

# Bank Loans Analysis

## Table of contents

<b>1</b>	<b>Import the required libraries</b>	<b>2</b>
<b>2</b>	<b>EDA &amp; Data Cleaning</b>	<b>2</b>
2.1	Import the dataset from MS SQL . . . . .	2
2.2	Explore the dataset shape & columns type . . . . .	2
2.3	Total rows and columns . . . . .	3
2.4	Change the date column type . . . . .	3
2.5	Deal with null values . . . . .	3
2.6	Add Columns . . . . .	4
2.7	Explore the unique values . . . . .	4
2.7.1	Companies . . . . .	4
2.7.2	Date Range . . . . .	4
2.8	Save the cleaned data . . . . .	4
<b>3</b>	<b>Data Analysis</b>	<b>5</b>
3.1	Import the new dataset . . . . .	5
3.2	How much amount the companies on R&D . . . . .	5
3.3	Revenue Earned by the companies . . . . .	6
3.4	Datewise Impact on the stock . . . . .	7
3.5	Company-wise stock impact over time . . . . .	9
3.6	Expenditure vs Revenue year-by-year . . . . .	10
3.7	AI Revenue growth of the companies . . . . .	11
3.7.1	Correlation Heatmap for AI Market data . . . . .	11
3.7.2	Growth analysis . . . . .	12

# 1 Import the required libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sqlalchemy import create_engine
```

## 2 EDA & Data Cleaning

### 2.1 Import the dataset from MS SQL

```
engine = create_engine(
    "mssql+pyodbc://localhost\\SQLEXPRESS/ai_fin_market"
    "?driver=ODBC+Driver+17+for+SQL+Server"
    "&trusted_connection=yes"
)
ai_market = pd.read_sql("SELECT * FROM dbo.ai_data", engine)
ai_market.head()
```

	Date	Company	R_D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Revenue_Growth	Event
0	2015-01-01	OpenAI	5.92	0.63	-36.820000	None
1	2015-01-02	OpenAI	5.41	1.81	80.589996	None
2	2015-01-03	OpenAI	4.50	0.61	-38.880001	None
3	2015-01-04	OpenAI	5.45	0.95	-5.340000	None
4	2015-01-05	OpenAI	3.40	1.48	48.450001	None

### 2.2 Explore the dataset shape & columns type

```
for i, (col, ty) in enumerate(ai_market.dtypes.items(), start=1):
    if i < 10:
        print(f"{i:>2}. {col:<26} {ty}")
    else:
        print(f"{i:>2}. {col:<25} {ty}")
```

1. Date	object
2. Company	object
3. R_D_Spending_USD_Mn	float64
4. AI_Revenue_USD_Mn	float64
5. AI_Revenue_Growth	float64
6. Event	object
7. Stock_Impact	float64

## 2.3 Total rows and columns

```
rows, cols = ai_market.shape
print(f"Rows: {rows}\nColumns: {cols}")
```

Rows: 10959

Columns: 7

## 2.4 Change the date column type

```
ai_market['Date'] = pd.to_datetime(ai_market['Date'])
ai_market.head()
```

	Date	Company	R_D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Revenue_Growth	Event
0	2015-01-01	OpenAI	5.92	0.63	-36.820000	None
1	2015-01-02	OpenAI	5.41	1.81	80.589996	None
2	2015-01-03	OpenAI	4.50	0.61	-38.880001	None
3	2015-01-04	OpenAI	5.45	0.95	-5.340000	None
4	2015-01-05	OpenAI	3.40	1.48	48.450001	None

## 2.5 Deal with null values

```
ai_market.isna().sum()
```

```
Date          0
Company        0
R_D_Spending_USD_Mn  0
AI_Revenue_USD_Mn  0
AI_Revenue_Growth  0
Event         10726
Stock_Impact    0
dtype: int64
```

```
ai_market['Event'].fillna('Unknown', inplace=True)
```

## 2.6 Add Columns

```
ai_market['Year'] = ai_market['Date'].dt.year
ai_market['Month'] = ai_market['Date'].dt.month
```

## 2.7 Explore the unique values

### 2.7.1 Companies

