

AI FINANCIAL MARKET ANALYSIS





DATA ANALYSIS USING PYTHON - AI FINANCIAL MARKET

```
In [44]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

Import Raw Data

```
In [4]: df = pd.read_csv("C:/Users/Dell Laptop/Downloads/ai_financial_market_daily_rea
```

Sample Data

```
In [5]: df.head(20)
```

Out[5]:

	Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Reve
0	2015-01-01	OpenAI	5.92	0.63	
1	2015-01-02	OpenAI	5.41	1.81	
2	2015-01-03	OpenAI	4.50	0.61	
3	2015-01-04	OpenAI	5.45	0.95	
4	2015-01-05	OpenAI	3.40	1.48	
5	2015-01-06	OpenAI	5.47	1.08	
6	2015-01-07	OpenAI	6.95	0.67	
7	2015-01-08	OpenAI	5.06	1.43	
8	2015-01-09	OpenAI	5.26	0.33	
9	2015-01-10	OpenAI	2.00	0.74	
10	2015-01-11	OpenAI	5.34	1.88	
11	2015-01-12	OpenAI	6.10	0.33	
12	2015-01-13	OpenAI	3.09	0.74	
13	2015-01-14	OpenAI	5.18	0.72	
14	2015-01-15	OpenAI	6.16	0.95	
15	2015-01-16	OpenAI	4.68	0.88	
16	2015-01-17	OpenAI	4.19	0.61	
17	2015-01-18	OpenAI	3.46	1.16	
18	2015-01-19	OpenAI	4.12	1.76	
19	2015-01-20	OpenAI	4.37	0.91	

In [6]: `df.tail(10)`

Out[6]:		Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_R
	10949	2024-12-22	Meta	101.15	103.02	
	10950	2024-12-23	Meta	102.20	103.05	
	10951	2024-12-24	Meta	99.39	102.14	
	10952	2024-12-25	Meta	100.36	103.17	
	10953	2024-12-26	Meta	101.21	103.46	
	10954	2024-12-27	Meta	100.19	103.54	
	10955	2024-12-28	Meta	99.12	102.37	
	10956	2024-12-29	Meta	98.95	103.11	
	10957	2024-12-30	Meta	100.74	103.21	
	10958	2024-12-31	Meta	100.08	103.41	

Size of Data

```
In [7]: print("size of data:",df.shape)
```

size of data: (10959, 7)

Field info

```
In [8]: df.columns
```

```
Out[8]: Index(['Date', 'Company', 'R&D_Spending_USD_Mn', 'AI_Revenue_USD_Mn',
              'AI_Revenue_Growth_%', 'Event', 'Stock_Impact_%'],
              dtype='object')
```

Data Type

```
In [9]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10959 entries, 0 to 10958
Data columns (total 7 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Date                                10959 non-null  object
1   Company                             10959 non-null  object
2   R&D_Spending_USD_Mn                 10959 non-null  float64
3   AI_Revenue_USD_Mn                   10959 non-null  float64
4   AI_Revenue_Growth_%                 10959 non-null  float64
5   Event                               233 non-null    object
6   Stock_Impact_%                      10959 non-null  float64
dtypes: float64(4), object(3)
memory usage: 599.4+ KB
```

Show the Companies Name

```
In [10]: df['Company'].unique()
```

```
Out[10]: array(['OpenAI', 'Google', 'Meta'], dtype=object)
```

Create a new Column for 'Year' only

```
In [11]: df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
df['Year'] = df['Date'].dt.year
```

How much amount the compamies spent on R & D ?

```
In [12]: print("Company's spending for R & D in $BN")

RD = df.groupby('Company')['R&D_Spending_USD_Mn'].sum()/1000

RD
```

Company's spending for R & D in \$BN

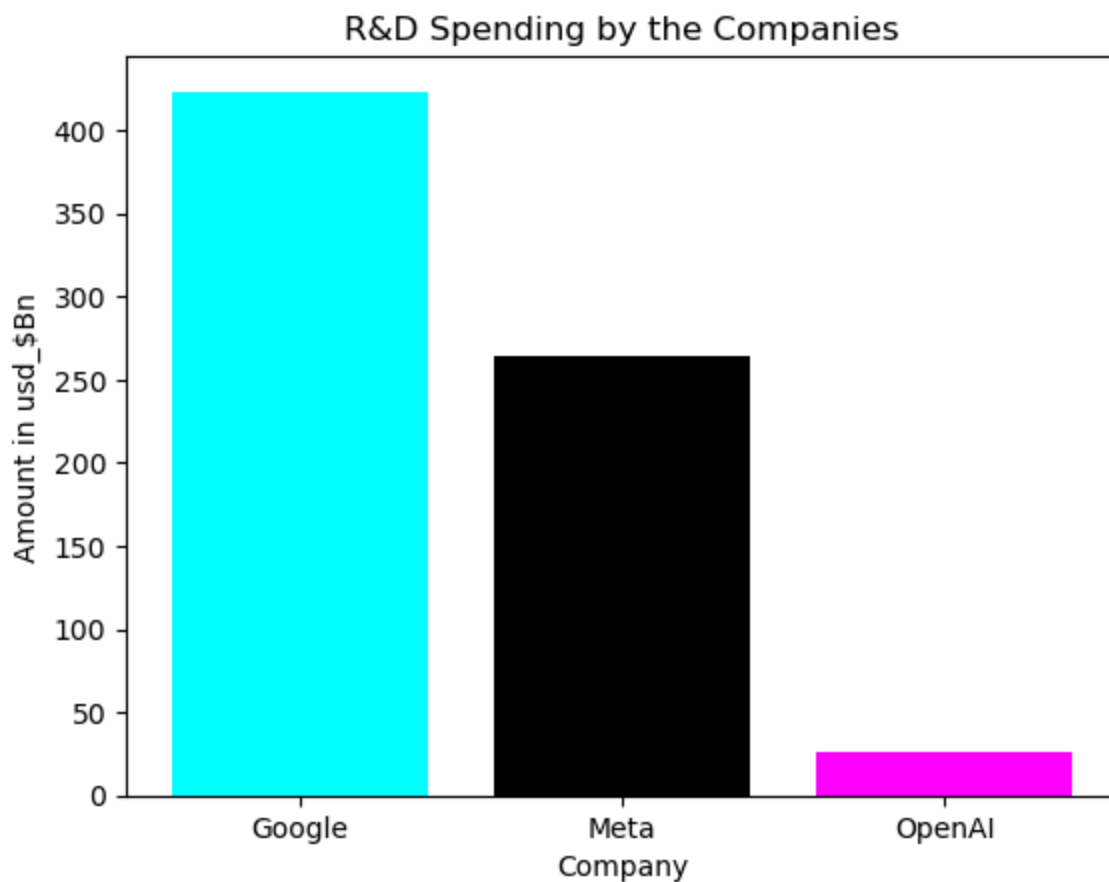
```
Out[12]: Company
Google    423.34114
Meta      264.53307
OpenAI    26.48277
Name: R&D_Spending_USD_Mn, dtype: float64
```

CHARTS REQUIRMENT

Total Amount Spent on R & D

```
In [13]: plt.bar(RD.index, RD.values, color = ['cyan', 'black', 'magenta'])

plt.title( "R&D Spending by the Companies")
plt.xlabel("Company")
plt.ylabel("Amount in usd_$Bn")
plt.show()
```



Total AI Revenue Earned

```
In [14]: print("Company's AI_Revenue_USD_Bn :")

