

```
In [84]: !pip install yfinance  
!pip install wordcloud
```

```
Collecting wordcloud  
  Downloading wordcloud-1.9.4-cp313-cp313-win_amd64.whl.metadata (3.5 kB)  
Requirement already satisfied: numpy>=1.6.1 in c:\users\adham\anaconda3\lib\site-packages (from wordcloud) (2.1.3)  
Requirement already satisfied: pillow in c:\users\adham\anaconda3\lib\site-packages (from wordcloud) (11.1.0)  
Requirement already satisfied: matplotlib in c:\users\adham\anaconda3\lib\site-packages (from wordcloud) (3.10.0)  
Requirement already satisfied: contourpy>=1.0.1 in c:\users\adham\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.3.1)  
Requirement already satisfied: cycler>=0.10 in c:\users\adham\anaconda3\lib\site-packages (from matplotlib->wordcloud) (0.11.0)  
Requirement already satisfied: fonttools>=4.22.0 in c:\users\adham\anaconda3\lib\site-packages (from matplotlib->wordcloud) (4.55.3)  
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\adham\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.4.8)  
Requirement already satisfied: packaging>=20.0 in c:\users\adham\anaconda3\lib\site-packages (from matplotlib->wordcloud) (24.2)  
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\adham\anaconda3\lib\site-packages (from matplotlib->wordcloud) (3.2.0)  
Requirement already satisfied: python-dateutil>=2.7 in c:\users\adham\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.9.0.post0)  
Requirement already satisfied: six>=1.5 in c:\users\adham\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.17.0)  
  Downloading wordcloud-1.9.4-cp313-cp313-win_amd64.whl (300 kB)  
Installing collected packages: wordcloud  
Successfully installed wordcloud-1.9.4
```

```
In [80]: import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import re  
from wordcloud import STOPWORDS  
from collections import Counter  
from wordcloud import WordCloud  
import seaborn as sns
```

```
In [3]: import yfinance as yf  
from io import StringIO
```

Downloading financial data using Yahoo Finance API

Ubisoft Stock

```
In [10]: ubiStock = yf.download('UBI.PA' , '2022-01-01' , '2025-08-28')  
ubiStock
```

```
C:\Users\adham\AppData\Local\Temp\ipykernel_12064\1712222148.py:1: FutureWarning: Y  
F.download() has changed argument auto_adjust default to True  
    ubiStock = yf.download('UBI.PA' , '2022-01-01' , '2025-08-28')  
[*****100%*****] 1 of 1 completed
```

```
Out[10]:
```

	Price	Close	High	Low	Open	Volume
Ticker	UBI.PA	UBI.PA	UBI.PA	UBI.PA	UBI.PA	UBI.PA
Date						
2022-01-03	42.880001	43.220001	42.540001	43.060001	240097	
2022-01-04	41.590000	43.259998	41.540001	42.980000	422725	
2022-01-05	41.330002	42.160000	41.119999	41.500000	594441	
2022-01-06	41.430000	41.750000	40.439999	41.049999	370862	
2022-01-07	40.939999	41.720001	40.770000	41.439999	389427	
...
2025-08-21	9.176000	9.282000	9.060000	9.204000	351425	
2025-08-22	9.452000	9.610000	9.136000	9.136000	441898	
2025-08-25	9.422000	9.590000	9.372000	9.474000	328730	
2025-08-26	9.254000	9.424000	9.158000	9.244000	438652	
2025-08-27	9.088000	9.306000	8.986000	9.226000	368397	

935 rows × 5 columns

Roblox Stock

```
In [11]: robloxStock = yf.download('RBLX' , '2022-01-01' , '2025-08-28')  
robloxStock
```

```
C:\Users\adham\AppData\Local\Temp\ipykernel_12064\2137125785.py:1: FutureWarning: Y  
F.download() has changed argument auto_adjust default to True  
    robloxStock = yf.download('RBLX' , '2022-01-01' , '2025-08-28')  
[*****100%*****] 1 of 1 completed
```

Out[11]:	Price	Close	High	Low	Open	Volume
	Ticker	RBLX	RBLX	RBLX	RBLX	RBLX
	Date					
	2022-01-03	98.809998	103.790001	97.620003	101.910004	16964300
	2022-01-04	95.150002	99.250000	91.769997	99.019997	23034600
	2022-01-05	88.540001	95.834000	88.059998	93.519997	15510000
	2022-01-06	89.199997	92.080002	84.690002	87.989998	17213300
	2022-01-07	84.370003	88.690002	82.580002	86.379997	25003600