```
for i, comp in enumerate(ai_market['Company'].unique(), start=1):
    print(f"{i}. {comp}\n")
```

1. OpenAI
2. Google
3. Meta

### 2.7.2 Date Range

```
start_date = ai_market['Date'].min()
end_date = ai_market['Date'].max()

print(f"start date: {start_date}\nend_date: {end_date}")
```

```
start date: 2015-01-01 00:00:00
end_date: 2024-12-31 00:00:00
```

## 2.8 Save the cleaned data

```
ai_market.to_sql(
    "ai_processed",
    con=engine,
    schema="dbo",
    if_exists="replace",
    index=False
)
```

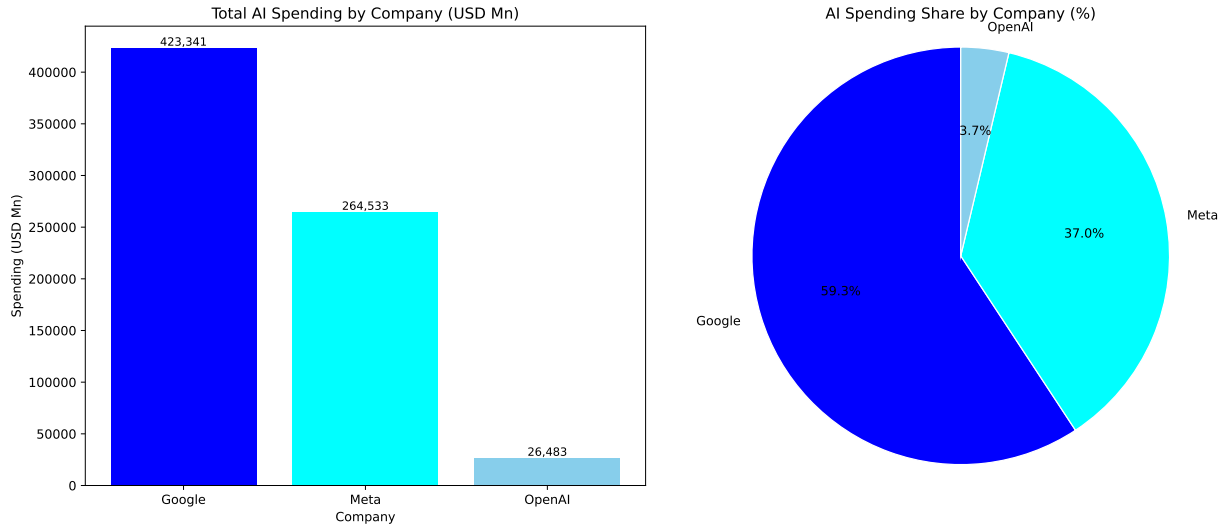
## 3 Data Analysis

### 3.1 Import the new dataset

```
ai_processed = pd.read_sql("SELECT * FROM dbo.ai_processed", engine)
ai = ai_processed.copy()
```

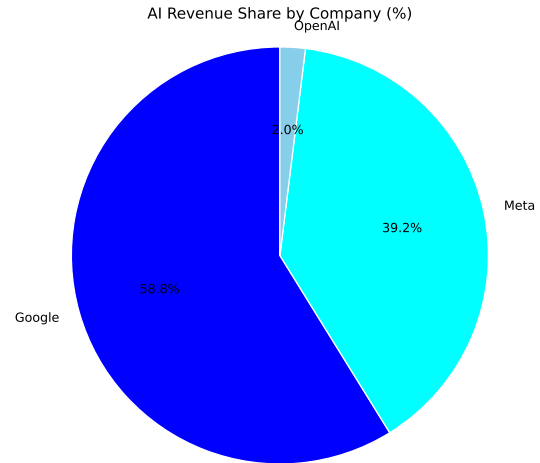
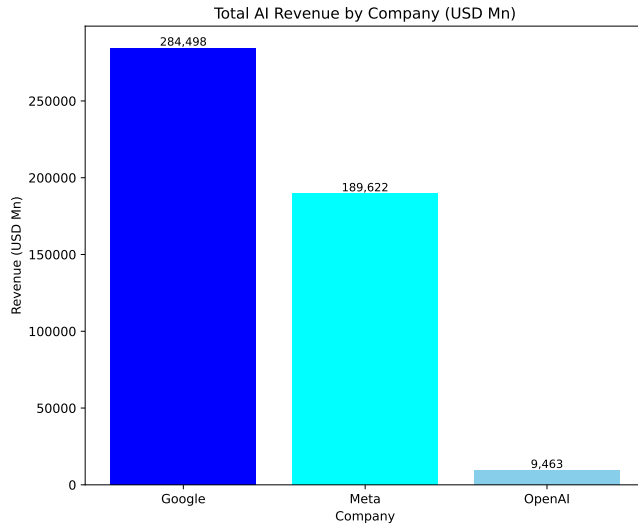
### 3.2 How much amount the companies on R&D