rev = df.groupby('Company')['AI_Revenue_USD_Mn'].sum()/1000

rev
```

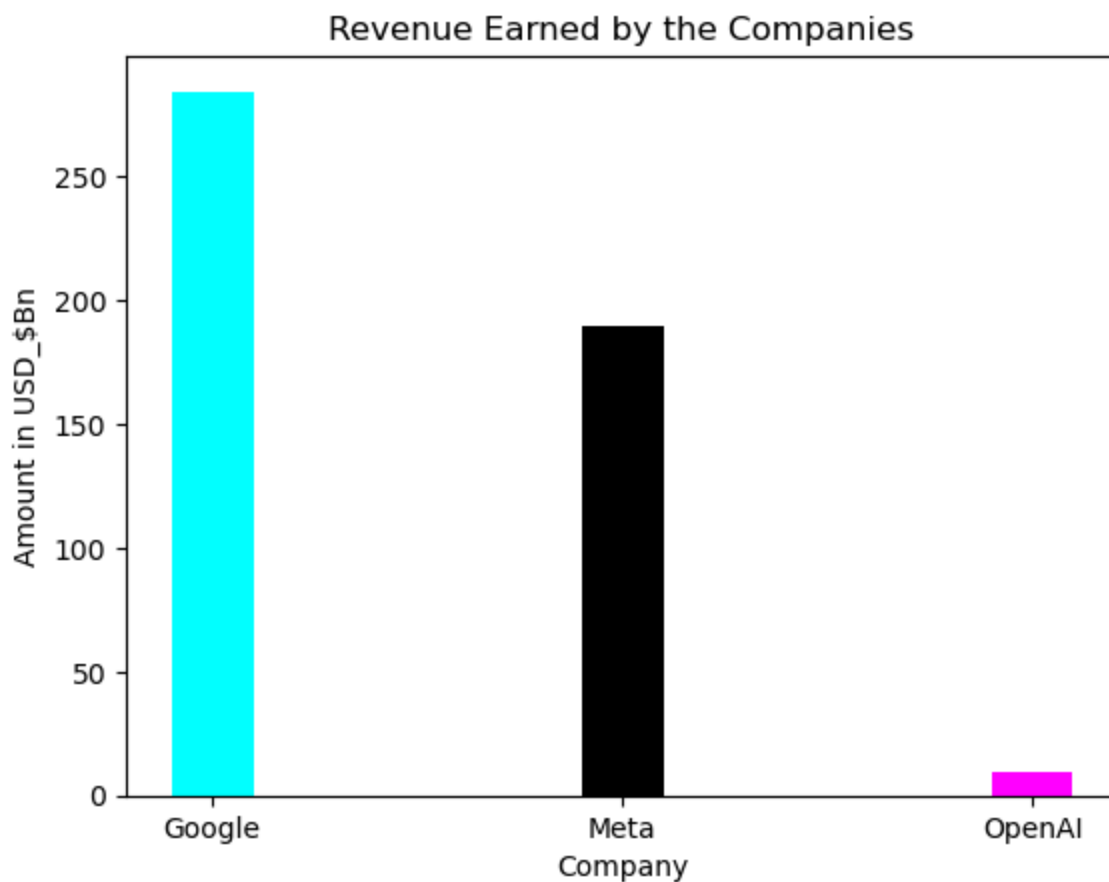
Company's AI_Revenue_USD_Bn :

```
Out[14]: Company
Google    284.49838
Meta      189.62182
OpenAI     9.46289
Name: AI_Revenue_USD_Mn, dtype: float64
```

```
In [15]: plt.bar(rev.index, rev.values, color = ['cyan', 'black', 'magenta'], width = 0.5)

plt.title( "Revenue Earned by the Companies")
plt.xlabel("Company")
plt.ylabel("Amount in USD_$Bn")

plt.show()
```



Comparision Expenditure & Revenue

```
In [16]: plt.figure(figsize = (10,4))

plt.subplot(1,2,1)

plt.bar(RD.index, RD.values, color = ['cyan', 'black', 'magenta'])

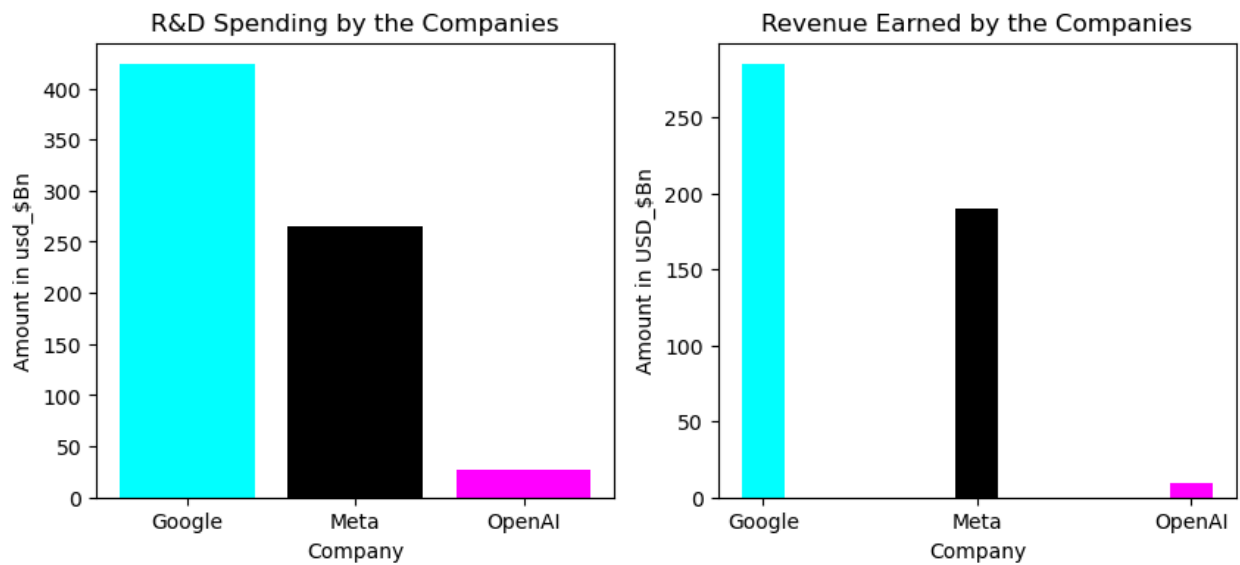
plt.title( "R&D Spending by the Companies")
plt.xlabel("Company")
plt.ylabel("Amount in usd_$Bn")

plt.subplot(1,2,2)

plt.bar(rev.index, rev.values, color = ['cyan', 'black', 'magenta'], width = 0.5)

plt.title( "Revenue Earned by the Companies")
plt.xlabel("Company")
plt.ylabel("Amount in USD_$Bn")

plt.show()
```



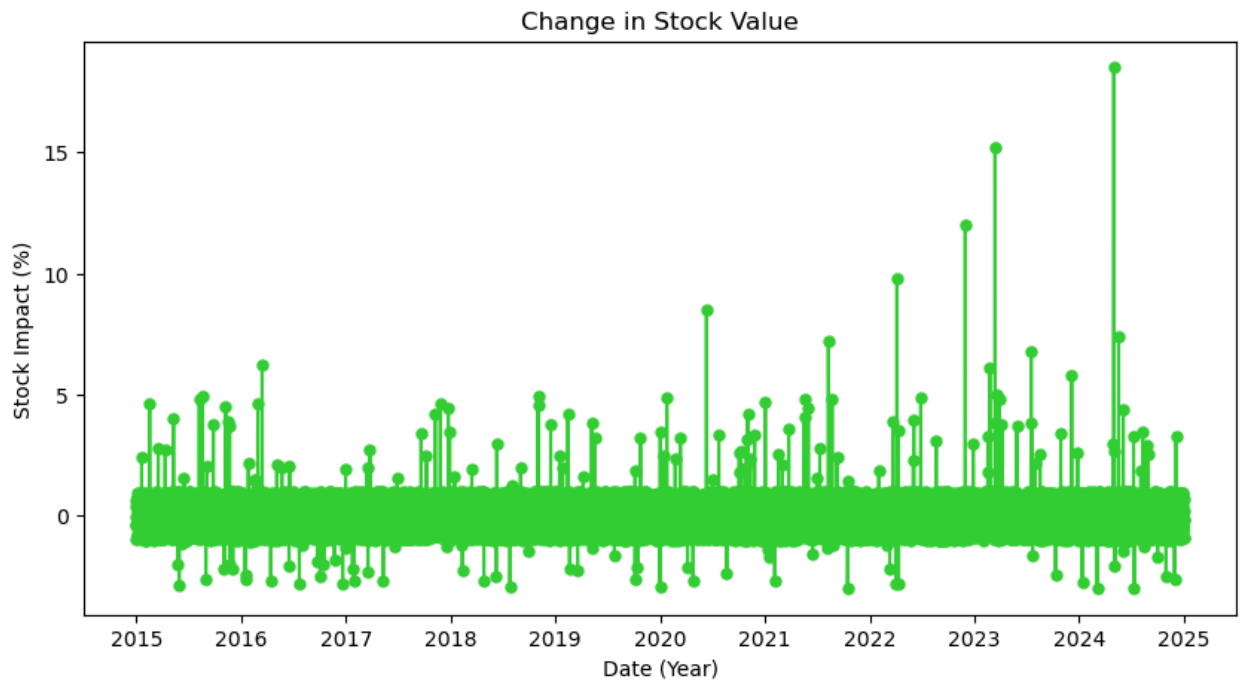
Datewise Impact on the Stock