	2025-08-21	114.849998	118.000000	113.699997	116.389999	7005200
	2025-08-22	117.519997	119.410004	114.120003	115.779999	7644700
	2025-08-25	124.809998	127.550003	122.000000	123.220001	13501100
	2025-08-26	124.760002	126.949997	122.629997	123.625000	13720800
	2025-08-27	122.699997	125.059998	121.279999	124.989998	6751200

916 rows × 5 columns

Uploading Activision Blizzard Stock

```
In [87]: activStock =pd.read_csv("activ stock.csv")
activStock.set_index("Date" , inplace = True)
activStock.head()
```

Out[87]:	Price	Open	High	Low	Vol.	Change %
	Date					
	16-Sep-2019	55.78	55.18	55.94	55.17	6.47M
	17-Sep-2019	55.35	55.50	55.72	54.77	6.98M
	18-Sep-2019	55.12	55.21	55.23	54.13	5.45M
	19-Sep-2019	55.01	55.43	55.56	54.71	4.57M
	20-Sep-2019	54.76	55.04	55.48	54.52	7.51M

Wanted to check the directory where jupyter is fetching files from

```
In [7]: import os  
print(os.getcwd())
```

```
C:\Users\adham\anaconda_projects\8f0910af-928a-4f1a-ba40-6bb9707bf49d
```

Analyzing Roblox

```
In [49]: robCount = pd.read_excel("robplayercount.xlsx")  
print(robCount.head())
```

```
   Year Quarter Player Count  
0  2018      Q1  10.3 million  
1  2018      Q2  11.3 million  
2  2018      Q3  12.7 million  
3  2018      Q4  13.7 million  
4  2019      Q1  15.8 million
```

```
In [ ]: #Cleaning the Player Count column
```

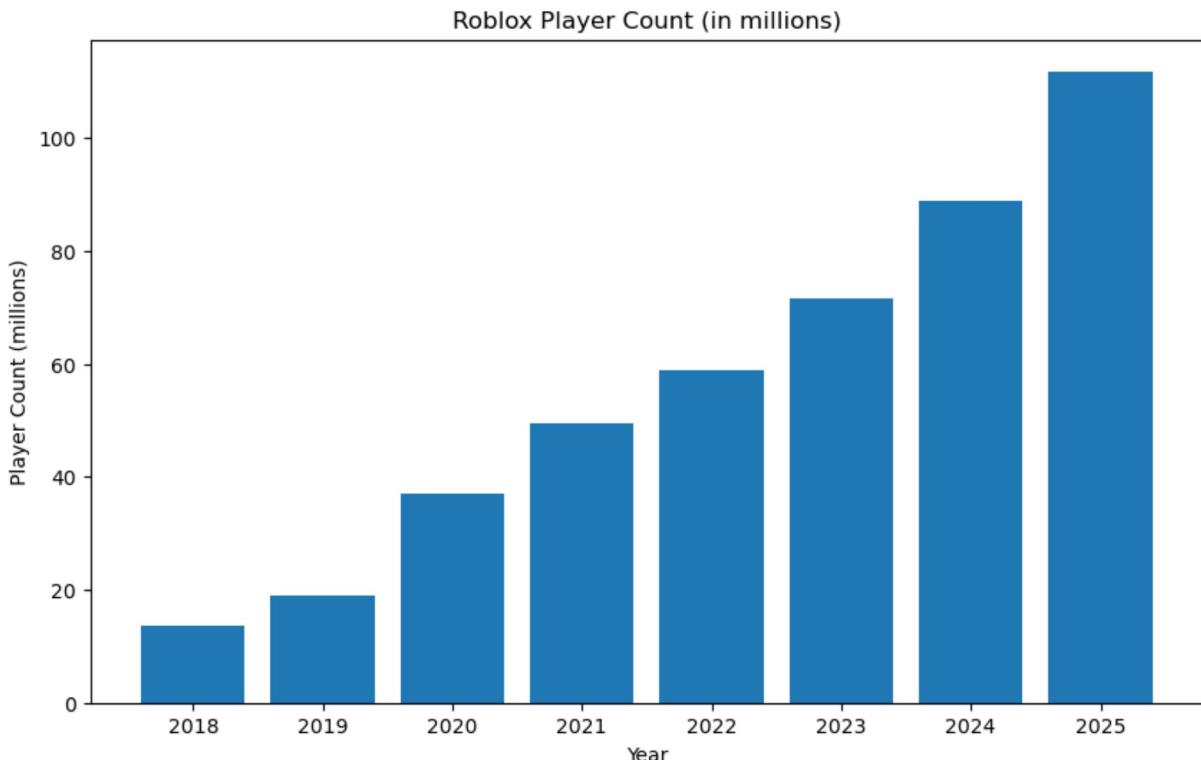
```
robCount["Player Count"] = robCount["Player Count"].astype(str)  
robCount["Player Count"] = (robCount["Player Count"].str.replace("million", "")  
                           .str.replace(" ", ""))  
robCount["Player Count"] = pd.to_numeric(robCount["Player Count"])
```

```
In [51]: print(robCount.head())
```

```
   Year Quarter Player Count  
0  2018      Q1        10.3  
1  2018      Q2        11.3  
2  2018      Q3        12.7  
3  2018      Q4        13.7  
4  2019      Q1        15.8
```

```
In [53]: #A bar plot of Roblox player count
```