```
RD = ai.groupby('Company')['R_D_Spending_USD_Mn'].sum().sort_values(ascending=False)
fig, axes = plt.subplots(1, 2, figsize=(14, 6))
# (Bar Chart)
axes[0].bar(
    RD.index,
    RD.values,
    color=['blue', 'cyan', 'skyblue']
)
axes[0].set_title('Total AI Spending by Company (USD Mn)')
axes[0].set_xlabel('Company')
axes[0].set_ylabel('Spending (USD Mn)')
for i, v in enumerate(RD.values):
    axes[0].text(i, v, f'{v:,.0f}', ha='center', va='bottom', fontsize=9)
# (Pie Chart)
axes[1].pie(
    RD.values,
    labels=RD.index,
    autopct='%1.1f%%',
    startangle=90,
    colors=['blue', 'cyan', 'skyblue'],
    wedgeprops={'edgecolor': 'white'}
)
axes[1].set_title('AI Spending Share by Company (%)')
axes[1].axis('equal')
plt.tight_layout()
plt.show()
```



### 3.3 Revenue Earned by the companies

```
RD = ai.groupby('Company')['AI_Revenue_USD_Mn'].sum().sort_values(ascending=False)
fig, axes = plt.subplots(1, 2, figsize=(14, 6))
# (Bar Chart)
axes[0].bar(
    RD.index,
    RD.values,
    color=['blue', 'cyan', 'skyblue']
)
axes[0].set_title('Total AI Revenue by Company (USD Mn)')
axes[0].set_xlabel('Company')
axes[0].set_ylabel('Revenue (USD Mn)')
for i, v in enumerate(RD.values):
    axes[0].text(i, v, f'{v:,.0f}', ha='center', va='bottom', fontsize=9)
# (Pie Chart)
axes[1].pie(
