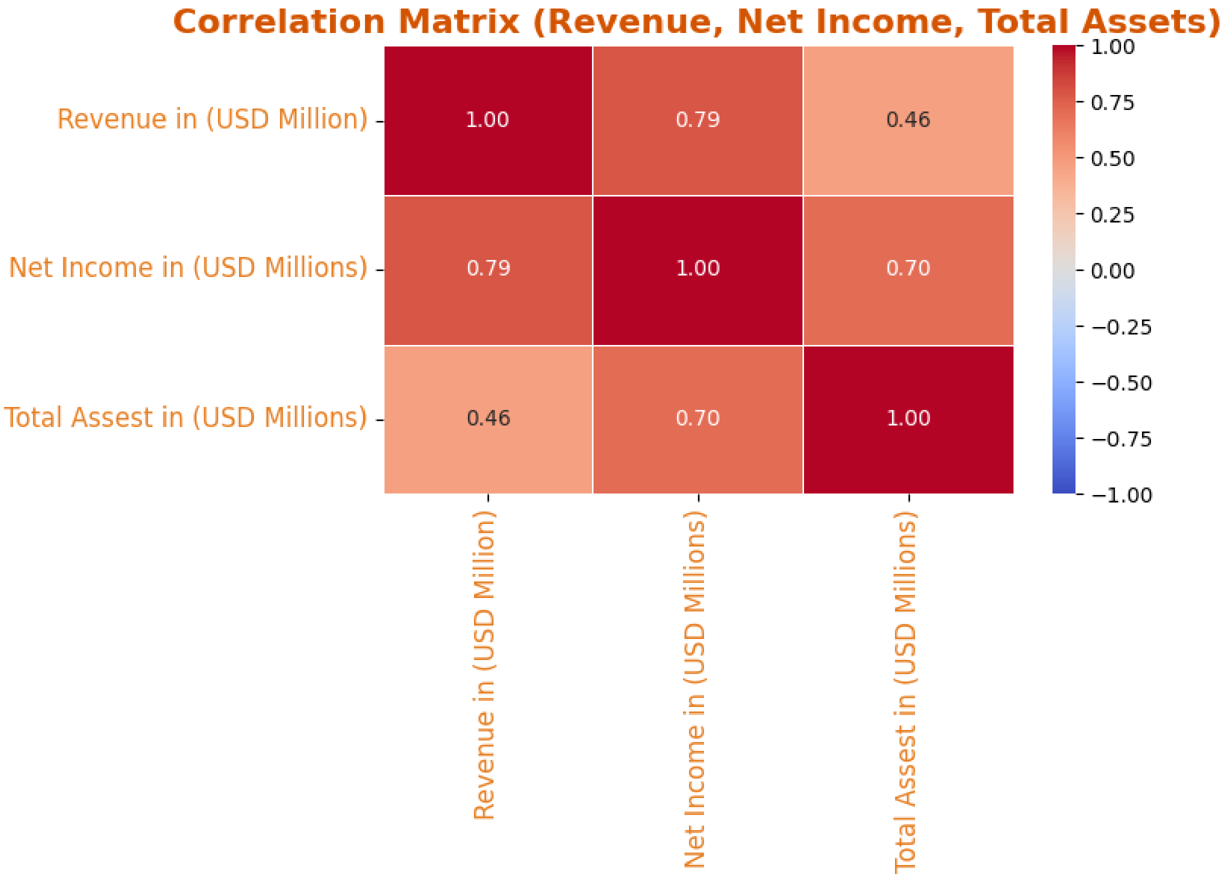
```
plt.figure(figsize=(12, 8))
industry_counts = df['Industry'].value_counts()
sns.barplot(
    x=industry_counts.values,
    y=industry_counts.index,
    palette='pastel'
)
plt.title('Distribution of Industries', fontsize=18,
fontweight='bold', color='#5D6D7E')
plt.xlabel('Number of Companies', fontsize=14, color='#34495E')
plt.ylabel('Industry', fontsize=14, color='#34495E')
plt.xticks(fontsize=12, color='#5D6D7E')
plt.yticks(fontsize=12, color='#5D6D7E')
plt.grid(axis='x', linestyle='--', alpha=0.7, color='gray')
plt.tight_layout()
plt.show()
```



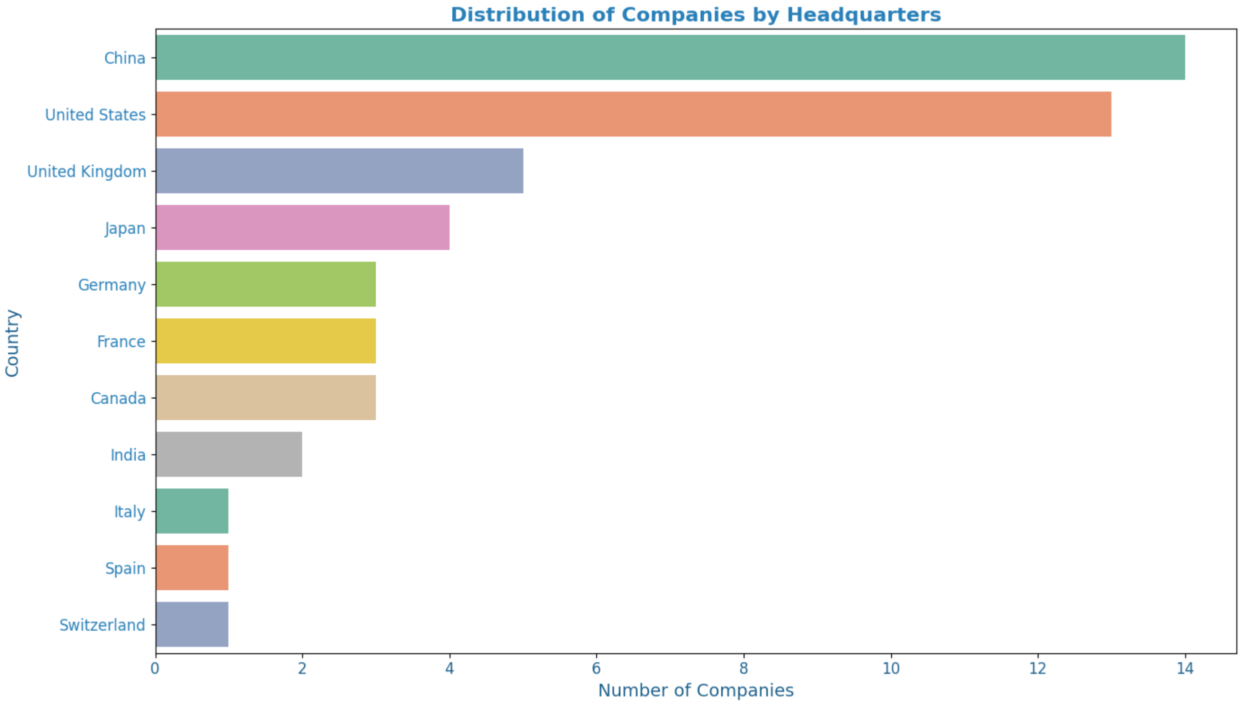
```
industry_aggregates = df.groupby('Industry')[['Revenue in (USD
Million)', 'Net Income in (USD Millions)', 'Total Assest in (USD
Millions)']].sum().reset_index()

fig, axes = plt.subplots(1, 3, figsize=(18, 6), sharey=False)

# Revenue by Industry
sns.barplot(
    x=industry_aggregates['Revenue in (USD Million)'],
    y=industry_aggregates['Industry'],
    palette='coolwarm',
    ax=axes[0]
)
axes[0].set_title('Total Revenue by Industry (USD Million)',
fontsize=14, fontweight='bold', color='#E74C3C')
axes[0].set_xlabel('Revenue (in USD Million)', fontsize=12,