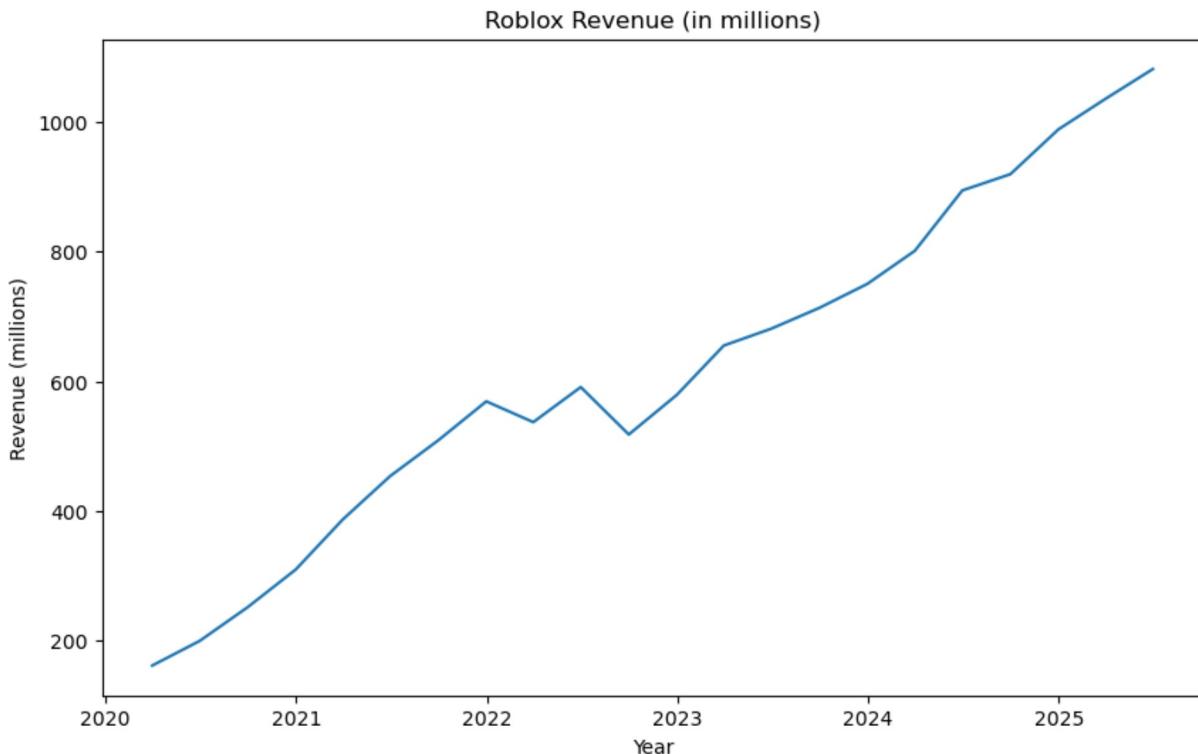
```
plt.figure(figsize=(10,6))  
plt.bar(robCount["Year"], robCount["Player Count"])  
  
plt.title("Roblox Player Count (in millions)")  
plt.xlabel("Year")  
plt.ylabel("Player Count (millions)")  
plt.show()
```



```
In [6]: #Importing Roblox Revenue Excel sheet  
  
robRev = pd.read_excel("robloxrev.xlsx")  
print(robRev.head())
```