```
In [17]: plt.figure(figsize=(10,5))

plt.plot(df['Date'], df['Stock_Impact_%'], color='limegreen',marker='o', marke

plt.title("Change in Stock Value")
plt.xlabel("Date (Year)")
plt.ylabel("Stock Impact (%)")

plt.show()
```



Create 3 Separate Dataframes

In [18]: `df.head()`

Out[18]:

	Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Reven
0	2015-01-01	OpenAI	5.92	0.63	
1	2015-01-02	OpenAI	5.41	1.81	
2	2015-01-03	OpenAI	4.50	0.61	
3	2015-01-04	OpenAI	5.45	0.95	
4	2015-01-05	OpenAI	3.40	1.48	

In [19]: `data_openai = df [df['Company'] == 'OpenAI']`
`data_google = df [df['Company'] == 'Google']`
`data_meta = df [df['Company'] == 'Meta']`

In [20]: `data_openai`

Out[20]:

	Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Re
0	2015-01-01	OpenAI	5.92	0.63	
1	2015-01-02	OpenAI	5.41	1.81	
2	2015-01-03	OpenAI	4.50	0.61	
3	2015-01-04	OpenAI	5.45	0.95	
4	2015-01-05	OpenAI	3.40	1.48	
...
3648	2024-12-27	OpenAI	10.06	4.71	
3649	2024-12-28	OpenAI	9.67	5.32	
3650	2024-12-29	OpenAI	9.17	5.46	
3651	2024-12-30	OpenAI	10.36	6.31	
3652	2024-12-31	OpenAI	10.24	4.92	

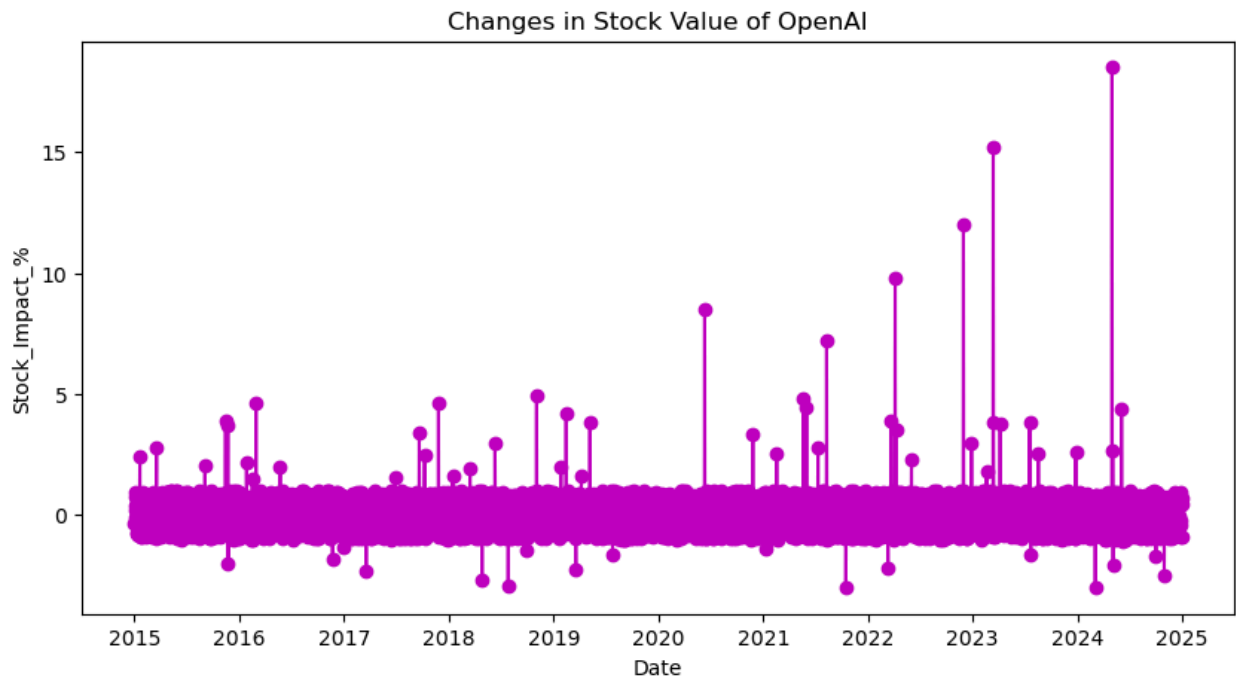
3653 rows × 8 columns

Stock Impact (%) — OpenAI

In [21]: `plt.figure(figsize = (10,5))`
`plt.plot(data_openai['Date'], data_openai['Stock_Impact_%'], color = 'm',marke`
`plt.title("Changes in Stock Value of OpenAI")`

```
plt.xlabel("Date")
plt.ylabel("Stock_Impact_%")

plt.show()
```



Stock Impact (%) — Google

```
In [22]: data_google
```

Out[22]:

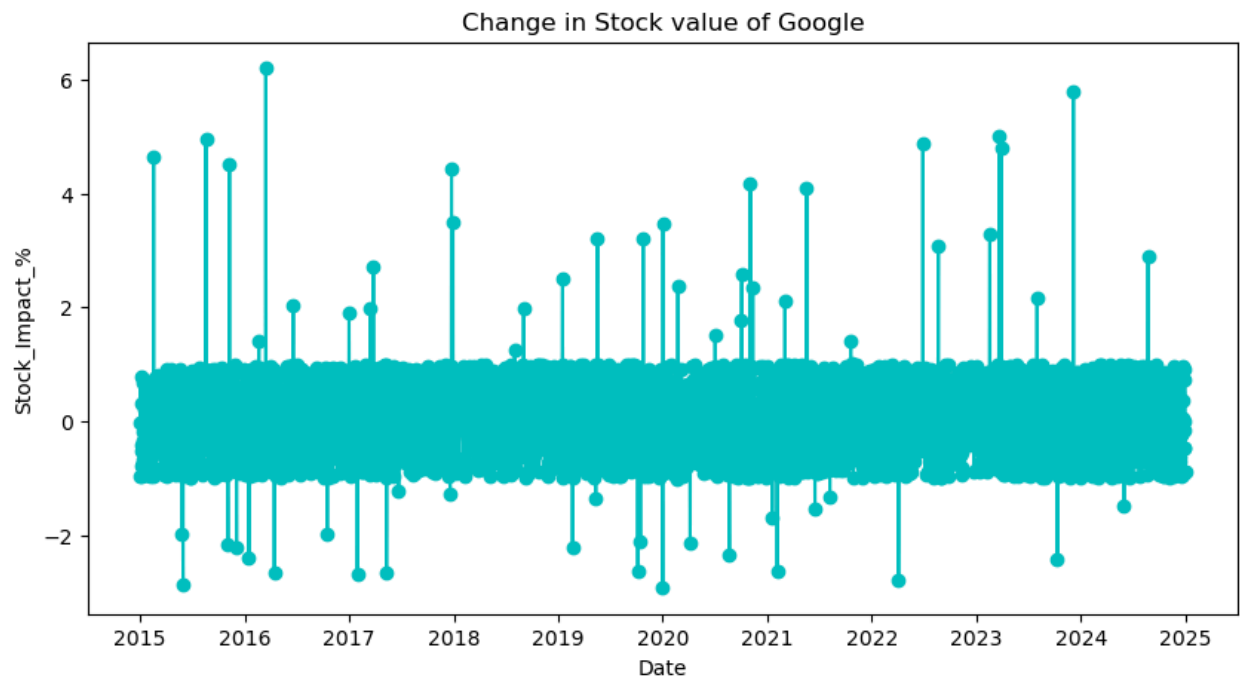
	Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Re
3653	2015-01-01	Google	79.89	30.19	
3654	2015-01-02	Google	78.99	30.44	
3655	2015-01-03	Google	79.20	30.46	
3656	2015-01-04	Google	79.59	30.55	
3657	2015-01-05	Google	81.50	30.59	
...
7301	2024-12-27	Google	162.16	155.36	
7302	2024-12-28	Google	159.69	154.47	
7303	2024-12-29	Google	161.69	154.59	
7304	2024-12-30	Google	158.48	155.05	
7305	2024-12-31	Google	159.48	154.77	

3653 rows x 8 columns

```
In [23]: plt.figure(figsize = (10,5))

plt.plot( data_google['Date'], data_google['Stock_Impact_%'], color = 'c',mark
plt.title("Change in Stock value of Google")
plt.xlabel("Date")
plt.ylabel("Stock_Impact_%")