color='#C0392B')
axes[0].set_ylabel('Industry', fontsize=12, color='#C0392B')

# Net Income by Industry
sns.barplot(
    x=industry_aggregates['Net Income in (USD Millions)'],
    y=industry_aggregates['Industry'],
    palette='viridis',
    ax=axes[1]
)
axes[1].set_title('Total Net Income by Industry (USD Million)',
fontsize=14, fontweight='bold', color='#1F618D')
axes[1].set_xlabel('Net Income (in USD Million)', fontsize=12,
color='#1A5276')
axes[1].set_ylabel('Industry', fontsize=12, color='#1A5276')

# Total Assets by Industry
sns.barplot(
    x=industry_aggregates['Total Assest in (USD Millions)'],
    y=industry_aggregates['Industry'],
    palette='Blues',
    ax=axes[2]
)
axes[2].set_title('Total Assets by Industry (USD Million)',
fontsize=14, fontweight='bold', color='#2980B9')
axes[2].set_xlabel('Total Assets (in USD Million)', fontsize=12,
color='#1F618D')
axes[2].set_ylabel('Industry', fontsize=12, color='#1F618D')

plt.tight_layout()
plt.show()
```



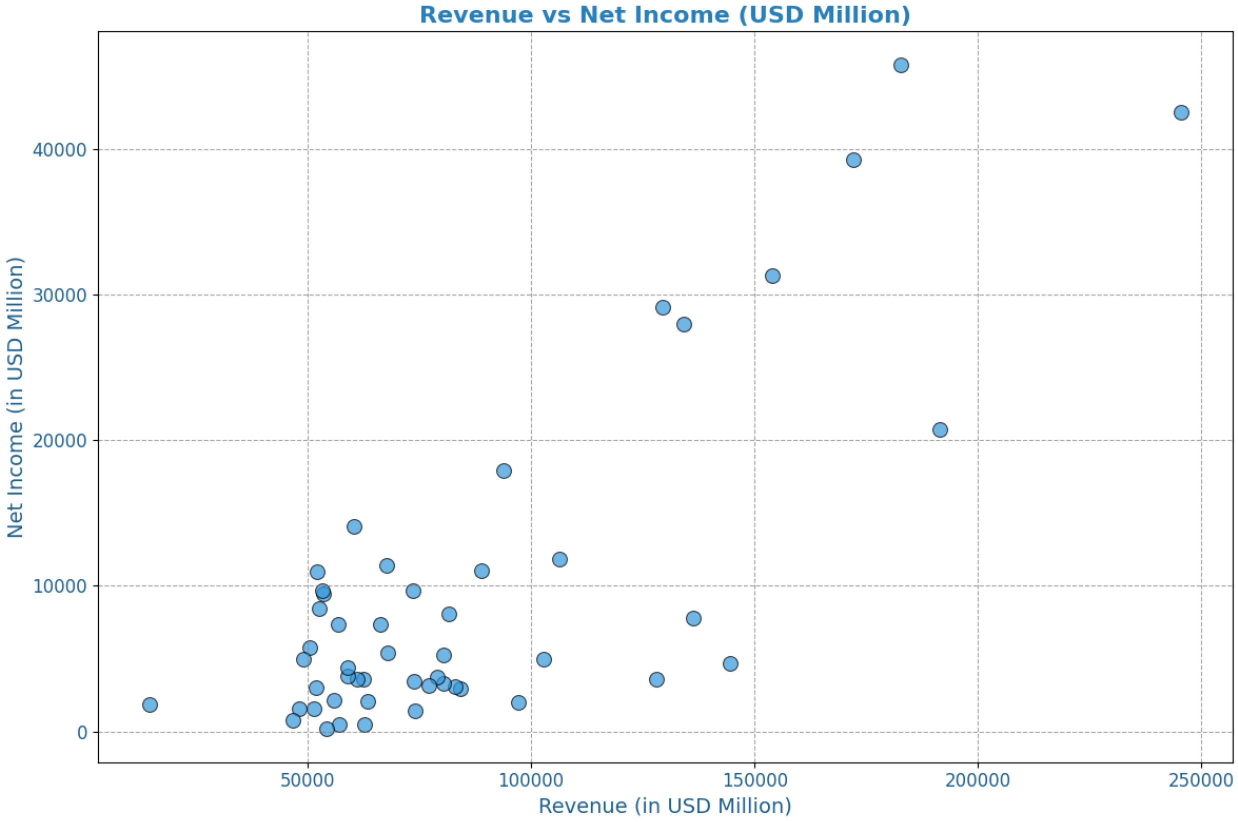
```
corr = df[['Revenue in (USD Million)', 'Net Income in (USD Millions)',
'Total Assest in (USD Millions)']].corr()

plt.figure(figsize=(8, 6))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f', cbar=True,