    RD.values,
    labels=RD.index,
    autopct='%1.1f%%',
    startangle=90,
    colors=['blue', 'cyan', 'skyblue'],
    wedgeprops={'edgecolor': 'white'}
)
axes[1].set_title('AI Revenue Share by Company (%)')
axes[1].axis('equal')
plt.tight_layout()
plt.show()
```



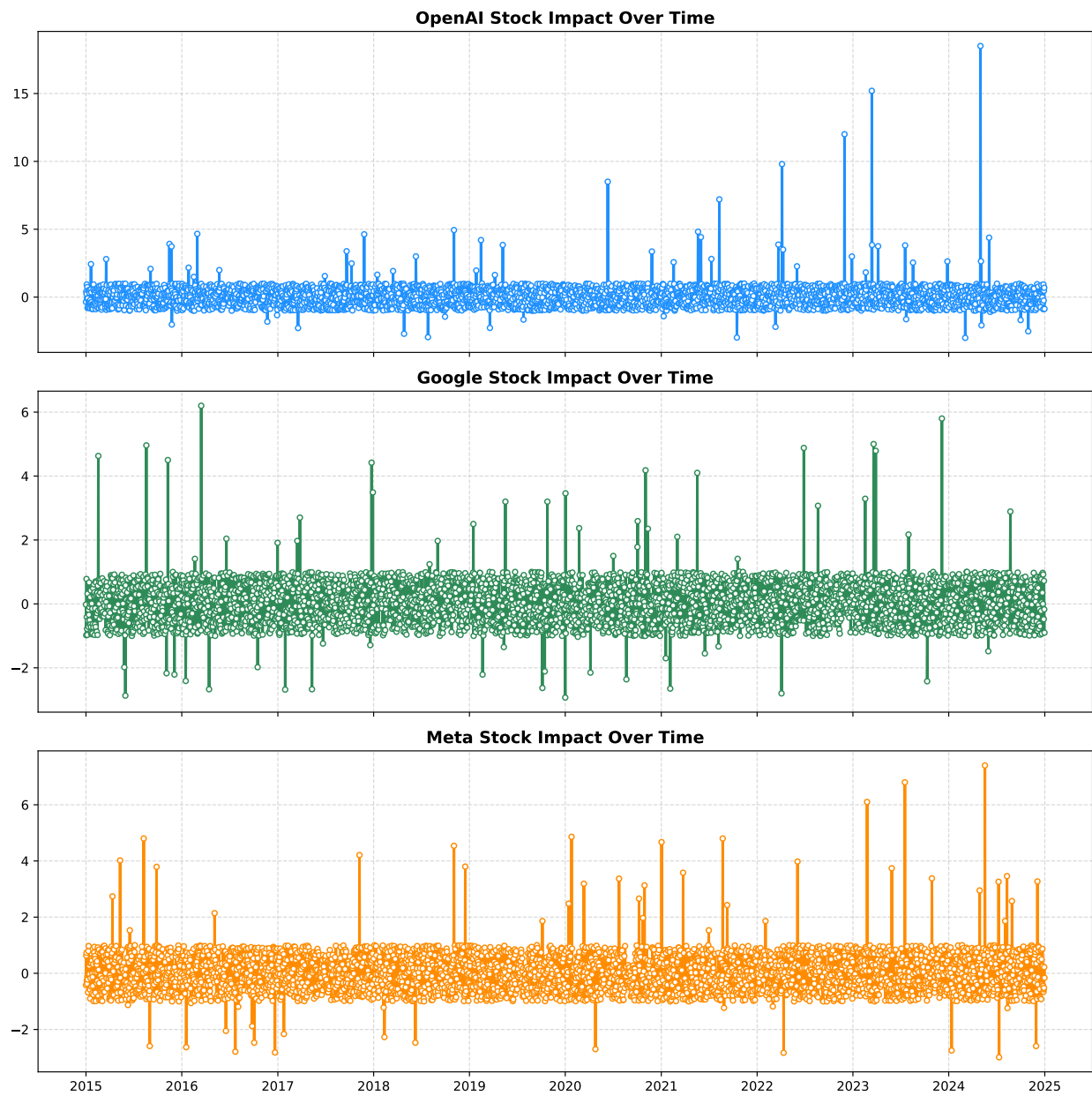
### 3.4 Datewise Impact on the stock

```
ai_openAi = ai[ai['Company'] == 'OpenAI']
ai_google = ai[ai['Company'] == 'Google']
ai_meta = ai[ai['Company'] == 'Meta']
fig, axes = plt.subplots(3, 1, figsize=(12, 12), sharex=True)

# (OpenAI)
axes[0].plot(ai_openAi['Date'], ai_openAi['Stock_Impact'], color='dodgerblue', linewidth=2,
             marker='o', markersize=4, markerfacecolor='white')
axes[0].fill_between(ai_openAi['Date'], ai_openAi['Stock_Impact'], color='lightblue', alpha=0.5)
axes[0].set_title('OpenAI Stock Impact Over Time', fontsize=13, weight='bold')
axes[0].grid(True, linestyle='--', alpha=0.5)

# (Google)
axes[1].plot(ai_google['Date'], ai_google['Stock_Impact'], color='seagreen',
             linewidth=2, marker='o', markersize=4, markerfacecolor='white')
axes[1].fill_between(ai_google['Date'], ai_google['Stock_Impact'], color='palegreen', alpha=0.3)
axes[1].set_title('Google Stock Impact Over Time', fontsize=13, weight='bold')
axes[1].grid(True, linestyle='--', alpha=0.5)

# (Meta)
axes[2].plot(ai_meta['Date'], ai_meta['Stock_Impact'], color='darkorange',
             linewidth=2, marker='o', markersize=4, markerfacecolor='white')
axes[2].fill_between(ai_meta['Date'], ai_meta['Stock_Impact'], color='moccasin', alpha=0.3)
axes[2].set_title('Meta Stock Impact Over Time', fontsize=13, weight='bold')
axes[2].grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```





### 3.5 Company-wise stock impact over time

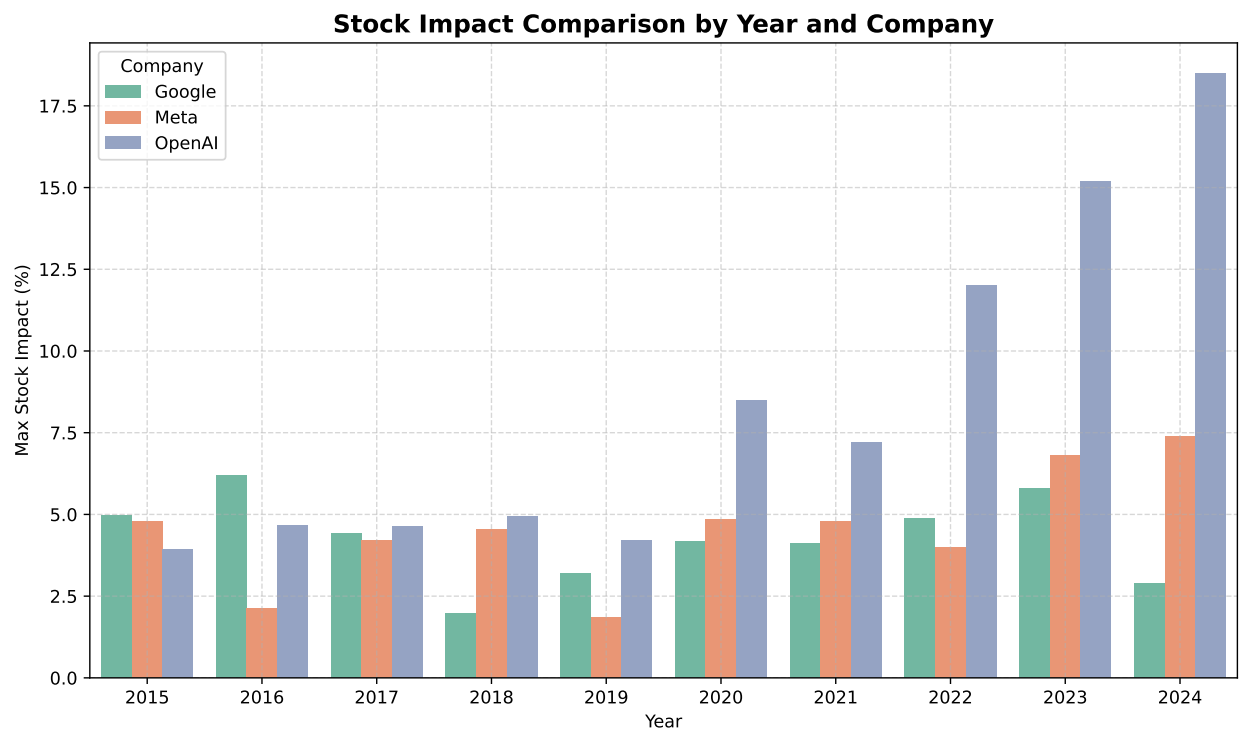
```
Impact = ai.groupby(['Year', 'Company'])['Stock_Impact'].max().reset_index()

plt.figure(figsize=(10,6))

sns.barplot(
    data = Impact,
    x = 'Year',
    y = 'Stock_Impact',
    hue = 'Company',
    palette = 'Set2'
)

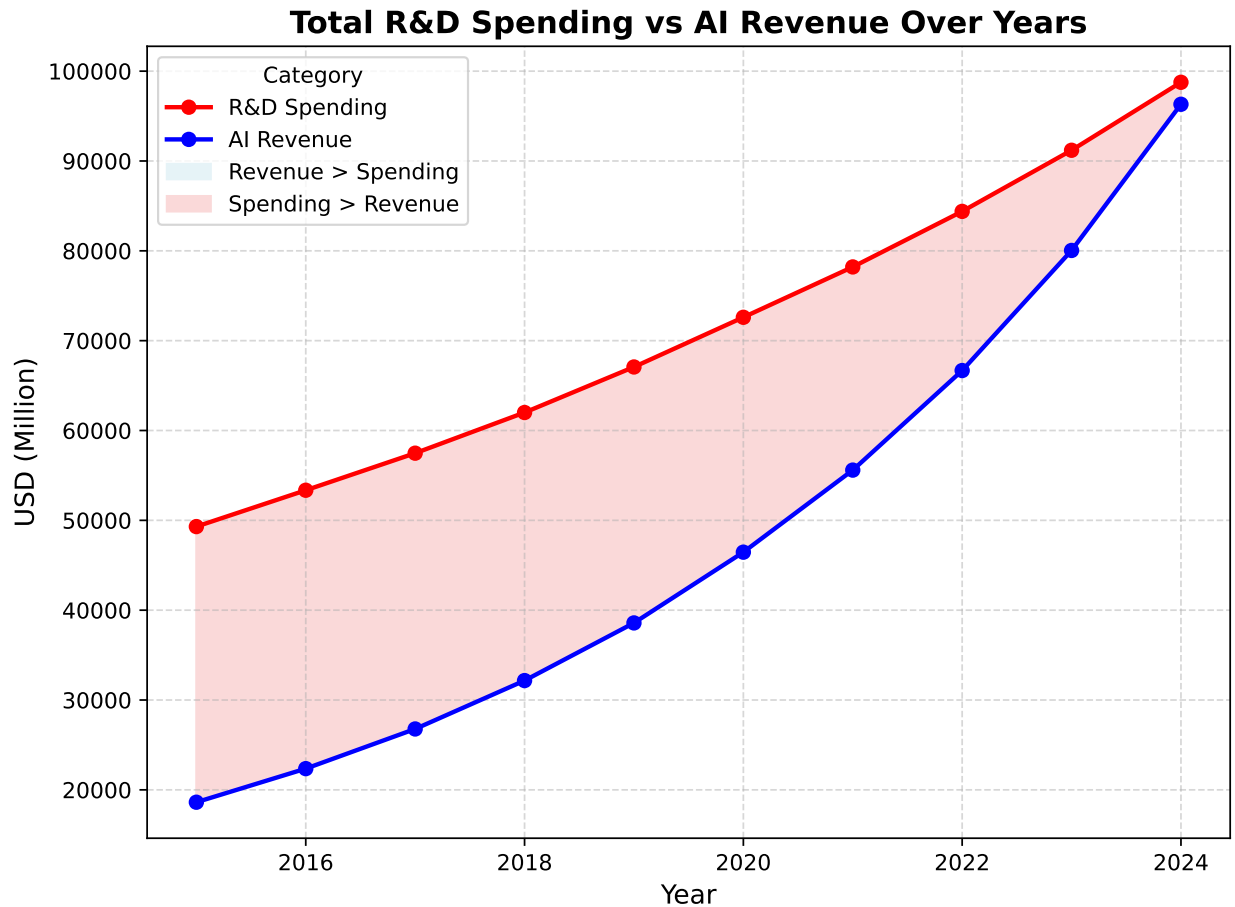
plt.title('Stock Impact Comparison by Year and Company', fontsize=14, weight='bold')
plt.xlabel('Year')
plt.ylabel('Max Stock Impact (%)')

plt.legend(title='Company')
plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```



### 3.6 Expenditure vs Revenue year-by-year