plt.show()
```



Stock Impact (%) — Meta

```
In [24]: data_meta
```

Out[24]:

	Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_R
7306	2015-01-01	Meta	50.39	18.95	
7307	2015-01-02	Meta	49.80	19.77	
7308	2015-01-03	Meta	49.09	19.96	
7309	2015-01-04	Meta	50.66	20.48	
7310	2015-01-05	Meta	51.36	19.84	
...	
10954	2024-12-27	Meta	100.19	103.54	
10955	2024-12-28	Meta	99.12	102.37	
10956	2024-12-29	Meta	98.95	103.11	
10957	2024-12-30	Meta	100.74	103.21	
10958	2024-12-31	Meta	100.08	103.41	

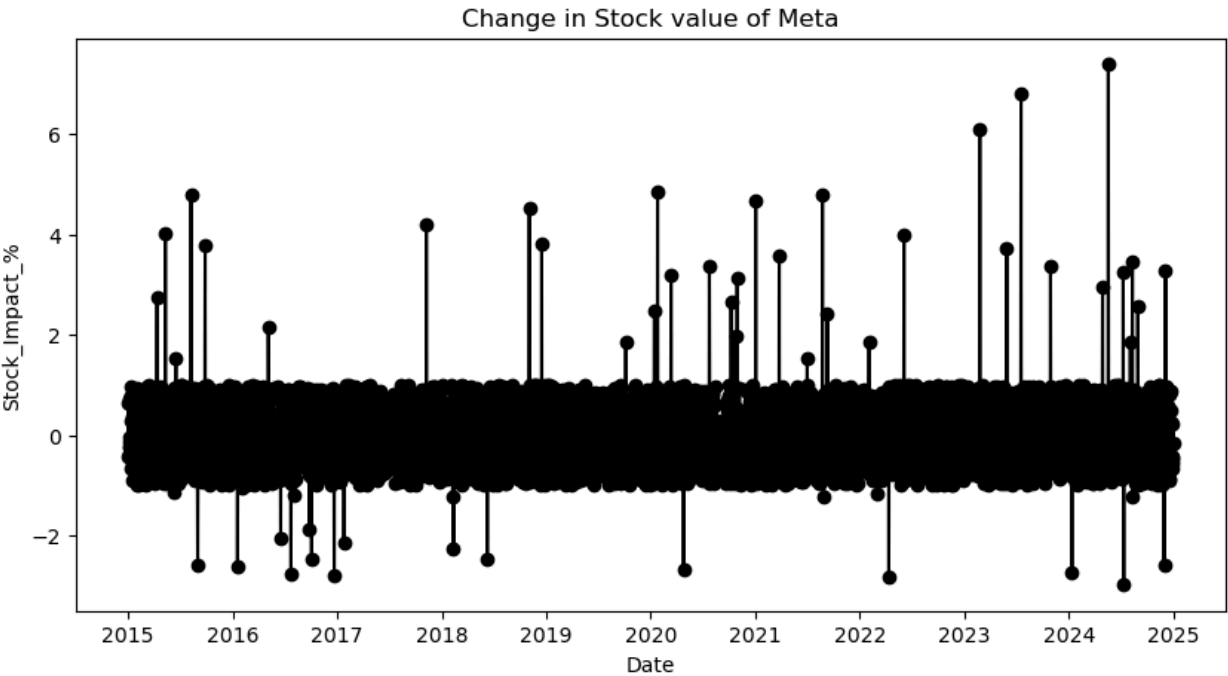
3653 rows × 8 columns

In [25]:

```
plt.figure(figsize = (10,5))

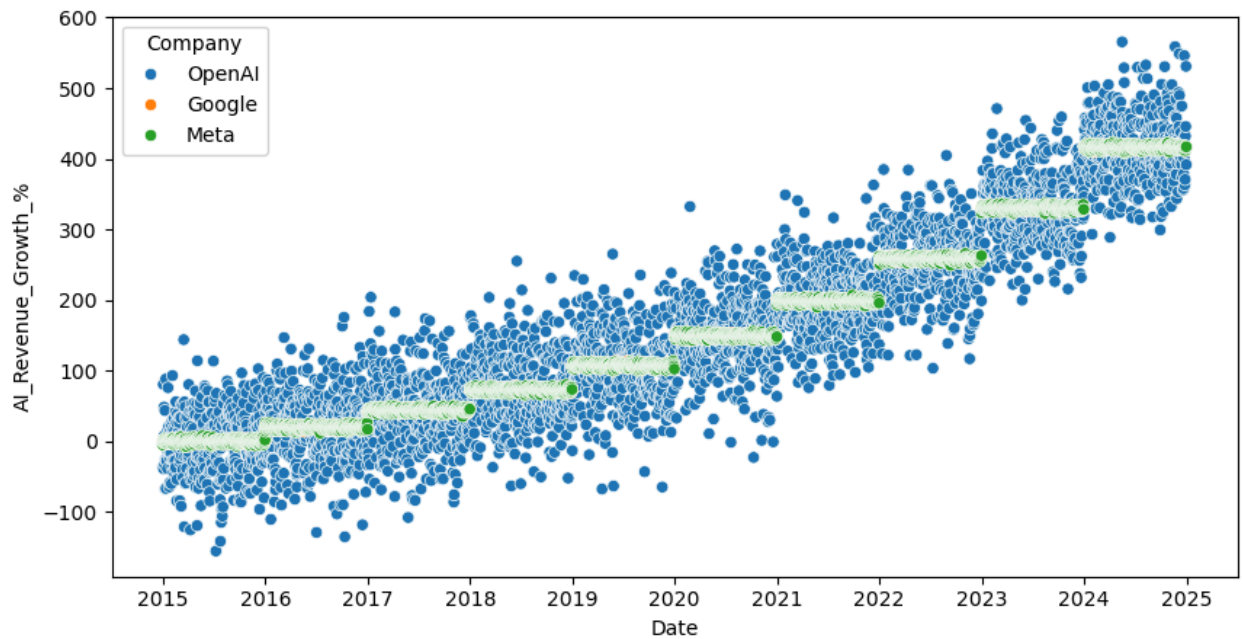
plt.plot( data_meta['Date'], data_meta['Stock_Impact_%'], color = 'black',mark
plt.title("Change in Stock value of Meta")
plt.xlabel("Date")
plt.ylabel("Stock_Impact_%")

plt.show()
```



AI Revenue Growth of the companies

```
In [26]: plt.figure(figsize = (10,5))  
  
sns.scatterplot(x = 'Date', y = 'AI_Revenue_Growth_%', data = df, hue = 'Company')  
  
plt.show()
```



AI Revenue Growth of the companies

```
In [27]: df.sort_values(by = ['AI_Revenue_Growth_%'])
```


Out[27]:

	Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Re
189	2015-07-09	OpenAI	5.66	-0.55	
205	2015-07-25	OpenAI	5.23	-0.42	
649	2016-10-11	OpenAI	5.37	-0.35	
548	2016-07-02	OpenAI	5.19	-0.29	
98	2015-04-09	OpenAI	4.80	-0.25	
...	
3508	2024-08-09	OpenAI	9.92	6.33	
3645	2024-12-24	OpenAI	8.79	6.46	
3627	2024-12-06	OpenAI	8.51	6.49	
3612	2024-11-21	OpenAI	10.16	6.59	
3423	2024-05-16	OpenAI	8.47	6.65	

10959 rows × 8 columns

In [28]: `df.sort_values(by = ['AI_Revenue_Growth_%'])`

Out[28]:

	Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Re
189	2015-07-09	OpenAI	5.66	-0.55	
205	2015-07-25	OpenAI	5.23	-0.42	
649	2016-10-11	OpenAI	5.37	-0.35	
548	2016-07-02	OpenAI	5.19	-0.29	
98	2015-04-09	OpenAI	4.80	-0.25	
...	
3508	2024-08-09	OpenAI	9.92	6.33	
3645	2024-12-24	OpenAI	8.79	6.46	
3627	2024-12-06	OpenAI	8.51	6.49	
3612	2024-11-21	OpenAI	10.16	6.59	
3423	2024-05-16	OpenAI	8.47	6.65	

10959 rows × 8 columns

In [29]: `data_openai`

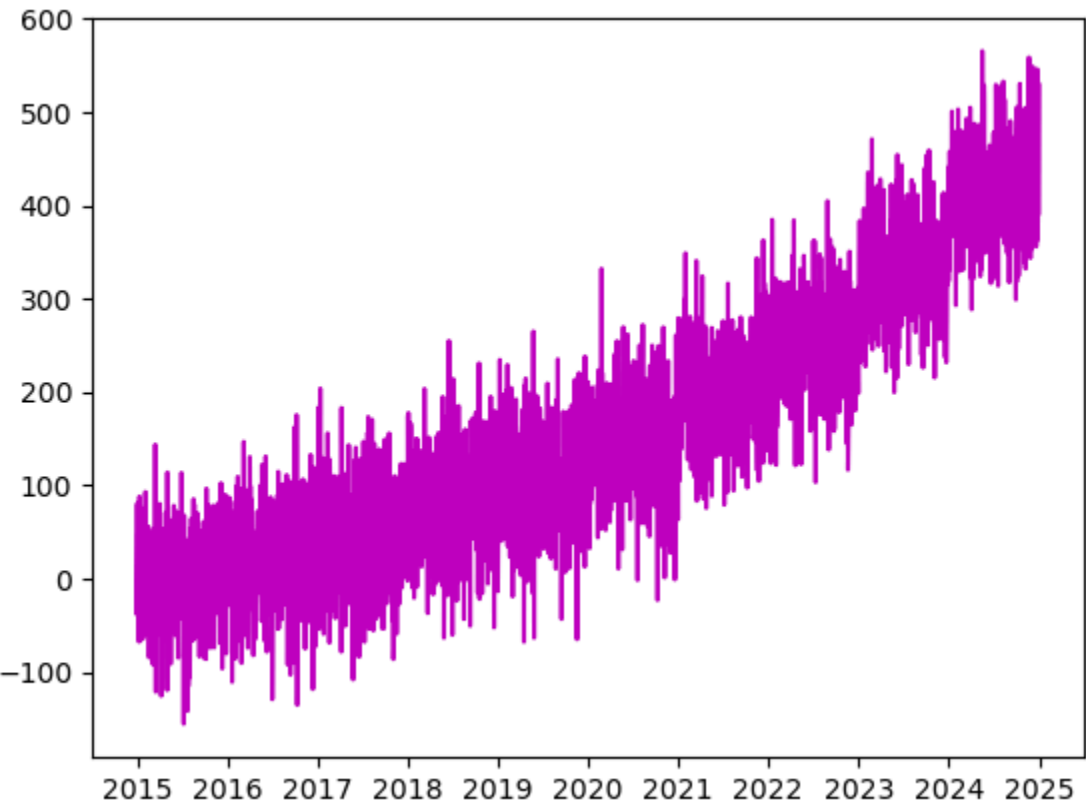
Out[29]:

	Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Re
0	2015-01-01	OpenAI	5.92	0.63	
1	2015-01-02	OpenAI	5.41	1.81	
2	2015-01-03	OpenAI	4.50	0.61	
3	2015-01-04	OpenAI	5.45	0.95	
4	2015-01-05	OpenAI	3.40	1.48	
...
3648	2024-12-27	OpenAI	10.06	4.71	
3649	2024-12-28	OpenAI	9.67	5.32	
3650	2024-12-29	OpenAI	9.17	5.46	
3651	2024-12-30	OpenAI	10.36	6.31	
3652	2024-12-31	OpenAI	10.24	4.92	

3653 rows × 8 columns

In [30]:

```
plt.plot( data_openai['Date'], data_openai['AI_Revenue_Growth_%'], color = 'm')
plt.show()
```



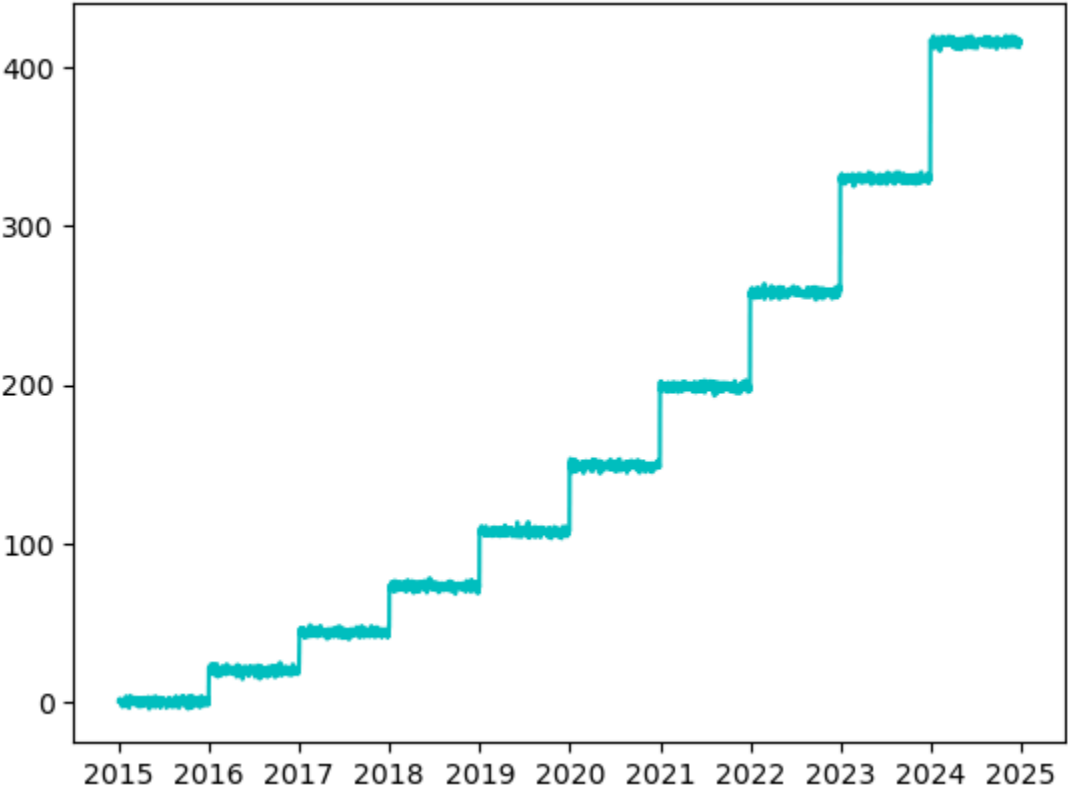
```
In [31]: data_google
```

Out[31]:

	Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Re
3653	2015-01-01	Google	79.89	30.19	
3654	2015-01-02	Google	78.99	30.44	
3655	2015-01-03	Google	79.20	30.46	
3656	2015-01-04	Google	79.59	30.55	
3657	2015-01-05	Google	81.50	30.59	
...
7301	2024-12-27	Google	162.16	155.36	
7302	2024-12-28	Google	159.69	154.47	
7303	2024-12-29	Google	161.69	154.59	
7304	2024-12-30	Google	158.48	155.05	
7305	2024-12-31	Google	159.48	154.77	

3653 rows × 8 columns

```
In [32]: plt.plot( data_google['Date'], data_google['AI_Revenue_Growth_%'], color = 'c')
plt.show()
```



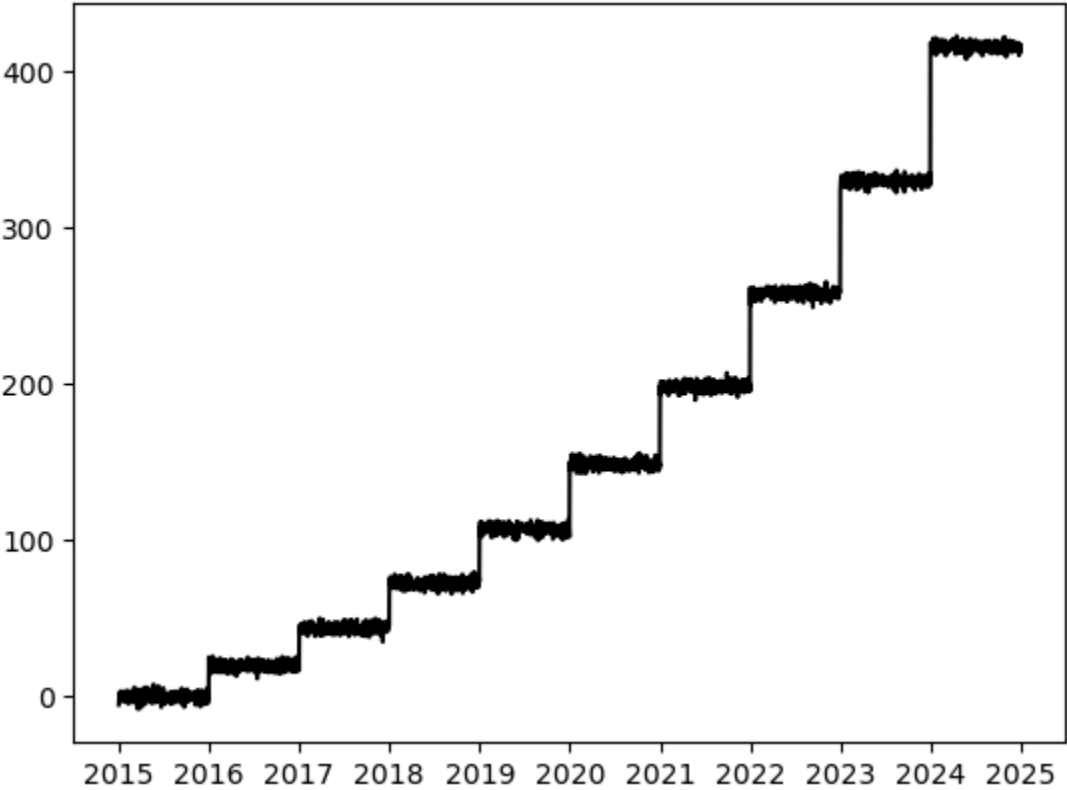
```
In [33]: data_meta
```

Out[33]:

	Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_R
7306	2015-01-01	Meta	50.39	18.95	
7307	2015-01-02	Meta	49.80	19.77	
7308	2015-01-03	Meta	49.09	19.96	
7309	2015-01-04	Meta	50.66	20.48	
7310	2015-01-05	Meta	51.36	19.84	
...
10954	2024-12-27	Meta	100.19	103.54	
10955	2024-12-28	Meta	99.12	102.37	
10956	2024-12-29	Meta	98.95	103.11	
10957	2024-12-30	Meta	100.74	103.21	
10958	2024-12-31	Meta	100.08	103.41	

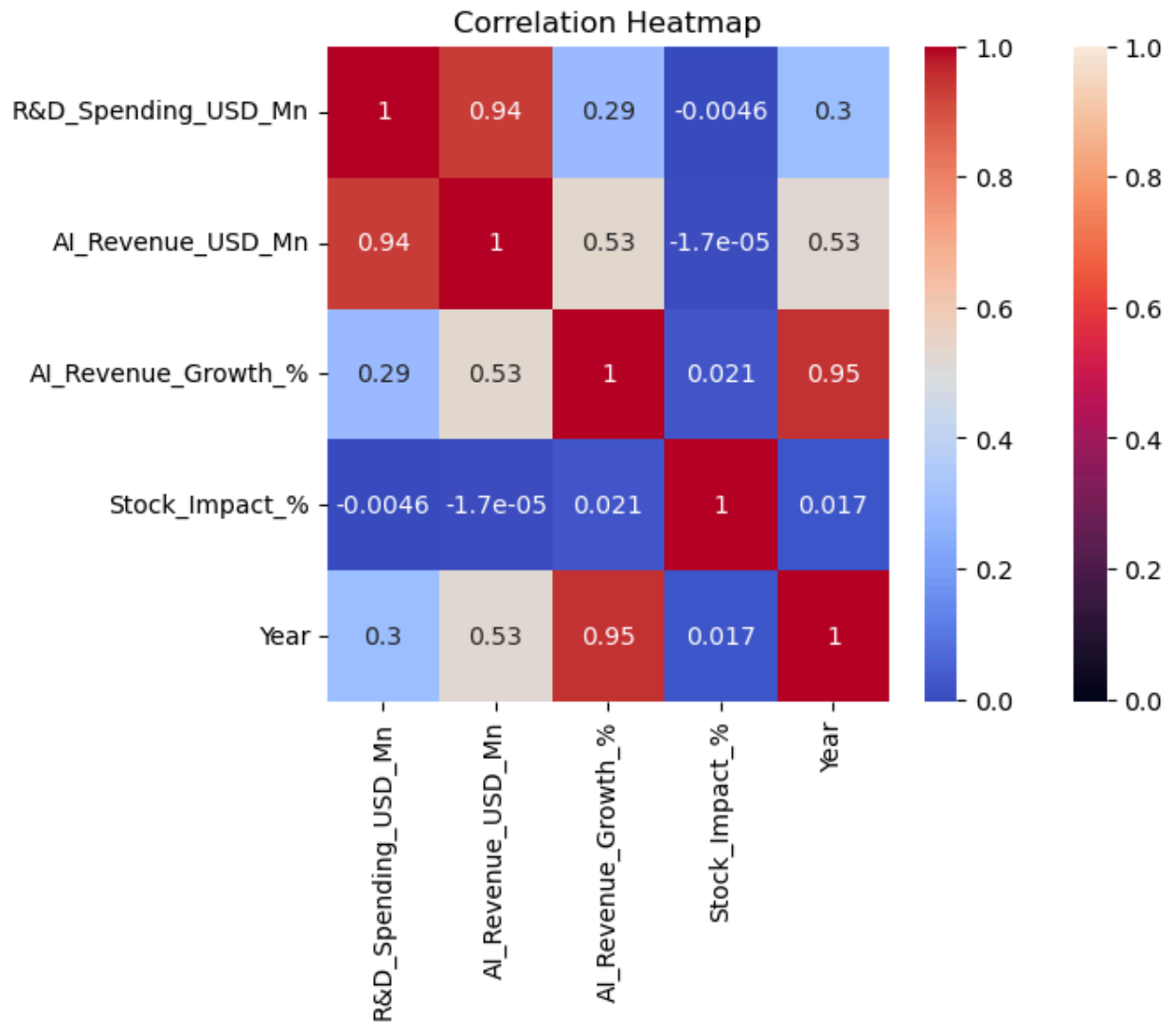
3653 rows × 8 columns

```
In [34]: plt.plot( data_meta['Date'], data_meta['AI_Revenue_Growth_%'], color = 'black'  
plt.show()
```



Correlation between the columns

```
In [36]: sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')  
plt.title("Correlation Heatmap")  
plt.show()
```



Expenditure vs Revenue year-by-year

```
In [37]: df.head()
```


Out[37]:		Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Reven
	0	2015-01-01	OpenAI	5.92	0.63	
	1	2015-01-02	OpenAI	5.41	1.81	
	2	2015-01-03	OpenAI	4.50	0.61	
	3	2015-01-04	OpenAI	5.45	0.95	
	4	2015-01-05	OpenAI	3.40	1.48	

```
In [38]: spend = df.groupby('Year')['R&D_Spending_USD_Mn'].sum()

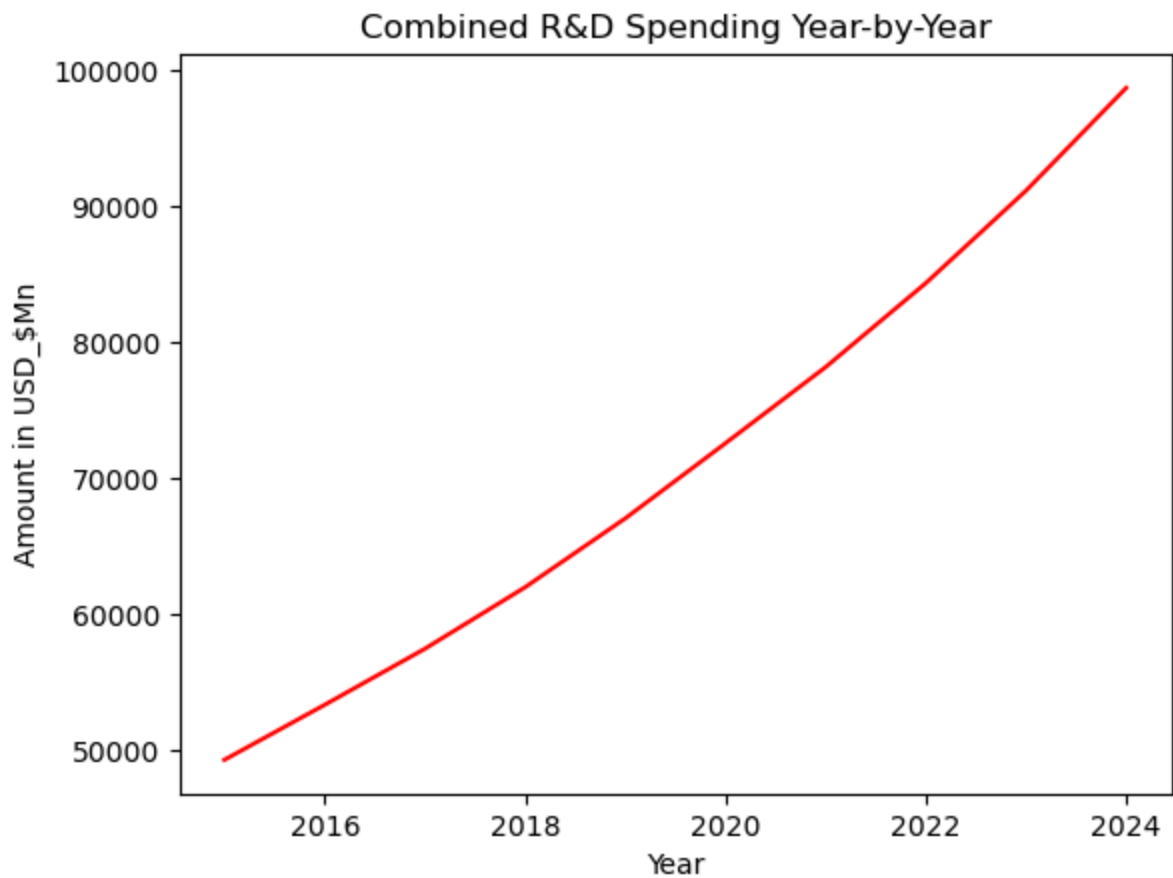
spend
```

```
Out[38]: Year
2015    49304.23
2016    53349.94
2017    57475.12
2018    62002.16
2019    67071.51
2020    72602.53
2021    78207.32
2022    84390.17
2023    91193.92
2024    98760.08
Name: R&D_Spending_USD_Mn, dtype: float64
```

```
In [39]: plt.plot(spend.index, spend.values, color = 'r')

plt.title("Combined R&D Spending Year-by-Year")
plt.xlabel("Year")
plt.ylabel("Amount in USD_$Mn")