linewidths=0.5, vmin=-1, vmax=1)
plt.title('Correlation Matrix (Revenue, Net Income, Total Assets)',
fontsize=16, fontweight='bold', color='#D35400')
plt.xticks(fontsize=12, color='#E67E22')
plt.yticks(fontsize=12, color='#E67E22')
plt.tight_layout()
plt.show()
```

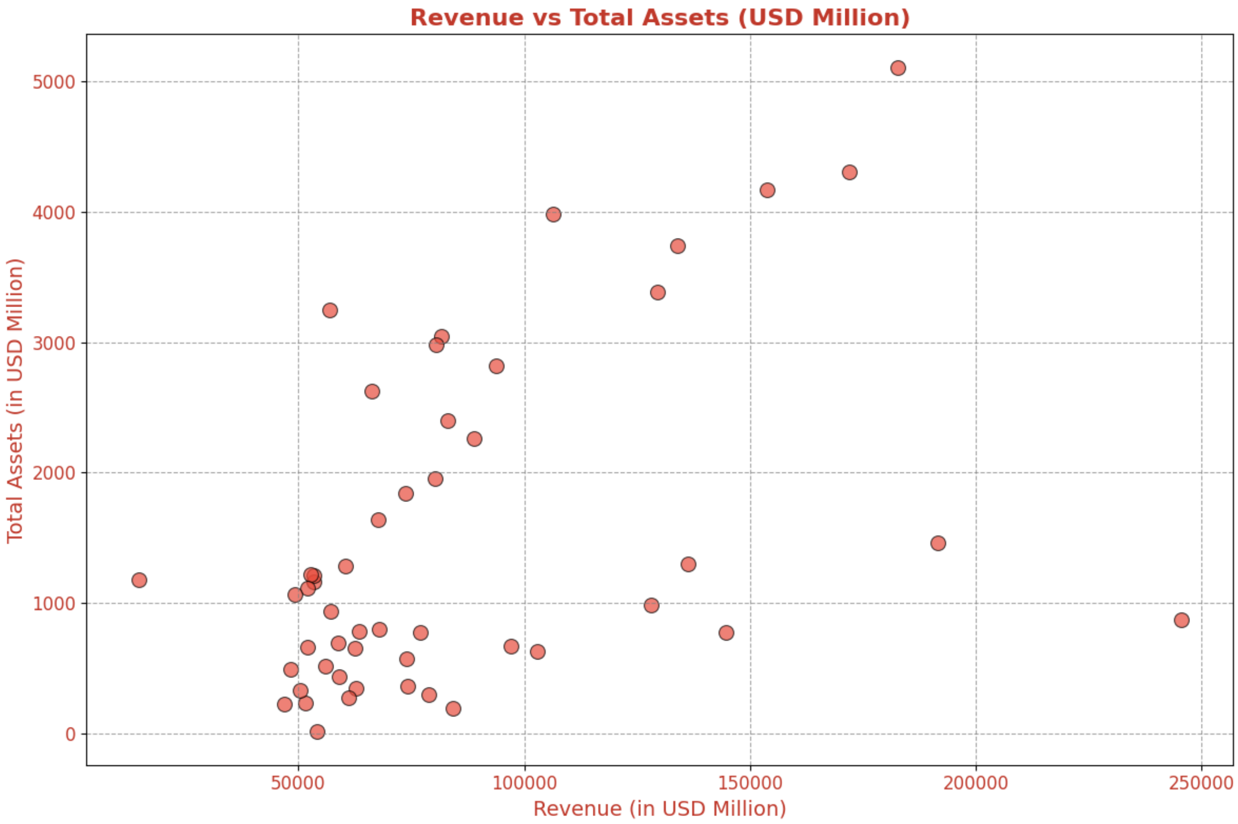


```
# number of companies per country (Headquarters)
hq_counts = df['Headquarters'].value_counts()

plt.figure(figsize=(14, 8))
sns.barplot(
    x=hq_counts.values,
    y=hq_counts.index,
    palette='Set2'
)
plt.title('Distribution of Companies by Headquarters', fontsize=16,
fontweight='bold', color='#2980B9')
plt.xlabel('Number of Companies', fontsize=14, color='#1F618D')
plt.ylabel('Country', fontsize=14, color='#1F618D')
plt.xticks(fontsize=12, color='#1F618D')
plt.yticks(fontsize=12, color='#2980B9')
plt.tight_layout()
plt.show()
```



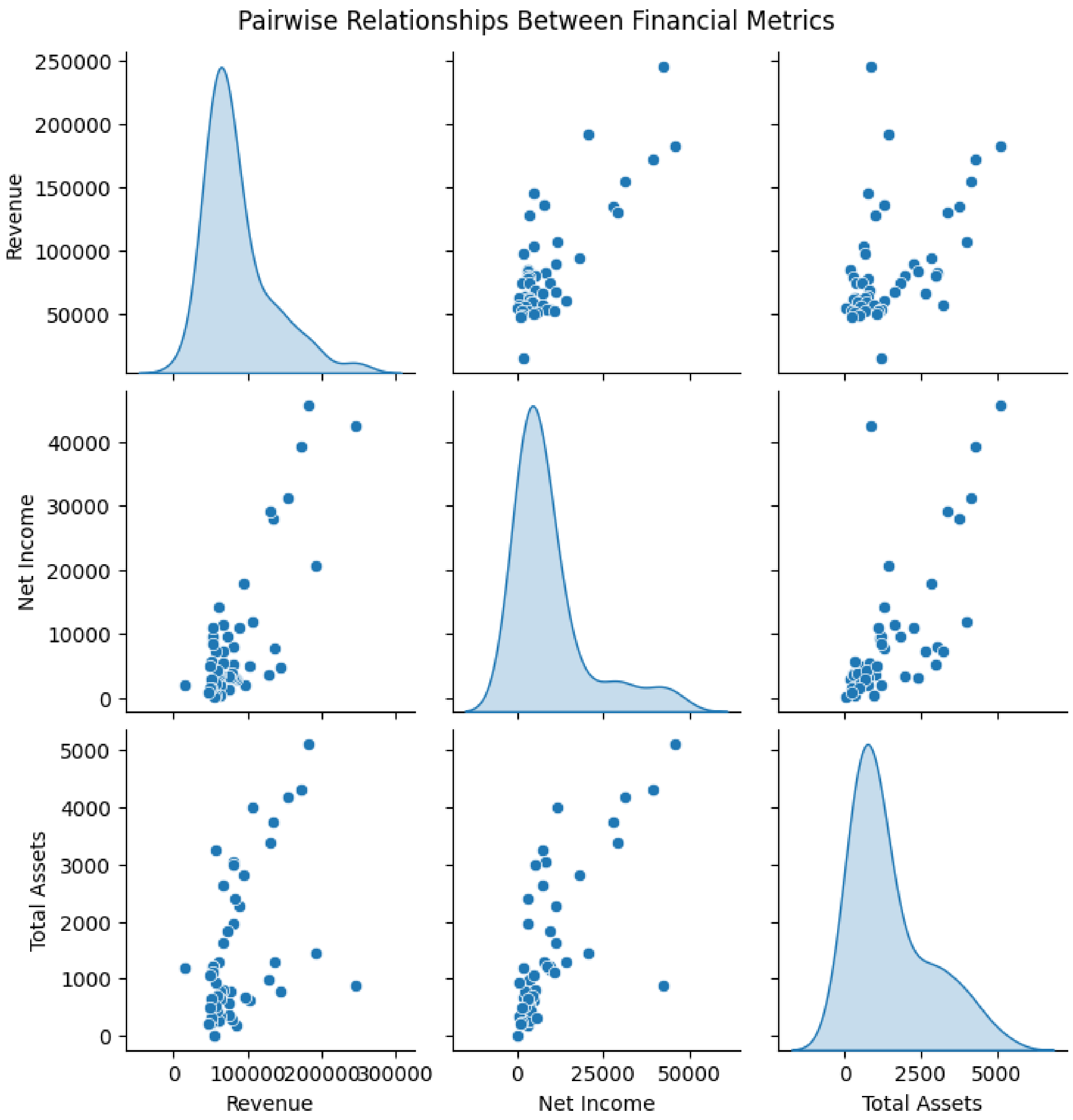
```
# Revenue vs Net Income
plt.figure(figsize=(12, 8))
sns.scatterplot(
    x=df['Revenue in (USD Million)'],
    y=df['Net Income in (USD Millions)'],
    color='#3498DB',
    s=100,
    edgecolor='black',
    alpha=0.7
)