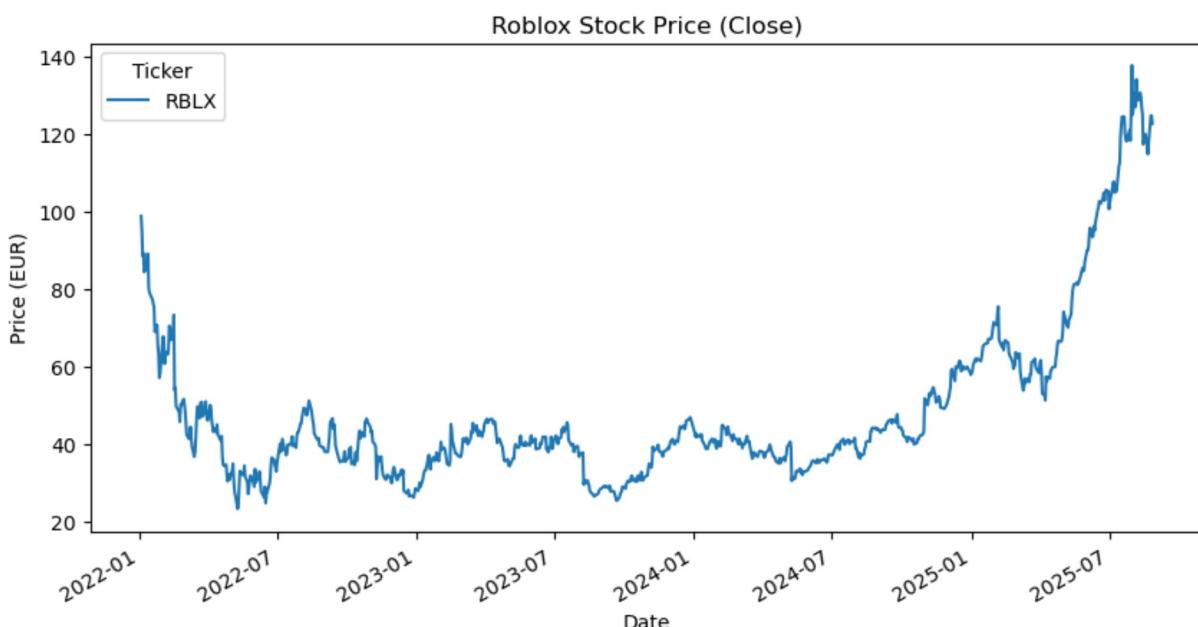
```
Year    Profit  
0 2025-06-30     1081  
1 2025-03-31     1035  
2 2024-12-31      988  
3 2024-09-30      919  
4 2024-06-30      894
```

```
In [64]: #A Line chart of Roblox Revenue  
  
plt.figure(figsize=(10,6))  
plt.plot(robRev["Year"], robRev["Profit"])  
  
plt.title("Roblox Revenue (in millions)")  
plt.xlabel("Year")  
plt.ylabel("Revenue (millions)")  
plt.show()
```



```
In [13]: #A Line chart of Roblox Stock Price
```

```
robloxStock["Close"].plot(figsize=(10,5), title="Roblox Stock Price (Close)")
plt.ylabel("Price (EUR)")
plt.show()
```



Plotting a cloud chart of Roblox Parents' Reviews

```
In [99]: roboreview = pd.read_excel("robloxreviews.xlsx")
```

In [119... #Cleaning the reviews Excel sheet and leaving out irrelevant words

```
text = " ".join(robreview["Review"].astype(str)).lower()

words = re.findall(r"[a-z]+", text)

stopwords = set(STOPWORDS)
stopwords.update(["roblox", "game", "kids", "children", "play", "much", "games", "title",
                  "make", "got", "adult", "times", "lot", "account", "little", "chat"])

filtered = [w for w in words if w not in stopwords]

word_freq = Counter(filtered).most_common(15)
print(word_freq)
```

```
[('sex', 31), ('unsafe', 21), ('violence', 19), ('swearing', 19), ('child', 18), ('drugs', 14), ('drinking', 13), ('smoking', 13), ('predators', 13), ('content', 12), ('playing', 10), ('dating', 10), ('consumerism', 9), ('many', 9), ('bad', 9)]
```

```
In [116]: clean_text = " ".join(filtered)
```

```
wc = WordCloud(  
    width=1000, height=500,  
    background_color="white",  
    stopwords=stopwords,  
    colormap="viridis"  
).generate(clean_text)  
  
plt.figure(figsize=(15, 7))  
plt.imshow(wc, interpolation="bilinear")  
plt.axis("off")  
plt.title("Word Cloud of Roblox Reviews", fontsize=15)  
plt.show()
```