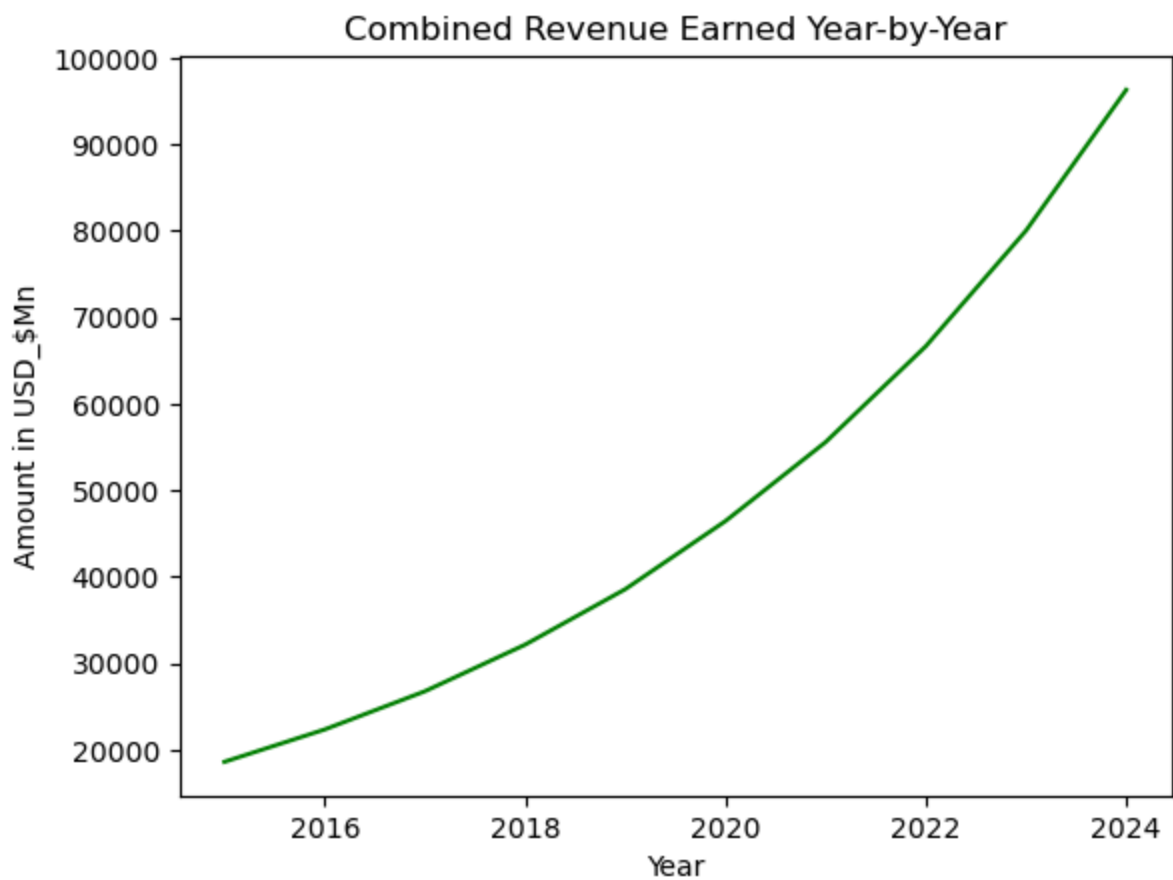
plt.show()
```



```
In [40]: revenue = df.groupby('Year')['AI_Revenue_USD_Mn'].sum()  
  
revenue
```

```
Out[40]: Year  
2015    18623.21  
2016    22361.20  
2017    26776.80  
2018    32164.44  
2019    38580.32  
2020    46456.19  
2021    55591.14  
2022    66674.24  
2023    80041.64  
2024    96313.91  
Name: AI_Revenue_USD_Mn, dtype: float64
```

```
In [41]: plt.plot(revenue.index, revenue.values, color = 'g')  
  
plt.title("Combined Revenue Earned Year-by-Year")  
plt.xlabel("Year")  
plt.ylabel("Amount in USD_$Mn")  
  
plt.show()
```

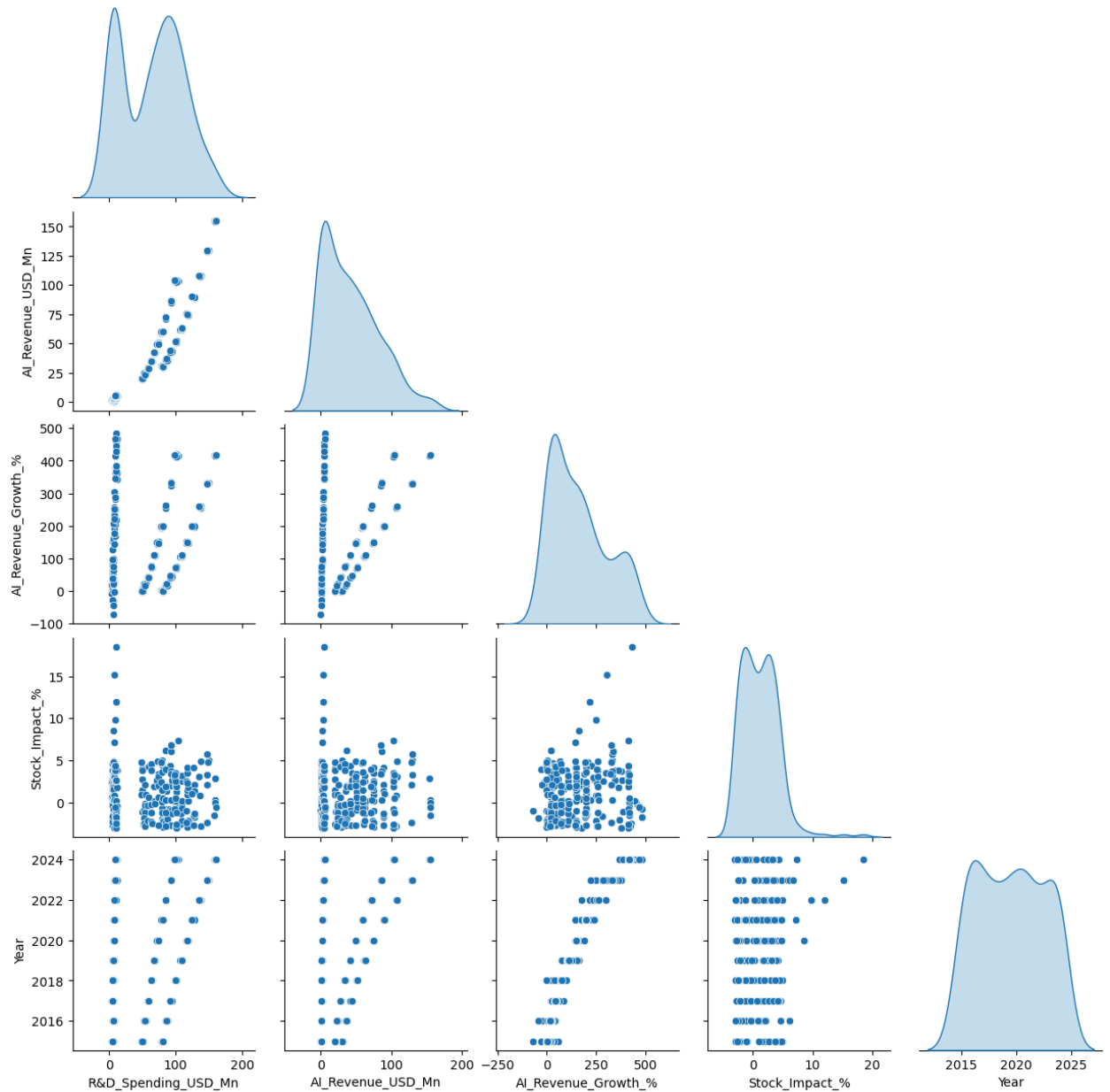


```
In [48]: # Clean the data: replace inf/-inf with NaN
df = df.replace([np.inf, -np.inf], np.nan)

# Drop rows with missing values (optional)
df = df.dropna()

# Pairplot
sns.pairplot(df, diag_kind="kde", corner=True)
plt.suptitle("Pairplot of Dataset Features", y=1.02)
plt.show()
```

Pairplot of Dataset Features



Event Impact Analysis

```
In [49]: df.head()
```

Out[49]:		Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Rev
	19	2015-01-20	OpenAI	4.37	0.91	
	77	2015-03-19	OpenAI	5.10	1.56	
	166	2015-06-16	OpenAI	6.01	0.28	
	246	2015-09-04	OpenAI	6.81	1.35	
	318	2015-11-15	OpenAI	5.84	0.69	

```
In [50]: df.Event.value_counts()
```

```
Out[50]: Event
AI speech recognition release    37
AI Ads Optimization upgrade      32
AI partnership deal              32
AI-powered search update         32
AI Video Recommendation upgrade  29
AI ethics policy update          27
Cloud AI launch                 27
GPT-1 release                   1
GPT-2 release                   1
GPT-3 release                   1
Codex release                   1
DALL·E 2 release                1
ChatGPT (GPT-3.5) launch        1
GPT-4 release                   1
GPT-5 release (predicted)       1
TensorFlow open-source release  1
AlphaGo beats Lee Sedol         1
BERT for Search launch          1
MUM Search Model launch         1
Bard chatbot launch             1
Gemini AI release               1
LLaMA 1 release                 1
LLaMA 2 release                 1
LLaMA 3 release (predicted)     1
Name: count, dtype: int64
```