plt.title('Revenue vs Net Income (USD Million)', fontsize=16,
fontweight='bold', color='#2980B9')
plt.xlabel('Revenue (in USD Million)', fontsize=14, color='#1F618D')
plt.ylabel('Net Income (in USD Million)', fontsize=14,
color='#1F618D')
plt.xticks(fontsize=12, color='#1F618D')
plt.yticks(fontsize=12, color='#1F618D')
plt.grid(True, linestyle='--', alpha=0.7, color='gray')
plt.tight_layout()
plt.show()

# Revenue vs Total Assets
plt.figure(figsize=(12, 8))
sns.scatterplot(
    x=df['Revenue in (USD Million)'],
    y=df['Total Assest in (USD Millions)'],
    color='#E74C3C',
    s=100,
    edgecolor='black',
    alpha=0.7
)
plt.title('Revenue vs Total Assets (USD Million)', fontsize=16,
fontweight='bold', color='#C0392B')
plt.xlabel('Revenue (in USD Million)', fontsize=14, color='#C0392B')
plt.ylabel('Total Assets (in USD Million)', fontsize=14,
color='#C0392B')
plt.xticks(fontsize=12, color='#C0392B')
plt.yticks(fontsize=12, color='#C0392B')
plt.grid(True, linestyle='--', alpha=0.7, color='gray')
plt.tight_layout()
plt.show()
```



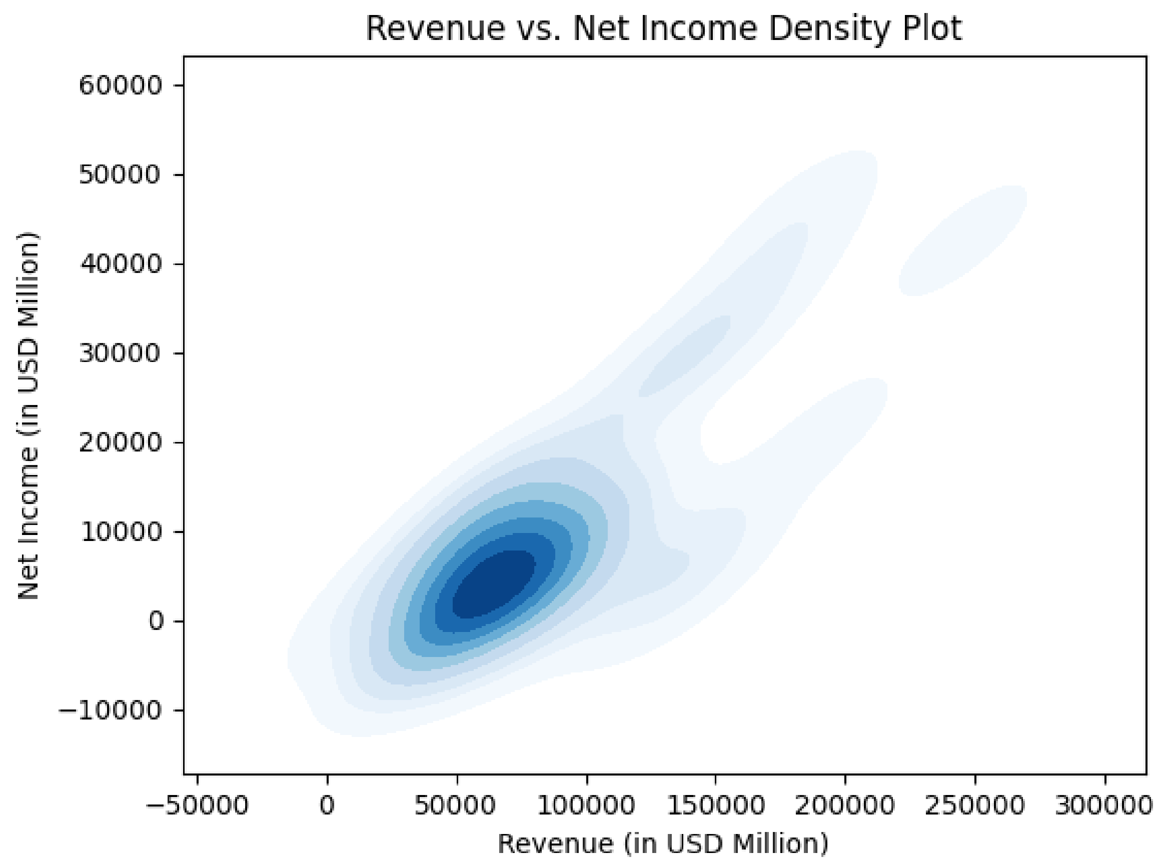


```
# number of companies per country (Headquarters)
hq_counts = df['Headquarters'].value_counts().reset_index()