Analyzing Activision Blizzard

```
In [88]: #A Line chart of Activision Blizzard Stock Price
```

```
activStock["Price"].plot(figsize=(10,5), title="Activision Blizzard Stock Price")
plt.ylabel("Price (USD)")
plt.show()
```



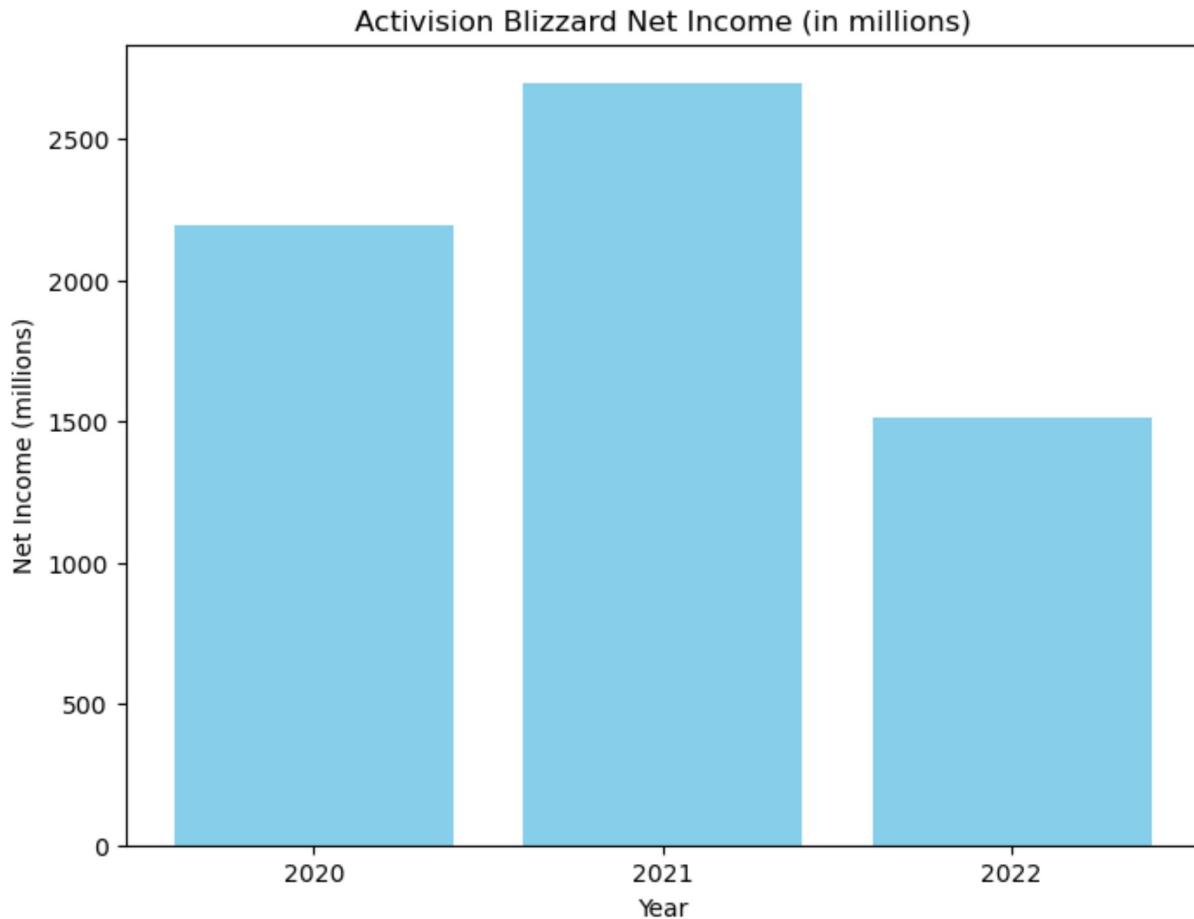
```
In [63]: activNI = pd.read_excel("activ ni.xlsx")
```

```
In [72]: #A bar plot of Activision Blizzard Net Income
```

```
activNI["Year"] = pd.to_numeric(activNI["Year"], errors="coerce").astype("Int64")
activNI = activNI.sort_values("Year")

plt.figure(figsize=(8,6))
plt.bar(activNI["Year"], activNI["NI"], color="skyblue")

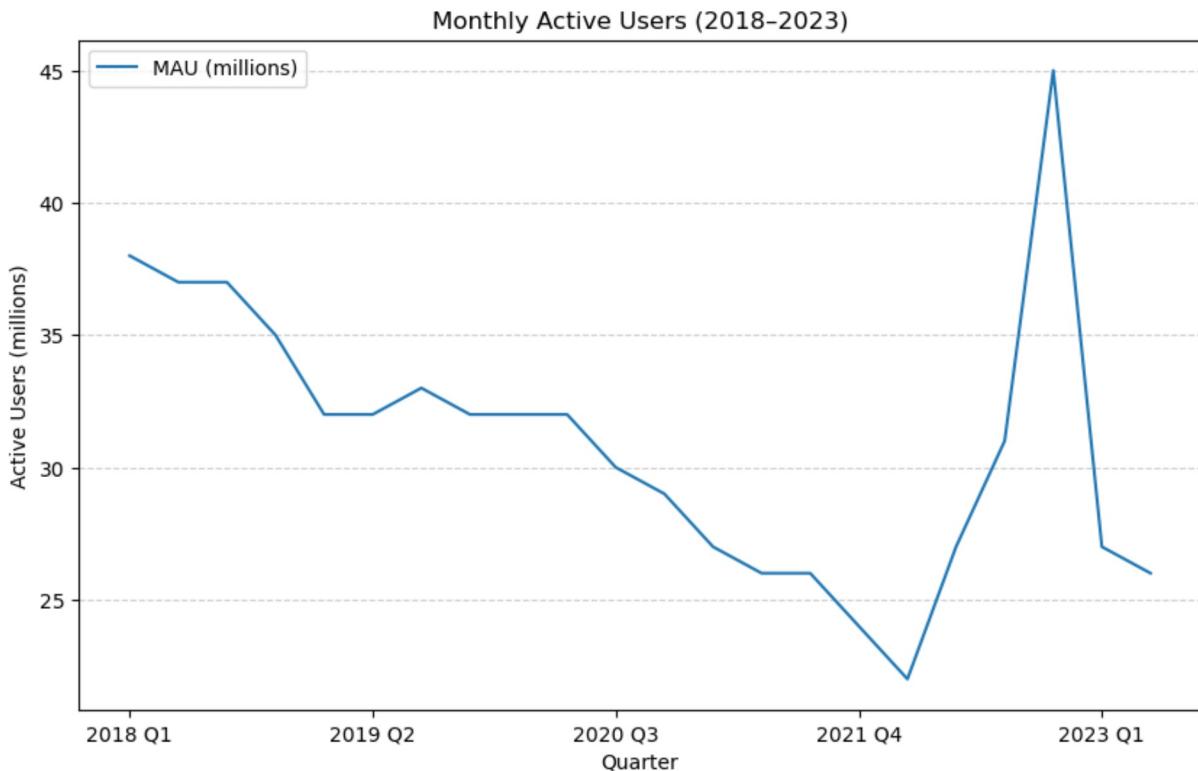
plt.title("Activision Blizzard Net Income (in millions)")
plt.xlabel("Year")
plt.ylabel("Net Income (millions)")
plt.xticks(activNI["Year"])
plt.show()
```



```
In [74]: blizzCount = pd.read_excel("blizzardpc.xlsx")
blizzCount.head()
```

```
Out[74]:    Quarter  MAU (millions)
0   2018 Q1        38
1   2018 Q2        37
2   2018 Q3        37
3   2018 Q4        35
4   2019 Q1        32
```

```
In [85]: blizzCount.plot(x="Quarter", kind="line", figsize=(10,6))
plt.title("Monthly Active Users (2018-2023)")
plt.ylabel("Active Users (millions)")
plt.grid(axis="y", linestyle="--", alpha=0.6)
plt.show()
```