```
In [51]: df[ df.Event == 'TensorFlow open-source release' ]
```



```
Out[51]:
```

	Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Re
3965	2015-11-09	Google	79.62	29.84	

```
In [52]: tf = df.loc[ 3955 : 3975 ]

tf
```

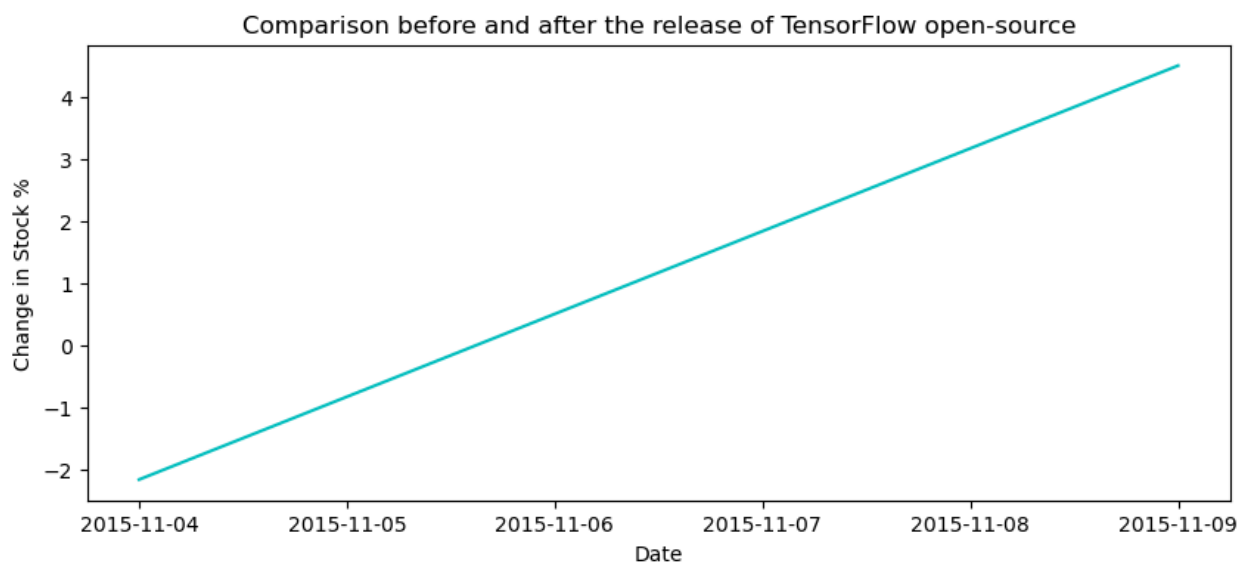
```
Out[52]:
```

	Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Re
3960	2015-11-04	Google	80.70	29.71	
3965	2015-11-09	Google	79.62	29.84	

```
In [53]: plt.figure(figsize = (10,4))

plt.plot( tf['Date'], tf['Stock_Impact_%'], color = 'c')
plt.title("Comparison before and after the release of TensorFlow open-source")
plt.xlabel("Date")
plt.ylabel("Change in Stock %")

plt.show()
```



```
In [54]: df[ df.Event == 'GPT-4 release']
```

```
Out[54]:
```

	Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Re
2994	2023-03-14	OpenAI	7.78	4.05	

```
In [55]: gpt4 = df.loc[ 2984 : 3004]

gpt4
```

```
Out[55]:
```

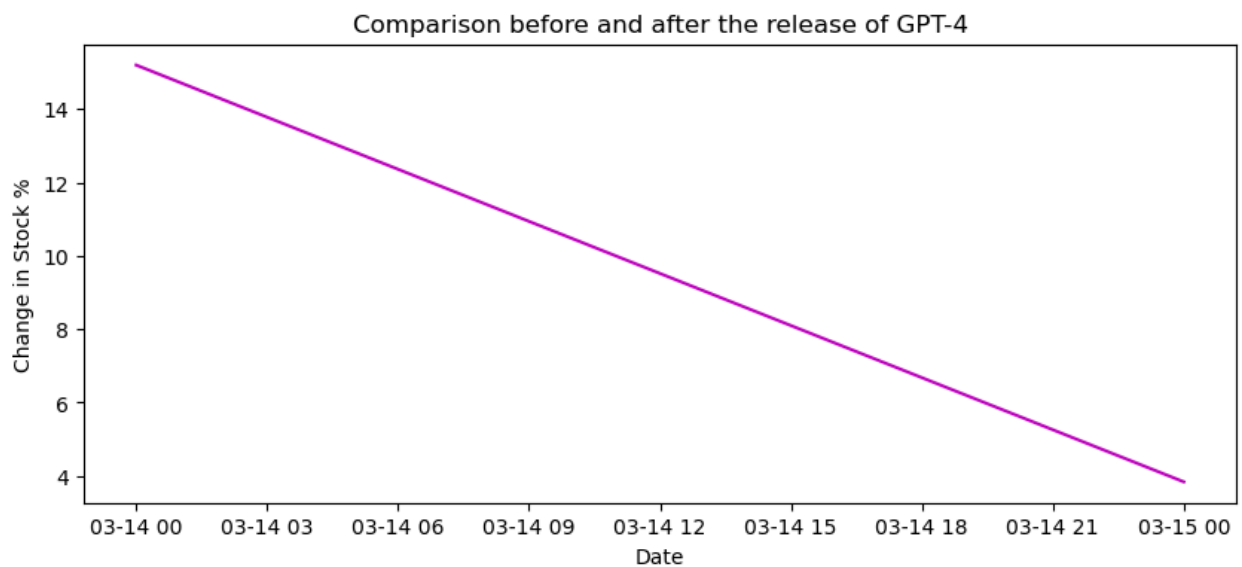
	Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Re
2994	2023-03-14	OpenAI	7.78	4.05	
2995	2023-03-15	OpenAI	12.31	4.44	

```
In [56]: plt.figure(figsize = (10,4))

plt.plot( gpt4['Date'], gpt4['Stock_Impact_%'], color = 'm')

plt.title("Comparison before and after the release of GPT-4")
plt.xlabel("Date")
plt.ylabel("Change in Stock %")

plt.show()
```



```
In [57]: df.head(2)
```

Out[57]:

	Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Reve
--	------	---------	---------------------	-------------------	---------

19	2015-01-20	OpenAI	4.37	0.91	
----	------------	--------	------	------	--

77	2015-03-19	OpenAI	5.10	1.56	
----	------------	--------	------	------	--

```
In [58]: df.groupby('Company')['Stock_Impact_%'].mean()*100
```

```
Out[58]: Company
Google      64.538462
Meta        101.389610
OpenAI      191.717949
Name: Stock_Impact_%, dtype: float64
```

```
In [59]: df.groupby('Company')['R&D_Spending_USD_Mn'].mean()
```

```
Out[59]: Company
Google      111.860513
Meta         72.912338
OpenAI        7.528718
Name: R&D_Spending_USD_Mn, dtype: float64
```

```
In [60]: df.groupby('Company')['Stock_Impact_%'].max()
```

```
Out[60]: Company
Google        6.2
Meta          7.4
OpenAI       18.5
Name: Stock_Impact_%, dtype: float64
```

Change in the index wrt Year & Company

```
In [61]: df.head(2)
```

Out[61]:

	Date	Company	R&D_Spending_USD_Mn	AI_Revenue_USD_Mn	AI_Reve
--	------	---------	---------------------	-------------------	---------

19	2015-01-20	OpenAI	4.37	0.91	
----	------------	--------	------	------	--

77	2015-03-19	OpenAI	5.10	1.56	
----	------------	--------	------	------	--

```
In [62]: stocks = df.groupby(['Year', 'Company'])['Stock_Impact_%'].max()

stocks
```

```
Out[62]:
```

Year	Company	
2015	Google	4.96
	Meta	4.80
	OpenAI	3.92
2016	Google	6.20
	Meta	2.14
	OpenAI	4.66
2017	Google	4.42
	Meta	4.21
	OpenAI	4.63
2018	Google	1.97
	Meta	4.54
	OpenAI	4.94
2019	Google	3.20
	Meta	1.86
	OpenAI	4.20
2020	Google	4.18
	Meta	4.86
	OpenAI	8.50
2021	Google	4.10
	Meta	4.80
	OpenAI	7.20
2022	Google	4.88
	Meta	3.98
	OpenAI	12.00
2023	Google	5.80
	Meta	6.80
	OpenAI	15.20
2024	Google	2.89
	Meta	7.40
	OpenAI	18.50

Name: Stock_Impact_%, dtype: float64

```
In [63]: stocks.plot(kind = 'barh', color = ['r', 'black', 'm'])

plt.title("change in index")

plt.show()
```