hq_counts.columns = ['Country', 'Company Count']

fig = px.choropleth(
    hq_counts,
    locations='Country',
    locationmode='country names',
    color='Company Count',
    hover_name='Country',
    color_continuous_scale='Viridis',
    labels={'Company Count': 'Number of Companies'},
    title='Distribution of Companies Across Countries'
)

fig.update_layout(
    geo=dict(showcoastlines=True, coastlinecolor='Black'),
    title_font=dict(size=20, color='RoyalBlue'),
    geo_scope='world',  # limit map scope to the world
    coloraxis_colorbar_title='Number of Companies'
)

fig.show()
fig.show(renderer='iframe')
```

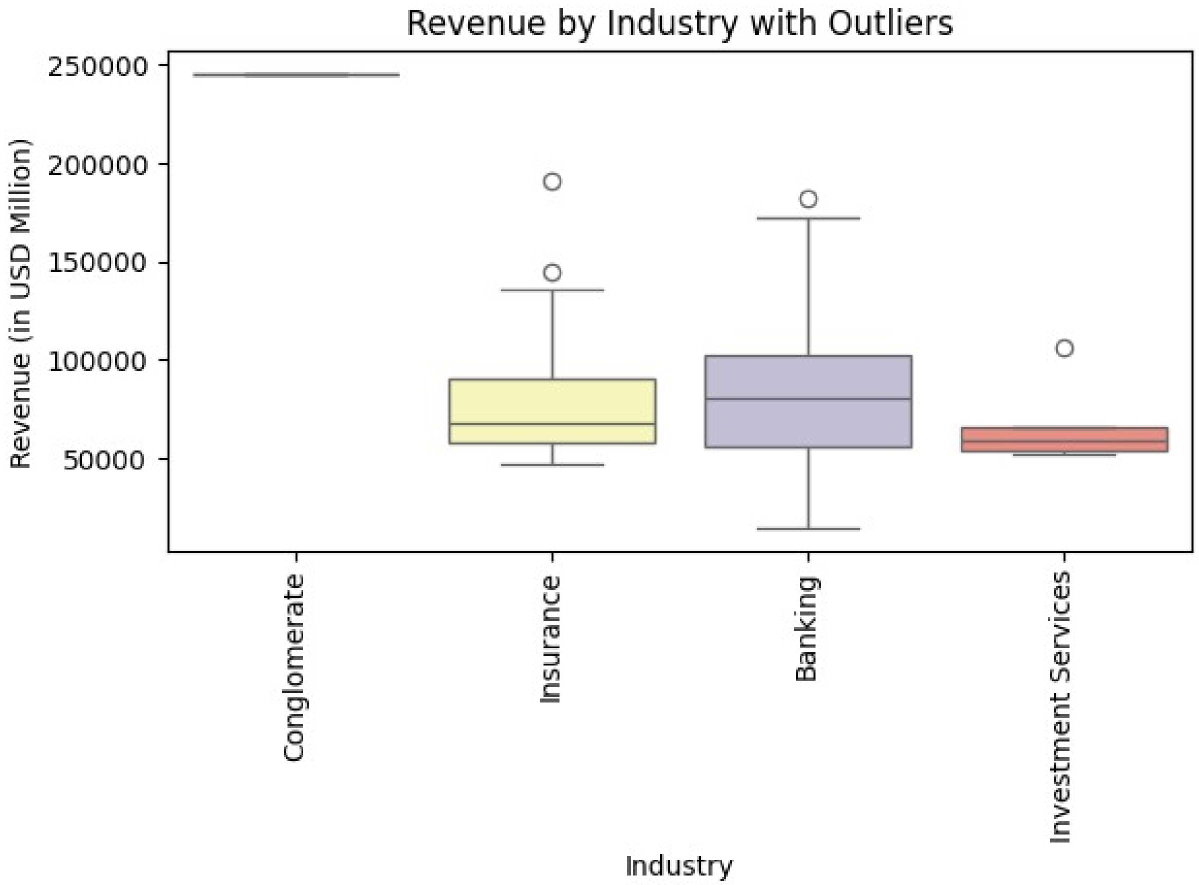
```
fig = px.bar(df, x='Company', y='Revenue in (USD Million)',
color='Industry', title='Company Revenue by Industry')
fig.show()

unique_counts = df.nunique()
print("\nUnique values per column:\n", unique_counts)


Unique values per column:
 Rank                                 50
Company                              50
Industry                              4
Revenue in (USD Million)             50
Net Income in (USD Millions)         49
Total Assest in (USD Millions)       50
Headquarters                         11
dtype: int64

df.rename(columns={
    'Revenue in (USD Million)': 'Revenue',
    'Net Income in (USD Millions)': 'Net Income',
    'Total Assest in (USD Millions)': 'Total Assets'
}, inplace=True)

# Pairplot to visualize pairwise relationships
sns.pairplot(df[['Revenue', 'Net Income', 'Total Assets']],
diag_kind='kde')
plt.suptitle('Pairwise Relationships Between Financial Metrics',
y=1.02)
plt.show()
```



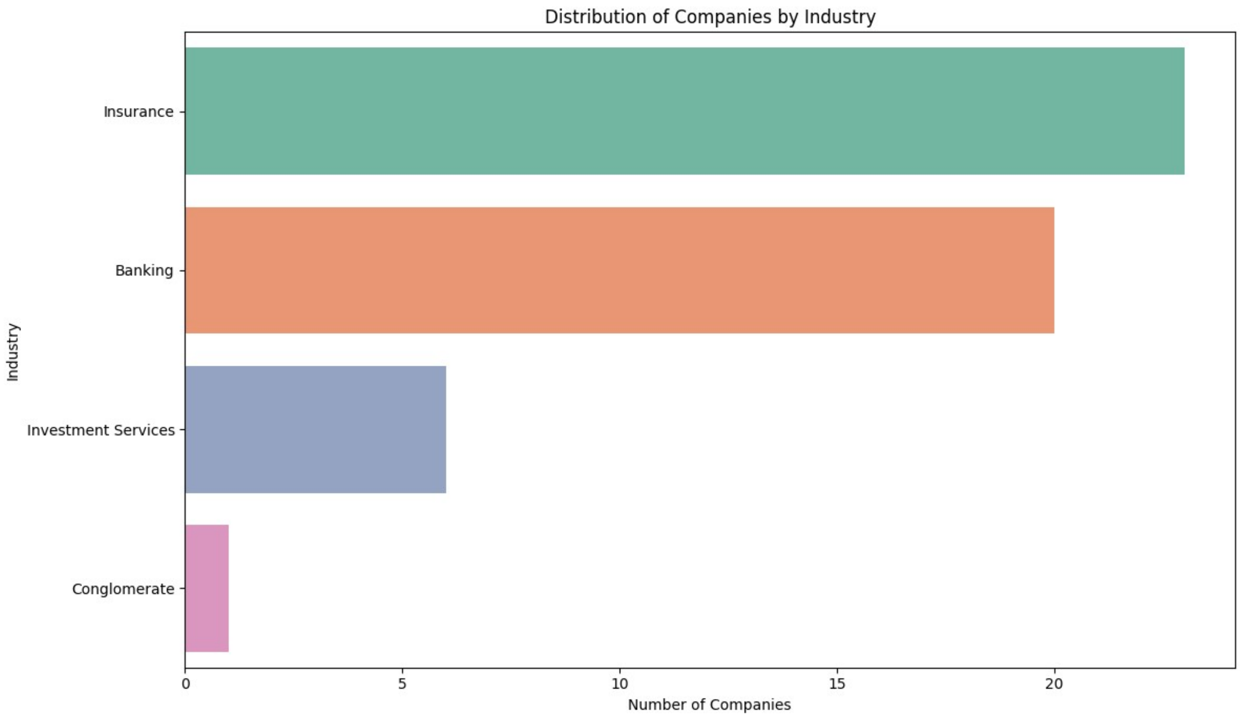
```
sns.kdeplot(data=df, x='Revenue', y='Net Income', fill=True,
cmap='Blues')
plt.title('Revenue vs. Net Income Density Plot')
plt.xlabel('Revenue (in USD Million)')
plt.ylabel('Net Income (in USD Million)')
plt.show()
```



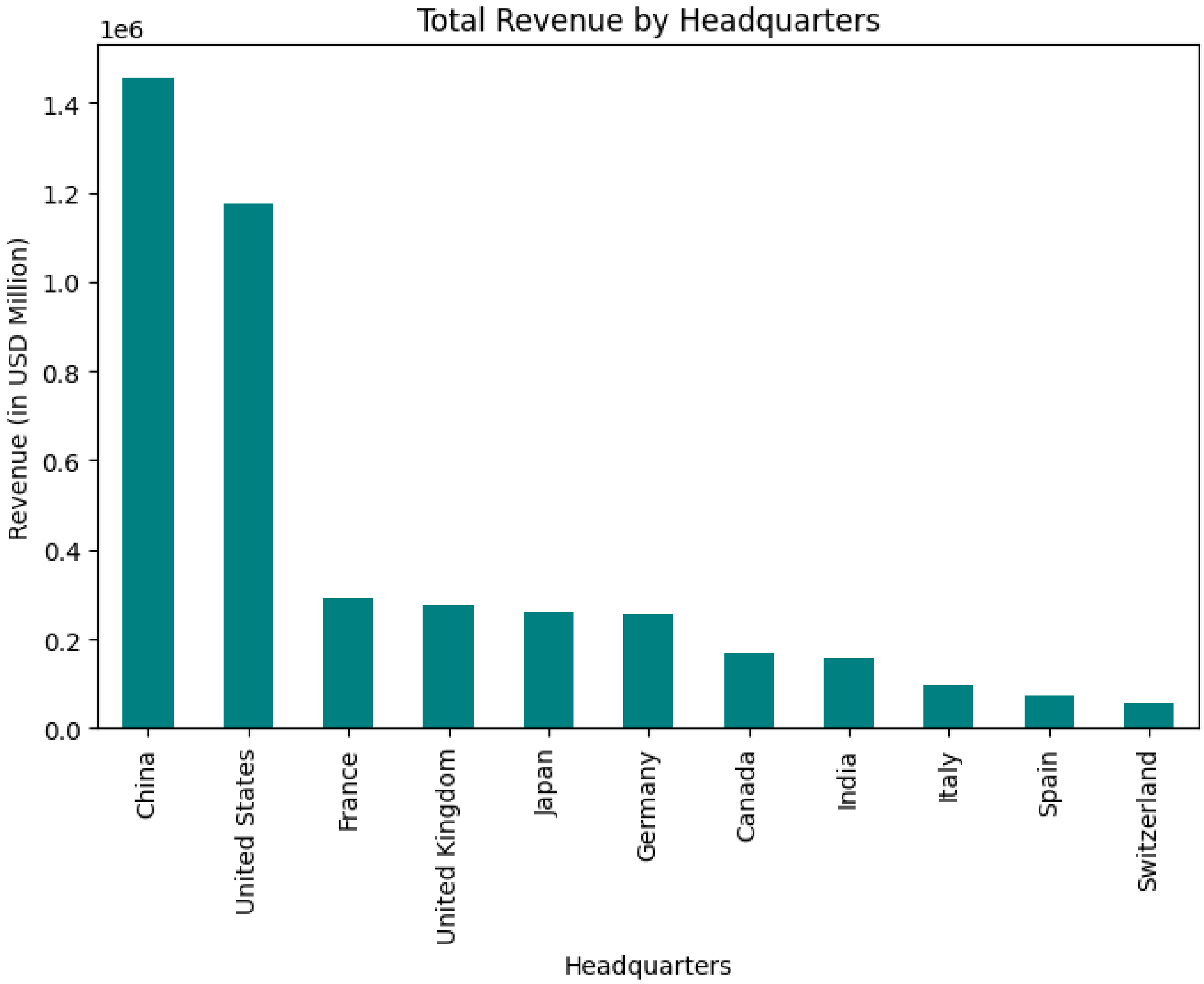
```
sns.boxplot(data=df, x='Industry', y='Revenue', palette='Set3')
plt.xticks(rotation=90)
plt.title('Revenue by Industry with Outliers')
plt.xlabel('Industry')
plt.ylabel('Revenue (in USD Million)')
plt.tight_layout()  # Adjusts the plot to fit the rotated labels
plt.show()
```



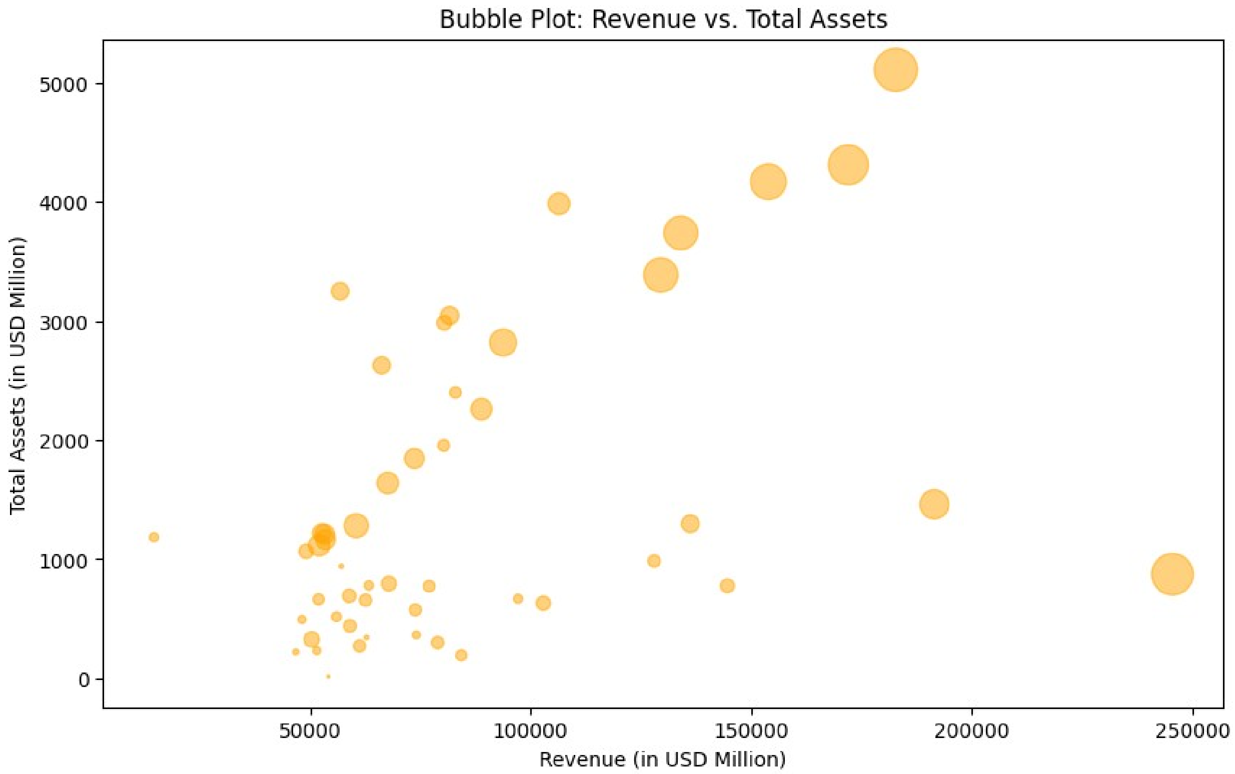
```
plt.figure(figsize=(12, 7))  # Increased size for better visualization
sns.countplot(data=df, y='Industry',
order=df['Industry'].value_counts().index, palette='Set2')
plt.title('Distribution of Companies by Industry')
plt.xlabel('Number of Companies')
plt.ylabel('Industry')
plt.tight_layout()  # Adjusts layout to prevent clipping
plt.show()
```



```
revenue_by_region = df.groupby('Headquarters')
['Revenue'].sum().sort_values(ascending=False)

# Plotting the bar chart
revenue_by_region.plot(kind='bar', color='teal', figsize=(8, 5))

# Adding title and labels
plt.title('Total Revenue by Headquarters')
plt.xlabel('Headquarters')
plt.ylabel('Revenue (in USD Million)')

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



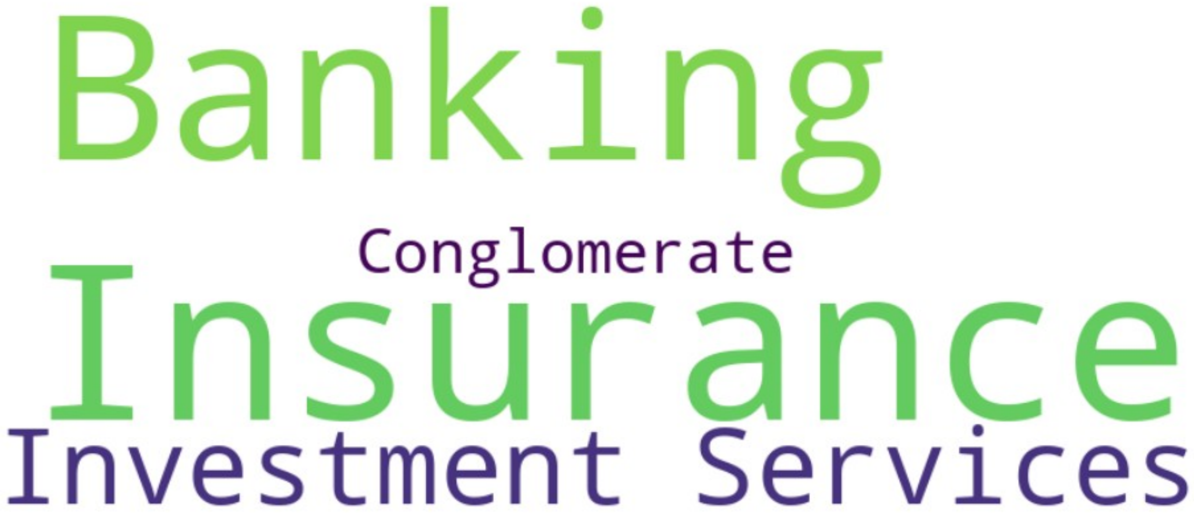
```
# Convert the necessary columns to numeric, in case they are not already
df['Revenue'] = pd.to_numeric(df['Revenue'], errors='coerce')
df['Total Assets'] = pd.to_numeric(df['Total Assets'], errors='coerce')
df['Net Income'] = pd.to_numeric(df['Net Income'], errors='coerce')

# Creating the bubble plot
plt.figure(figsize=(10, 6))
plt.scatter(df['Revenue'], df['Total Assets'], s=df['Net Income']/100, alpha=0.5, c='orange')

# Adding title and labels
plt.title('Bubble Plot: Revenue vs. Total Assets')
plt.xlabel('Revenue (in USD Million)')
plt.ylabel('Total Assets (in USD Million)')

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



```python
if 'Employees' in df.columns:
    # Ensure no division by zero errors by replacing 0 employees with
NaN
    df['Employees'] = df['Employees'].replace(0, pd.NA)

    # Compute the Revenue per Employee
    df['Revenue per Employee'] = df['Revenue'] / df['Employees']

    # Display the top 5 companies by Revenue per Employee
    print("\nTop 5 Companies by Revenue per Employee:")
    print(df[['Company', 'Revenue', 'Employees', 'Revenue per
Employee']].sort_values(by='Revenue per Employee',
ascending=False).head())

from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Make sure the 'Industry' column does not contain null or non-string
values
df['Industry'] = df['Industry'].fillna('')  # Replace NaN values with
an empty string

# Generate the word cloud
plt.figure(figsize=(12, 6))
wordcloud = WordCloud(width=800, height=400,
background_color='white').generate(' '.join(df['Industry']))
```

```python
# Display the word cloud
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')  # Hide the axis
plt.title('Word Cloud of Industries')
plt.show()
```

Word Cloud of Industries


Banking
Conglomerate
Insurance
Investment Services