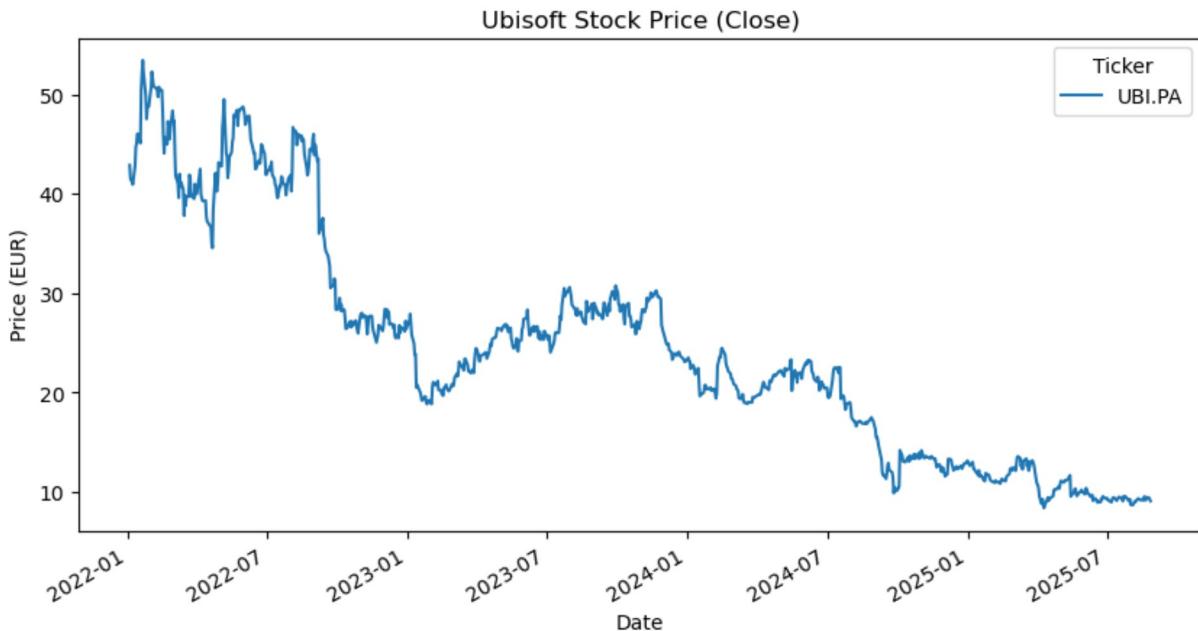
Analyzing Ubisoft

```
In [28]: ubiStock.reset_index(inplace=True)  
ubiStock.head()
```

```
Out[28]:
```