```
spend = ai.groupby('Year')['R_D_Spending_USD_Mn'].sum()
rev = ai.groupby('Year')['AI_Revenue_USD_Mn'].sum()
plt.figure(figsize=(8,6))
plt.plot(
    spend.index,
    spend.values,
    color='red',
    marker='o',
    linewidth=2,
    label='R&D Spending'
)
plt.plot(
    rev.index,
    rev.values,
    color = 'blue',
    marker = 'o',
    linewidth = 2,
    label = 'AI Revenue'
)
plt.fill_between(
    spend.index,
    spend.values,
    rev.values,
    where=(rev.values >= spend.values),
    color='lightblue',
    alpha=0.3,
    label='Revenue > Spending'
)
plt.fill_between(
    spend.index,
    spend.values,
    rev.values,
    where=(rev.values < spend.values),
    color='lightcoral',
    alpha=0.3,
    label='Spending > Revenue'
)
plt.title('Total R&D Spending vs AI Revenue Over Years', fontsize=14, weight='bold')
plt.xlabel('Year', fontsize=12)
plt.ylabel('USD (Million)', fontsize=12)
plt.grid(True, linestyle='--', alpha=0.5)
plt.legend(title='Category')
plt.tight_layout()
plt.show()
```

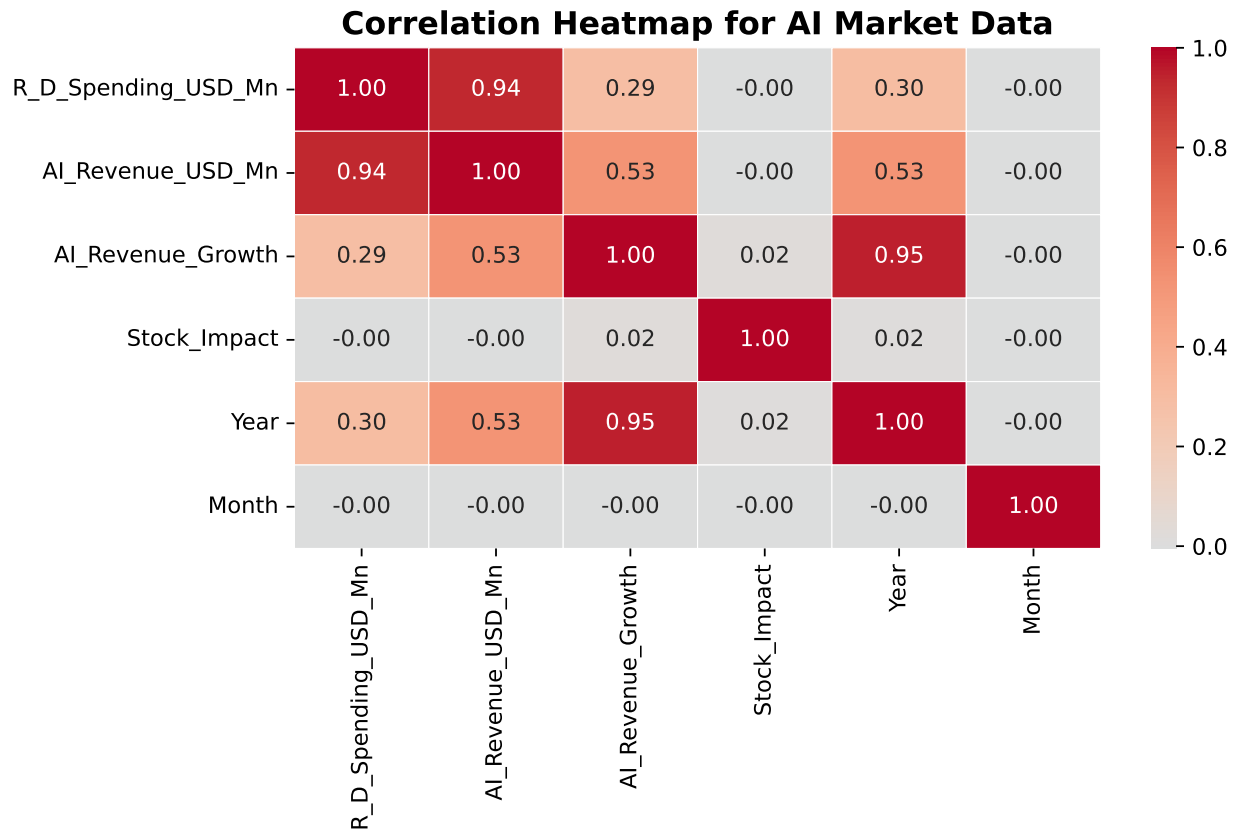


## 3.7 AI Revenue growth of the companies

### 3.7.1 Correlation Heatmap for AI Market data

```
# sns.heatmap(ai.corr(numeric_only=True))

plt.figure(figsize=(8,4))
sns.heatmap(
    ai.corr(numeric_only=True),
    annot = True,
    fmt = ".2f",
    cmap = 'coolwarm',
    center = 0,
    linewidths = 0.5
)
plt.title('Correlation Heatmap for AI Market Data', fontsize=14, weight='bold')
plt.show()
```



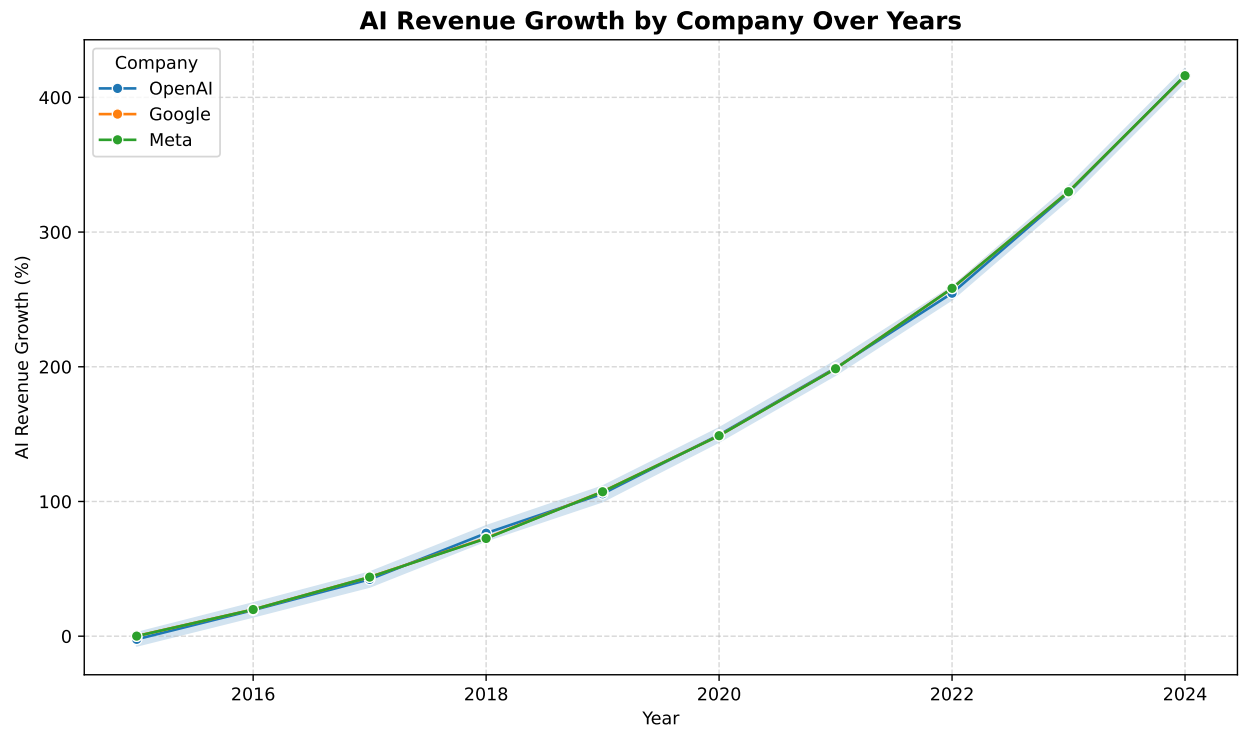
### 3.7.2 Growth analysis

```
plt.figure(figsize=(10,6))

sns.lineplot(
    data = ai,
    x = 'Year',
    y = 'AI_Revenue_Growth',
    hue = 'Company',
    marker = 'o'
)

plt.title('AI Revenue Growth by Company Over Years', fontsize=14, weight='bold')
plt.xlabel('Year')
plt.ylabel('AI Revenue Growth (%)')

plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```



```
sns.barplot(data=ai, x='Year', y='AI_Revenue_Growth', hue='Company')
```