	Price	Date	Close	High	Low	Open	Volume
Ticker			UBI.PA	UBI.PA	UBI.PA	UBI.PA	UBI.PA
0	2022-01-03	42.880001	43.220001	42.540001	43.060001	240097	
1	2022-01-04	41.590000	43.259998	41.540001	42.980000	422725	
2	2022-01-05	41.330002	42.160000	41.119999	41.500000	594441	
3	2022-01-06	41.430000	41.750000	40.439999	41.049999	370862	
4	2022-01-07	40.939999	41.720001	40.770000	41.439999	389427	

```
In [16]: ubiStock["Close"].plot(figsize=(10,5), title="Ubisoft Stock Price (Close)")  
plt.ylabel("Price (EUR)")  
plt.show()
```



```
In [47]: ubirevenue = pd.read_excel("ubirev.xlsx")
ubirevenue.head()
```

```
Out[47]:
```

	Year	Revenue	Change
0	2025	\$2.05 B	-0.1726
1	2024	\$2.48 B	0.2600
2	2023	\$1.97 B	-0.1624
3	2022	\$2.35 B	-0.0978
4	2021	\$2.60 B	0.4832

```
In [ ]: #Cleaning the Revenue column

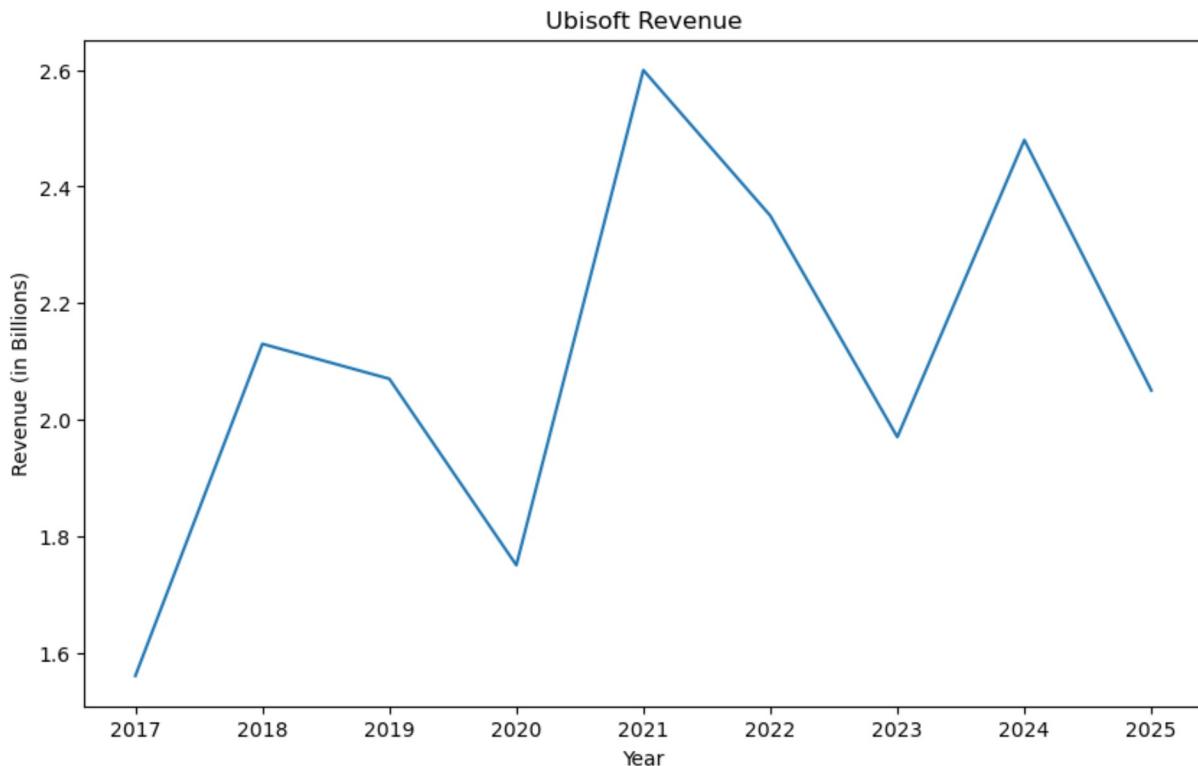
ubirevenue['Revenue'] = (ubirevenue['Revenue'].str.replace(" ", ""))
                     .str.replace("B", "")
                     .str.replace("$", "").astype(float))
```

```
In [64]: #Sorting by the Year column

ubirevenue = ubirevenue.sort_index()
```

```
In [72]: plt.figure(figsize=(10,6))
plt.plot(ubirevenue["Year"], ubirevenue["Revenue"])

plt.title("Ubisoft Revenue")
plt.ylabel("Revenue (in Billions)")
plt.xlabel("Year")
plt.xticks(rotation=0)
plt.show()
```



```
In [74]: ubireview = pd.read_excel("trustpilotubi.xlsx")
ubireview.head()
```

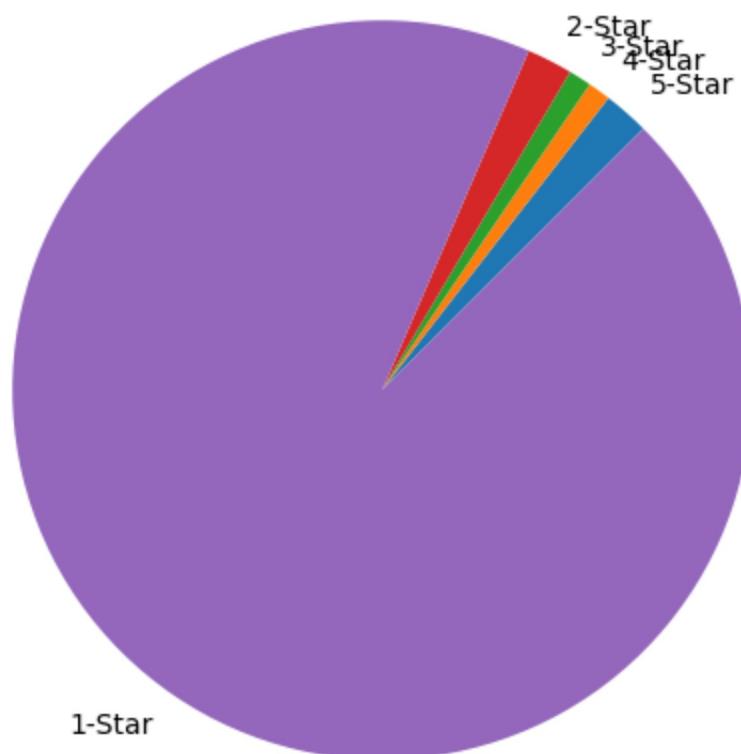
Out[74]:

	Review	Percentage
0	5-Star	0.02
1	4-Star	0.01
2	3-Star	0.01
3	2-Star	0.02
4	1-Star	0.94

```
In [88]: #A pie chart showing reviews of Ubisoft as a whole from Trustpilot.com

plt.figure(figsize=(6,6))
plt.pie(
    ubireview["Percentage"],
    labels=ubireview["Review"],
    startangle=45,
    labeldistance=1.1,
    textprops={'fontsize': 10, 'color': 'black'}
)
plt.title("Trustpilot's Ubisoft Reviews")
plt.show()
```

Trustpilot's Ubisoft Reviews



In []: