

# Unicorn Companies Dataset Analysis

**Guided by:**

DR. Shilpa Balan

College of Business and  
Economics,  
California State  
University, LA

**By:**

Hemalatha Ramakrishna,  
Priya Ramdas

## Introduction

---

Unicorn companies or unicorn startups are private companies worth over \$ 1 billion. The 21st century is the century of startups. Over the last two decades, we've seen many unicorn startups responsible for new technologies, space exploration, educational improvements, and more. In 2019, there will be 354 active unicorn companies worldwide, based on the previous year's peak of 348. At the end of 2019, there were 439 active unicorn companies, including 139 new startups. By 2020, 162 new companies worldwide gained unicorn status, bringing the total number of active unicorns to 538. In 2021, we saw an astonishing 355 new unicorn companies, ending the year with 537 companies, much the same as last year. As of March 2022, there are 1,000 unicorns around the world. Popular ex-unicorns include Airbnb, Facebook, and Google. Variants include ten kernels worth over \$ 10 billion and Hectocorn seeds worth over \$ 100 billion. Many unicorn companies are born because the explosive growth of unicorn companies is increasingly converging between the private and public markets. Companies have often relied on IPOs to generate the capital to grow their businesses. But today, companies can raise more private funding early on and reach billions of valuations without making it public. [1]

In the startup world, unicorns are companies that are said to have a rare kind of entrepreneurial magic. They are a privately held company worth over \$ 1 billion. Uber and Spotify were unicorns. DoorDash and Airbnb are still described that way. But as an investment in Silicon Valley has boomed in recent years, there are far more \$1 billion-plus startups than ever. The number of unicorns has been rising because of an influx of both startups and investors. After the recession caused a sharp decline in startups and other new businesses, many new companies entered the market, and many of them were startups. Today, companies can raise

more significant amounts of private funding early on, allowing them to reach billion-dollar valuations without going public. [2]

Of the top 10 new unicorns in the United States, 60% were FinTech start-ups, and five were related to blockchain or cryptocurrencies. Given the performance of these leading fintech unicorns, there seems to be a great deal of interest in this area, from which more unicorn companies may emerge. Most unicorns come from seven sectors: e-commerce, fintech, internet software, healthcare, travel technology, and education technology. San Francisco's Silicon Valley is still synonymous with startups, but more unicorn companies are emerging elsewhere. The number of startups is exploding overall, so you'll see more and more unicorns over time. Technological innovations allow startups to grow faster. With new technology, unicorn startups can reach their customers quicker and reduce the time required for mass production. [3]

In this project, we're looking at the unicorn companies' data set to understand the growth based on valuation, location, industry, total fundraised, and investors. We will do data analysis and visualization of the Startup company that emerges as a Unicorn company. We will create a visualization to analyze how these companies have evolved over the years using python. These analyses and visualization provided insights about different unicorn companies and their growth based on various industries, valuations, and countries.

## Data set URLs and Data set Description

This data set is regarding Unicorn Companies Dataset. This data set is collected from Kaggle.

The URL to the data set is

<https://www.kaggle.com/datasets/deepcontractor/unicorn-companies-dataset>. This data set

contains the company name, valuation; date joined, country and city of origin, industry, investor details, financial stage, and total funds raised. It has all the details of different unicorn companies in other parts of the world. The dataset contains 1038 rows and 13 columns.

### Screenshot of the sample dataset

Company	Valuation	Date Joined	Country	City	Industry	Select Investors	Founded Year	Total Raised	Financial Stage	Investors Count	Deal Terms	Portfolio Exits
Bytedance	\$140	4/7/2017	China	Beijing	Artificial intelligence	Sequoia Capital, Cf	2012	\$7.44B	IPO	28	8	5
SpaceX	\$100.30	12/1/2012	United States	Hawthorne	Other	Founders Fund, Dr	2002	\$6.874B	None	29	12	None
Stripe	\$95	1/23/2014	United States	San Francisco	Fintech	Khosla Ventures, L	2010	\$2.901B	Asset	39	12	1
Klarna	\$45.60	12/12/2011	Sweden	Stockholm	Fintech	Institutional Venti	2005	\$3.472B	Acquired	56	13	1
Epic Games	\$42	10/26/2018	United States	Cary	Other	Tencent Holdings,	1991	\$4.377B	Acquired	25	5	2
Canva	\$40	1/8/2018	Australia	Surry Hills	Internet software &	Sequoia Capital, Cf	2012	\$571.26M	None	26	8	None
Checkout.com	\$40	5/2/2019	United Kingdom	London	Fintech	Tiger Global Mana	2012	\$1.83B	None	15	4	None
Instacart	\$39	12/30/2014	United States	San Francisco	Supply chain, logisti	Khosla Ventures, F	2012	\$2.686B	None	29	12	None
Revolut	\$33	4/26/2018	United Kingdom	London	Fintech	index Ventures, D	2015	\$1.716B	None	31	6	None
FTX	\$32	7/20/2021	Bahamas	Fintech	Sequoia Capital, Tho	None	2018	\$1.829B	Acq	40	3	1
Fanatics	\$27	6/6/2012	United States	Jacksonville	E-commerce & direct	SoftBank Group, A	1995	\$4.19B	None	21	10	None
Chime	\$25	3/5/2019	United States	San Francisco	Fintech	Forerunner Ventu	2013	\$2.599B	Divestiture	24	9	1
BYJU's	\$21	7/25/2017	India	Bengaluru	Edtech	Tencent Holdings,	2008	\$5.183B	None	45	19	None
Xiaohongshu	\$20	3/31/2016	China	Shanghai	E-commerce & direct	GGV Capital, Zhen	2013	\$917.5M	None	9	3	None
J&T Express	\$20	4/7/2021	Indonesia	Jakarta	Supply chain, logisti	Hillhouse Capital I	2015	\$4.653B	None	9	3	None
Miro	\$17.50	1/5/2022	United States	San Francisco	Internet software &	Accel, Altair Capit	2011	\$475M	None	18	1	None
Yuanfudao	\$15.50	5/31/2017	China	Beijing	Edtech	Tencent Holdings,	2012	\$4.044B	Acquired	18	7	1
DJI Innovatio	\$15	1/23/2015	China	Shenzhen	Hardware	Accel Partners, Sei	2006	\$1.135B	None	7	3	None
SHEIN	\$15	7/3/2018	China	Shenzhen	E-commerce & direct	Tiger Global Mana	2008	\$553.36M	None	8	3	None
goPuff	\$15	10/8/2020	United States	Philadelphia	E-commerce & direct	Accel, Softbank Gr	2013	\$3.397B	None	16	4	None
Yuanqi Senlir	\$15	3/1/2020	China	Beijing	Consumer & retail	Sequoia Capital, Cf	2016	\$721.31M	None	13	3	None

Figure 1: Screenshot of the sample dataset

### Data Description of each variable

Field Name	Data Description	Example
Company	It shows the name of the company	Name of a few unicorn companies are Bytedance, Canva, SHEIN, etc
Valuation (\$ in Billion)	It shows an estimation of company's worth in billions	The valuation of Bytedance company is \$140 billion
Date Joined	It shows the date on which company has founded	The company Bytedance was founded on 4/7/2017
Country	Shows the country of the company located	The company Bytedance is located in the country China

City	Shows the city of the company located	The Company Bytedance is located in Beijing city in China
Industry	Describes the type of industry the applicant will work in, like Research & Development, Insurance, Advertising & Marketing, Enterprise Software & Network Solutions, etc	The company Bytedance belongs to Artificial intelligence industry
Select Investors	It shows the companies invested in different companies	Investors of the company Bytedance are Sequoia Capital China, SIG Asia Investments, Sina Weibo, Softbank Group

Founded Year	It shows the year in which company was founded	The company Bytedance was founded in the year 2012
Total Raised	It shows the total fund raised by the company in dollars	The total fund raised by the company Bytedance is \$7.44 billion
Financial Stage	It shows the financial stage of the company	The financial stage of the company Bytedance is IPO
Investors Count	It shows the total number of investors in the company	28 investors invested in the company Bytedance
Deal Terms	It shows how many deals the company has	The company Bytedance has 8 deals in total
Portfolio Exits	It shows whether portfolio exists for the company or not	There is one portfolio for the company Bytedance

*Table 1: Table shows the data description of each variable we have in the data set*

## Data Cleaning:

---

Data cleaning is the process of identifying and fixing problems in a dataset. The purpose of data cleansing is to correct data that are inaccurate, incomplete, malformed, duplicated, or irrelevant to the purpose of the dataset. This is typically achieved by replacing, modifying, or deleting data that falls into one of these categories.

The data will likely be duplicated or mislabeled when combining multiple data sources. If the data is wrong, the results and algorithms are unreliable, even if they look correct. Since the process is different for each dataset, there is no absolute way to indicate the exact steps of the data cleansing process. Our decisions are usually based on datasets, so if the quality of the data is poor, our results will not be accurate. Therefore, data cleaning is essential because you can get high-quality data that leads to better quality decisions.

Not all data is good data in the data set. There was a little junk data. This dataset used for this analysis contained some null values. Some datasets had blank / missing values, so these data were deleted and filtered while focusing on the required data set. Unnecessary columns were removed, and a few cues were split. Below are the few steps taken to clean the dataset.

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. There is no one absolute way to prescribe the exact steps in the data cleaning process because the processes will



vary from dataset to dataset. But it is crucial to establish a template for the data cleaning process so we know we are doing it the right way every time. [4]

### **Check null values:**

In our dataset, there were NA/NAN values for the column “Valuation” and we filtered out those rows to get the consistency in our dataset and to get better results. For this process, we have used the in-built function to check if the NA value is present using `dataframe.isna()` then `dataframe.dropna()` to get the expected result.

### **Code:**

```
import pandas as pd
import numpy as np
uc = pd.read_csv('Unicorn_Companies.csv')
print(uc)
#Check if NaN is present in the dataframe
uc = uc.isna()
print(uc)
#Check if drop Nan values from dataframe
uc = uc.dropna()
print(uc)
```

### Before Screenshot:

	Company	Valuation (\$B)	...	Deal Terms	Portfolio Exits
0	False	False	...	False	False
1	False	False	...	False	False
2	False	False	...	False	False
3	False	True	...	False	False
4	False	False	...	False	False
5	False	True	...	False	False
6	False	False	...	False	False
7	False	False	...	False	False
8	False	False	...	False	False
9	False	False	...	False	False

Figure 2: The above shows the result of function “isna()” and as we can see it is showing *TRUE* for the value which is “NA”.

### After Screenshot:

	Company	Valuation (\$B)	...	Deal Terms	Portfolio Exits
0	False	False	...	False	False
1	False	False	...	False	False
2	False	False	...	False	False
3	False	False	...	False	False
4	False	False	...	False	False
...	...	...	...	...	...
1033	False	False	...	False	False
1034	False	False	...	False	False
1035	False	False	...	False	False
1036	False	False	...	False	False
1037	False	False	...	False	False

Figure 3: The above figure shows the result after running “dropna()” and it has removed the “NA” values from our dataset.

### Duplicate rows:

In this section, we have focused on removing the duplicate values from our dataset as it might affect our analysis. Deduplication is one of the largest areas to be considered in this process. For this process, we have used functions such as “drop\_duplicates()” to remove the duplicate values.

### Code:

```
import pandas as pd
```

```
import numpy as np

uc = pd.read_csv('Unicorn_Companies.csv')

print(uc.head(15))

#To remove duplicate values

uc = uc.drop_duplicates()

print(uc.head(15))
```

**Before:**

	Company	Valuation (\$B)	...	Deal Terms	Portfolio Exits
0	Bytedance	\$140	...	8	5
1	SpaceX	\$100.3	...	12	None
2	Stripe	\$95	...	12	1
3	Klarna	\$45.6	...	13	1
4	Epic Games	\$42	...	5	2
5	Canva	\$40	...	8	None
6	Checkout.com	\$40	...	4	None
7	Instacart	\$39	...	12	None
8	Instacart	\$39	...	12	None
9	Revolut	\$33	...	6	None
10	FTX	\$32	...	3	1
11	Fanatics	\$27	...	10	None
12	Fanatics	\$27	...	10	None
13	Crime	\$25	...	9	1
14	BYJU's	\$21	...	19	None

*Figure 4: The above screenshot shows that a duplicate row is present in the dataset so we have to remove that to get better results for our analysis.*

**After:**

	Company	Valuation (\$B)	...	Deal Terms	Portfolio Exits
0	Bytedance	\$140	...	8	5
1	SpaceX	\$100.3	...	12	None
2	Stripe	\$95	...	12	1
3	Klarna	\$45.6	...	13	1
4	Epic Games	\$42	...	5	2
5	Canva	\$40	...	8	None
6	Checkout.com	\$40	...	4	None
7	Instacart	\$39	...	12	None
9	Revolut	\$33	...	6	None
10	FTX	\$32	...	3	1
11	Fanatics	\$27	...	10	None
13	Chime	\$25	...	9	1
14	BYJU's	\$21	...	19	None
15	Xiaohongshu	\$20	...	3	None
16	J&T Express	\$20	...	3	None

*Figure 5: The above screenshot shows that after using the “drop\_duplicates()” function the duplicate data has been filtered out from the dataset.*

### **Changing the data type:**

In this section, we have focused on removing the ‘\$’ symbol from our dataset as it might affect our analysis. For this process, we removed \$ from the Valuation column and renamed the “Valuation (\$B)” column name to “Valuation”.

**Code:**

```
import pandas as pd

df = pd.read_csv('Unicorn_Companies.csv')

#print(df)

df = pd.DataFrame(df)

pd.set_option('display.max_rows',1000)

pd.set_option('display.max_columns',10)
```

```

pd.set_option('display.max_colwidth',100)

pd.set_option('display.width', None)

#print(df)

#Rename Column "Valuation ($B)" to "Valuation" and remove "$B" from the column
df.rename(columns = {'Valuation ($B)' : 'Valuation'}, inplace = True)

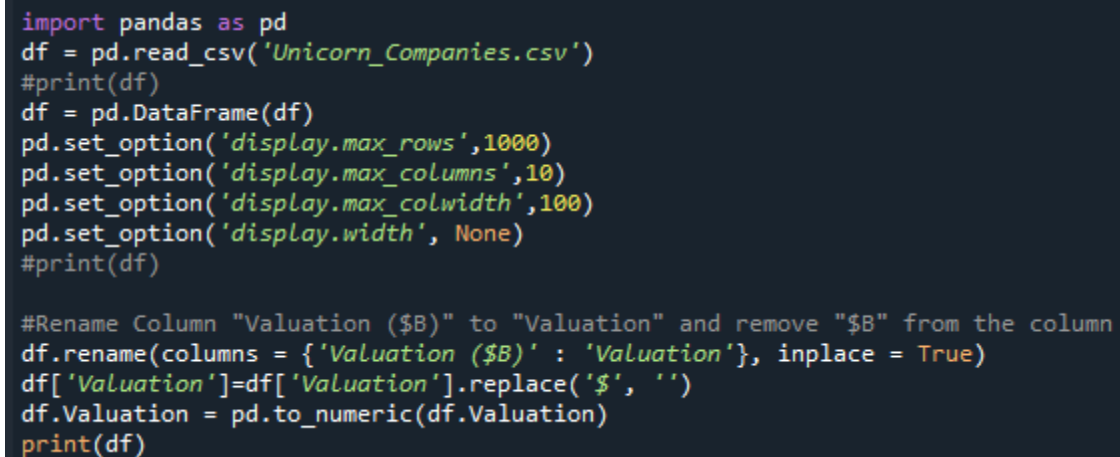
df['Valuation']=df['Valuation'].replace('$', '')

df.Valuation = pd.to_numeric(df.Valuation)

print(df)

```

### Screenshot of the code



```

import pandas as pd
df = pd.read_csv('Unicorn_Companies.csv')
#print(df)
df = pd.DataFrame(df)
pd.set_option('display.max_rows',1000)
pd.set_option('display.max_columns',10)
pd.set_option('display.max_colwidth',100)
pd.set_option('display.width', None)
#print(df)

#Rename Column "Valuation ($B)" to "Valuation" and remove "$B" from the column
df.rename(columns = {'Valuation ($B)' : 'Valuation'}, inplace = True)
df['Valuation']=df['Valuation'].replace('$', '')
df.Valuation = pd.to_numeric(df.Valuation)
print(df)

```

Figure 6: The above screenshot shows the code for data cleaning for the column “valuation”

## Before Data Cleaning

	Company	Valuation (\$B)	Date Joined	Country	City
0	Bytedance	\$140	04/07/17	China	Beijing
1	SpaceX	\$100.3	12/01/12	United States	Hawthorne
2	Stripe	\$95	1/23/2014	United States	San Francisco
3	Klarna	\$45.6	12/12/11	Sweden	Stockholm
4	Epic Games	\$42	10/26/2018	United States	Cary
...	...	...	...	...	...
1031	Timescale	\$1	2/22/2022	United States	New York
1032	Scalapay	\$1	2/23/2022	Italy	Milan
1033	Omada Health	\$1	2/23/2022	United States	San Francisco
1034	BlueVoyant	\$1	2/23/2022	United States	New York
1035	Veeva	\$1	2/24/2022	United States	San Mateo

Figure 7: The above screenshot shows the details of the column “Valuation” before data cleaning

## After Data cleaning

	Company	Valuation	Date Joined	Country	City
0	Bytedance	140.0	4/7/2017	China	Beijing
1	SpaceX	100.3	12/1/2012	United States	Hawthorne
2	Stripe	95.0	1/23/2014	United States	San Francisco
3	Klarna	45.6	12/12/2011	Sweden	Stockholm
4	Epic Games	42.0	10/26/2018	United States	Cary
...	...	...	...	...	...
1032	Timescale	1.0	2/22/2022	United States	New York
1033	Scalapay	1.0	2/23/2022	Italy	Milan
1034	Omada Health	1.0	2/23/2022	United States	San Francisco
1035	BlueVoyant	1.0	2/23/2022	United States	New York
1036	Veeva	1.0	2/24/2022	United States	San Mateo

Figure 8: The above screenshot shows that details of the column “Valuation” after data cleaning

## Summary Statistics

Below is the statistical analysis of our dataset columns. For the statistical analysis, we have used the in-built function called describe, and we have applied separate statistical analyses such as mean, min, max, etc.

### 1. Column Valuation:

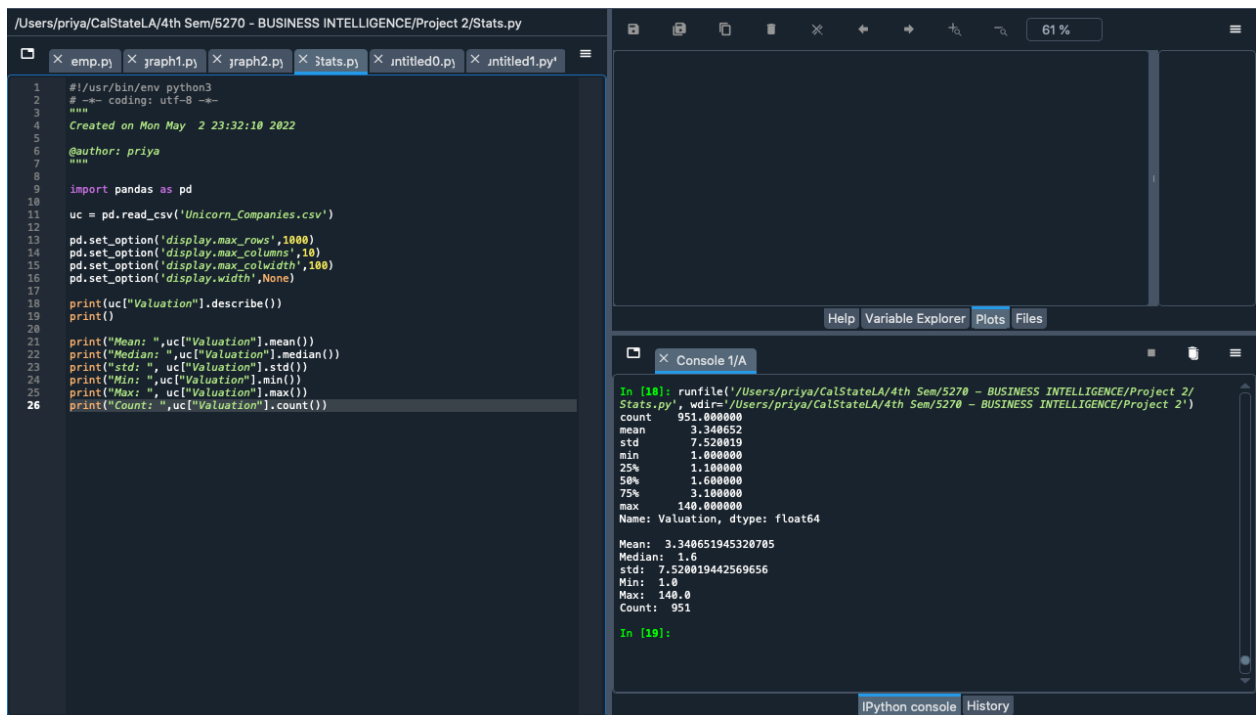


Figure 9: Screenshot of Valuation Column Statistics.

```

import pandas as pd

uc = pd.read_csv('Unicorn_Companies.csv')

pd.set_option('display.max_rows',1000)
pd.set_option('display.max_columns',10)
pd.set_option('display.max_colwidth',100)
pd.set_option('display.width',None)

print(uc["Valuation"].describe())
print()

print("Mean: ",uc["Valuation"].mean())
print("Median: ",uc["Valuation"].median())
print("std: ", uc["Valuation"].std())
print("Min: ",uc["Valuation"].min())
print("Max: ", uc["Valuation"].max())
print("Count: ",uc["Valuation"].count())

```

Figure 10: Screenshot of Valuation Column Statistics code.

```

count    951.000000
mean      3.340652
std       7.520019
min       1.000000
25%      1.100000
50%      1.600000
75%      3.100000
max      140.000000
Name: Valuation, dtype: float64

Mean: 3.340651945320705
Median: 1.6
std: 7.520019442569656
Min: 1.0
Max: 140.0
Count: 951

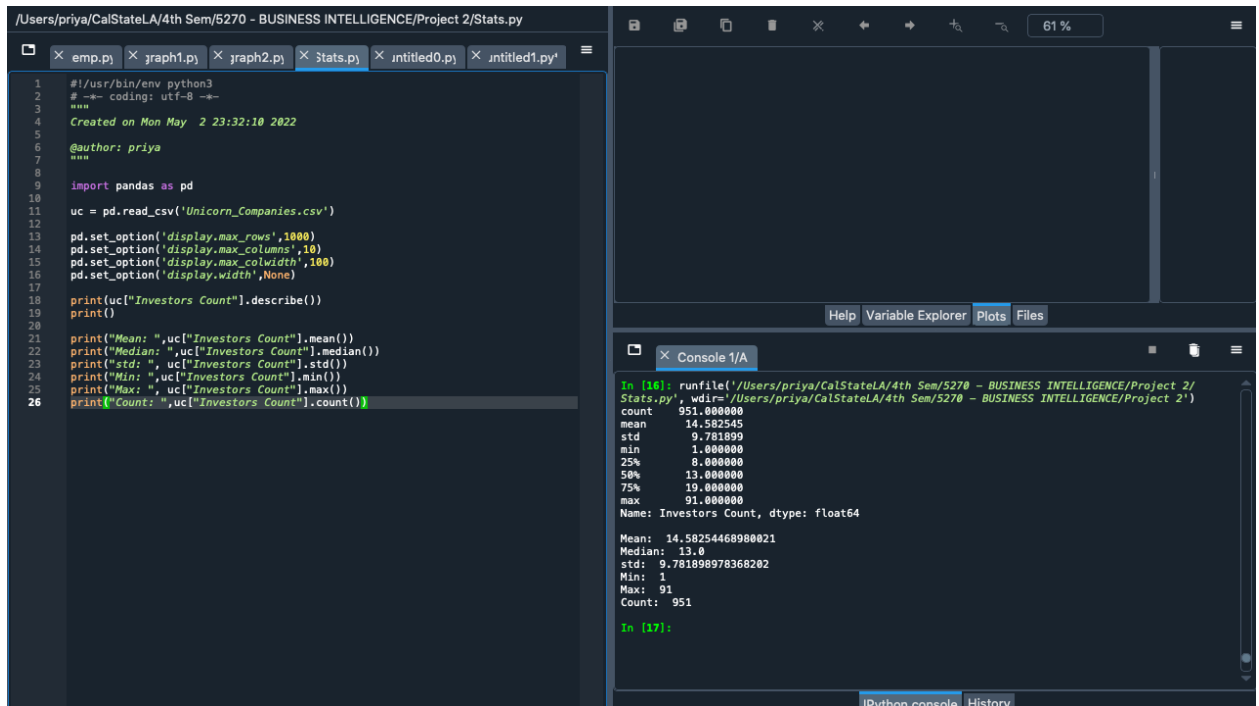
```

Figure 11: Screenshot of Valuation Column Statistics result.

The above screenshot represents the statistical analysis for the column Valuation. To get the analysis, we have used the in-built function describe and other in-built functions such as mean, min, etc. As we can see, the total row count is 957, and the mean states that the average valuation of companies is 3.33B, where the minimum is 1.1B, and the max is 140.0B. The standard deviation for this column is 7.49, and by looking at the 1st percentile and 2nd, we can say that the valuation is not spread that much.



## 2. Column Investors Count



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  """
4  Created on Mon May 2 23:32:10 2022
5
6  @author: priya
7  """
8
9  import pandas as pd
10
11  uc = pd.read_csv('Unicorn_Companies.csv')
12
13  pd.set_option('display.max_rows',1000)
14  pd.set_option('display.max_columns',10)
15  pd.set_option('display.max_colwidth',100)
16  pd.set_option('display.width',None)
17
18  print(uc["Investors Count"].describe())
19  print()
20
21  print("Mean: ",uc["Investors Count"].mean())
22  print("Median: ",uc["Investors Count"].median())
23  print("std: ", uc["Investors Count"].std())
24  print("Min: ",uc["Investors Count"].min())
25  print("Max: ", uc["Investors Count"].max())
26  print("Count: ",uc["Investors Count"].count())
```

Console 1/A

```
In [16]: runfile('/Users/priya/CalStateLA/4th Sem/5270 - BUSINESS INTELLIGENCE/Project 2/Stats.py', wdir='/Users/priya/CalStateLA/4th Sem/5270 - BUSINESS INTELLIGENCE/Project 2')
count      951.000000
mean       14.582545
std         9.781899
min          1.000000
25%          8.000000
50%         13.000000
75%         19.000000
max         91.000000
Name: Investors Count, dtype: float64

Mean: 14.58254468988021
Median: 13.0
std: 9.781898978368202
Min: 1
Max: 91
Count: 951

In [17]:
```

Figure 12: Screenshot of Investors Count Column Statistics.

```
"""
Created on Mon May 2 23:32:10 2022

@author: priya
"""

import pandas as pd

uc = pd.read_csv('Unicorn_Companies.csv')

pd.set_option('display.max_rows',1000)
pd.set_option('display.max_columns',10)
pd.set_option('display.max_colwidth',100)
pd.set_option('display.width',None)

print(uc["Investors Count"].describe())
print()

print("Mean: ",uc["Investors Count"].mean())
print("Median: ",uc["Investors Count"].median())
print("std: ", uc["Investors Count"].std())
print("Min: ",uc["Investors Count"].min())
print("Max: ", uc["Investors Count"].max())
print("Count: ",uc["Investors Count"].count())
```

*Figure 13: Screenshot of Investors Count Column Statistics code.*

```
count    951.000000
mean     14.582545
std      9.781899
min      1.000000
25%      8.000000
50%     13.000000
75%     19.000000
max     91.000000
Name: Investors Count, dtype: float64

Mean: 14.58254468980021
Median: 13.0
std: 9.781898978368202
Min: 1
Max: 91
Count: 951
```

*Figure 14: Screenshot of Investors Count Column Statistics result.*

The above screenshot represents the statistical analysis for the column Investors Count. To get the analysis, we have used the in-built function describe and other in-built functions such as mean, min, etc. As we can see, the total row count is 957, and the mean states that the average investors of companies are 14.50, where the minimum is one, and the max is 91. The standard division for this column is 9.79, and by looking at the 1st percentile and 2nd, we can say that the investor count is spread out a lot from 1 to 19.

### 3. Column Total Raised

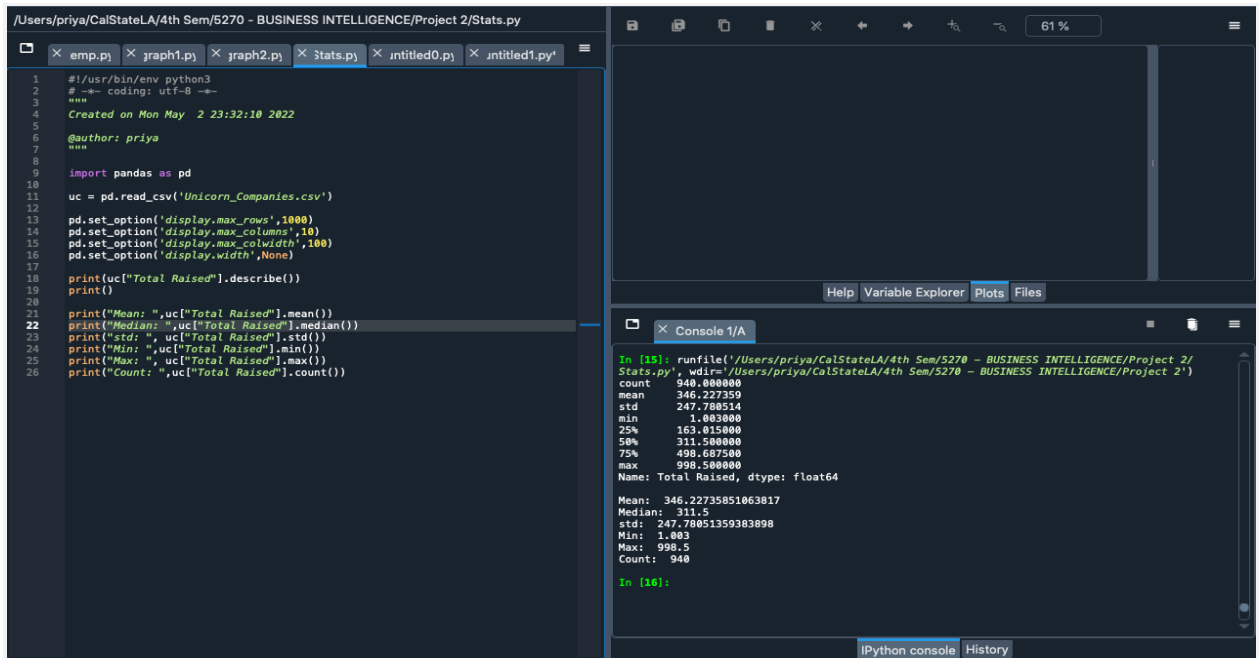


Figure 15: Screenshot of Total Raised Column Statistics.



Figure 16: Screenshot of Total Raised Column Statistics code.

```
count    940.000000
mean     346.227359
std      247.780514
min       1.003000
25%      163.015000
50%      311.500000
75%      498.687500
max      998.500000
Name: Total Raised, dtype: float64

Mean: 346.22735851063817
Median: 311.5
std: 247.78051359383898
Min: 1.003
Max: 998.5
Count: 940

In [16]: |
```

*Figure 17: Screenshot of Total Raised Column Statistics result.*

The above screenshot represents the statistical analysis for the column Total Raised. To get the analysis we have used the in-built function describe and other in-built functions such as mean, min, etc. As we can see the total row count is 940 and the mean states that the average valuation of companies is 346.22B, where the minimum is 1.0B and the max is 998.5B. The standard deviation for this column is 247.78B and by looking at the 1st percentile and 2nd we can say that the Total Raised is spread out from 163.0B to 998.5B.

# Data Visualization

Data visualization is a graphical representation of information and data. By using visual elements such as charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in your data. In the world of big data, data visualization tools and technologies are essential to analyze vast amounts of information and make informed decisions. Once the dataset is cleaned, it is now possible to start loading the dataset into a data visualization software and model the data. The software used was python to create visually appealing graphs and charts that can help discover insights not possible by simply viewing the dataset.

## 1. What are the top 10 most valued unicorn companies?

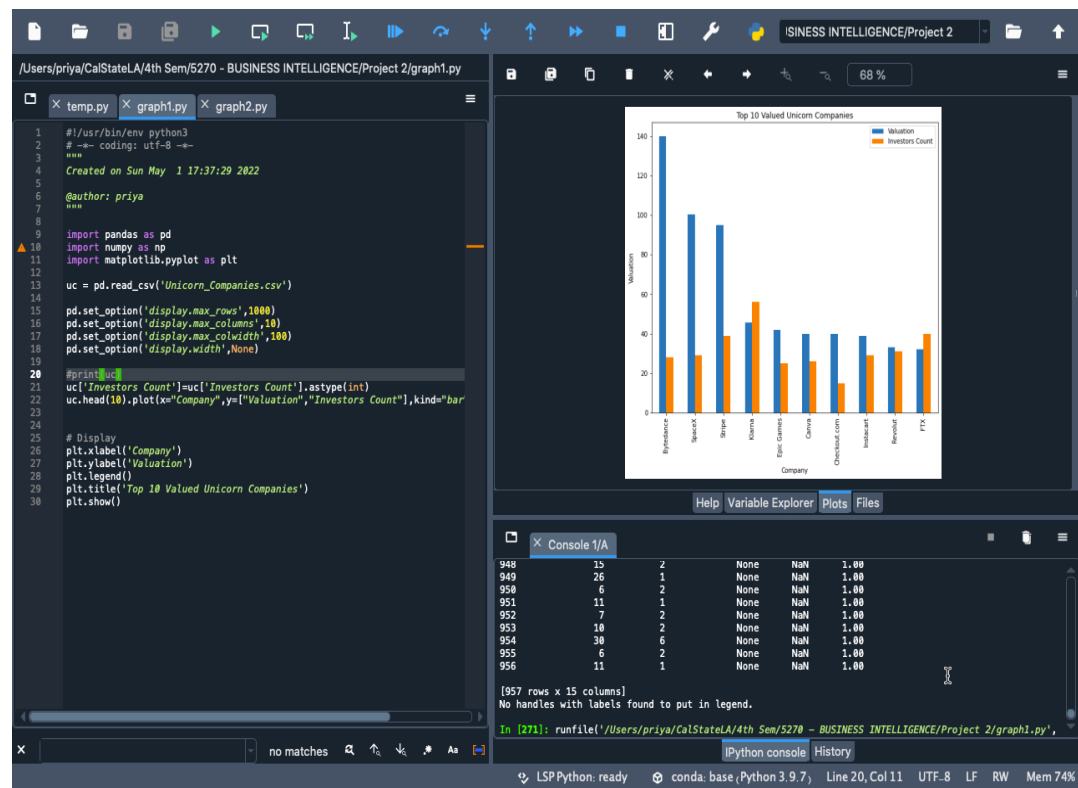


Figure 18: Top Ten Valued unicorn Companies

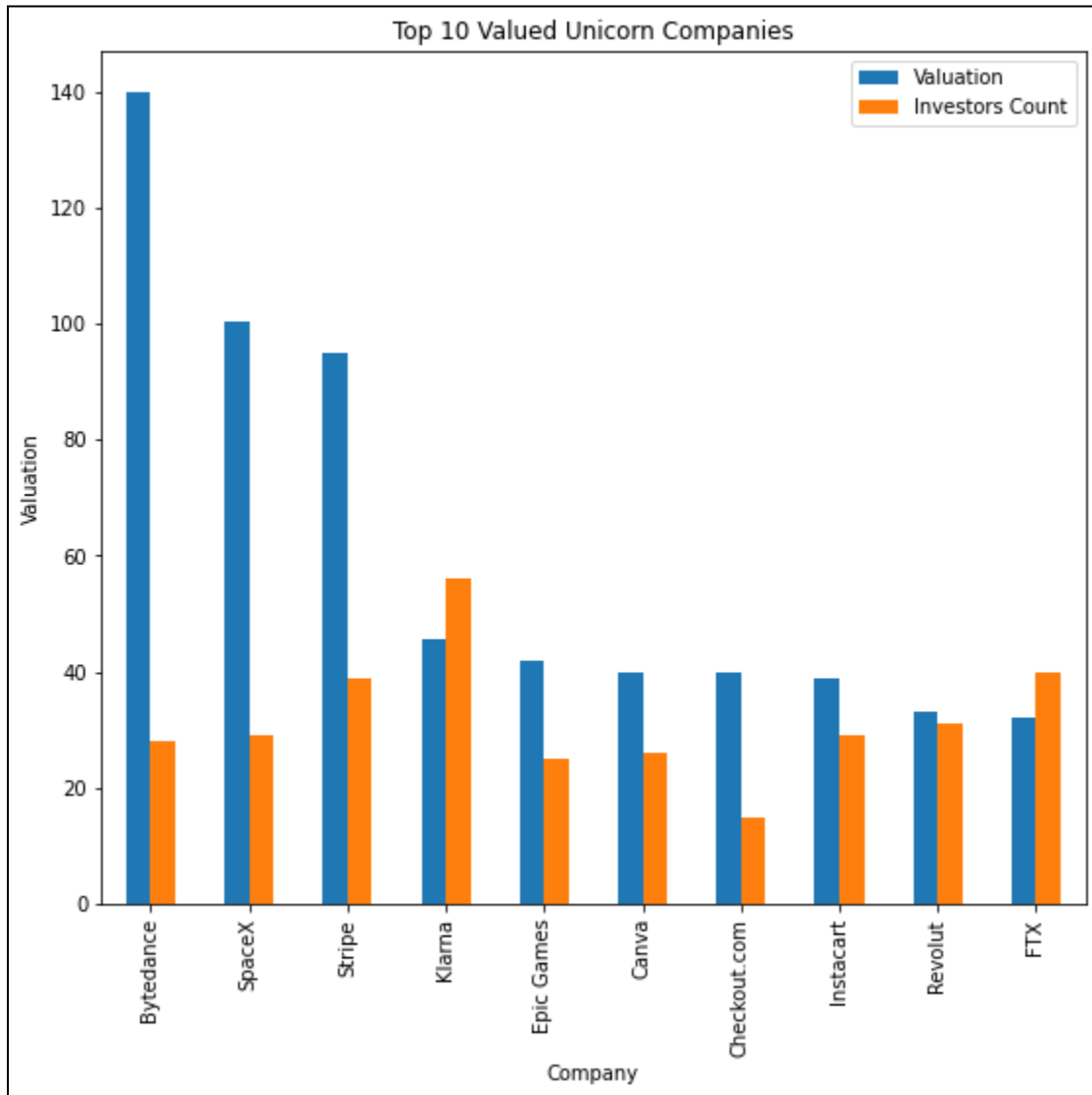


Figure 19: Top Ten Valued unicorn Companies

### Python Features used:

- **Packages/Libraries:** pandas, NumPy, matplotlib
- **read\_csv():** To read the dataset from the csv file.
- **set\_option():** To make changes for the number of rows and columns to display.

- **astype(int):** To convert the column type from string to integer type.
- **plot():** To plot the bar graph for the analysis.
- **xlabel():** To add the x-axis label.
- **ylabel():** To add the y-axis label.
- **title():** To add the title for the chart.
- **show():** To display the chart.
- **legend():** To show the legends for the chart.

**Code:**

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

uc = pd.read_csv('Unicorn_Companies.csv')

pd.set_option('display.max_rows',1000)
pd.set_option('display.max_columns',10)
pd.set_option('display.max_colwidth',100)
pd.set_option('display.width',None)

#print(uc)

uc['Investors Count']=uc['Investors Count'].astype(int)

uc.head(10).plot(x="Company",y=["Valuation","Investors
Count"],kind="bar",figsize=(9,8))

# Display

plt.xlabel('Company')
```

```
plt.ylabel('Valuation')

plt.legend()

plt.title('Top 10 Valued Unicorn Companies')

plt.show()
```

### Screenshot of the Code

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  """
4  Created on Sun May  1 17:37:29 2022
5
6  @author: priya
7  """
8
9  import pandas as pd
10 import numpy as np
11 import matplotlib.pyplot as plt
12
13 uc = pd.read_csv('Unicorn_Companies.csv')
14
15 pd.set_option('display.max_rows',1000)
16 pd.set_option('display.max_columns',10)
17 pd.set_option('display.max_colwidth',100)
18 pd.set_option('display.width',None)
19
20 #print(uc)
21 uc['Investors Count']=uc['Investors Count'].astype(int)
22 uc = uc.sort_values(by="Valuation", ascending=False)
23 uc.head(10).plot(x="Company",y=["Valuation","Investors Count"],kind="bar",figsize=(9,8))
24 print(uc.head(10))
25
26 # Display
27 plt.xlabel('Company')
28 plt.ylabel('Valuation')
29 plt.legend()
30 plt.title('Top 10 Valued Unicorn Companies')
31 plt.show()
```

*Figure 20: Screenshot of the Code*

### Insight:

The above analysis represents the top ten valued unicorns. For this analysis, we have used a bar chart. The blue bar represents the company's valuation, and the orange one represents the company's investor count. As we can see from the graph, the Bytedance company has the highest



company valuation with 140.0B, SpaceX with 100.3B valuation, Stripe with 95.0B, and so on.

Klama has the highest number of investors compared to others, with a count of 56, followed by

FTX, Stripe, etc. And from the above analysis, we can also observe that the company

Checkout.com has the lowest investor count.

## 2. Which industry has the most unicorn companies and the minor ones?

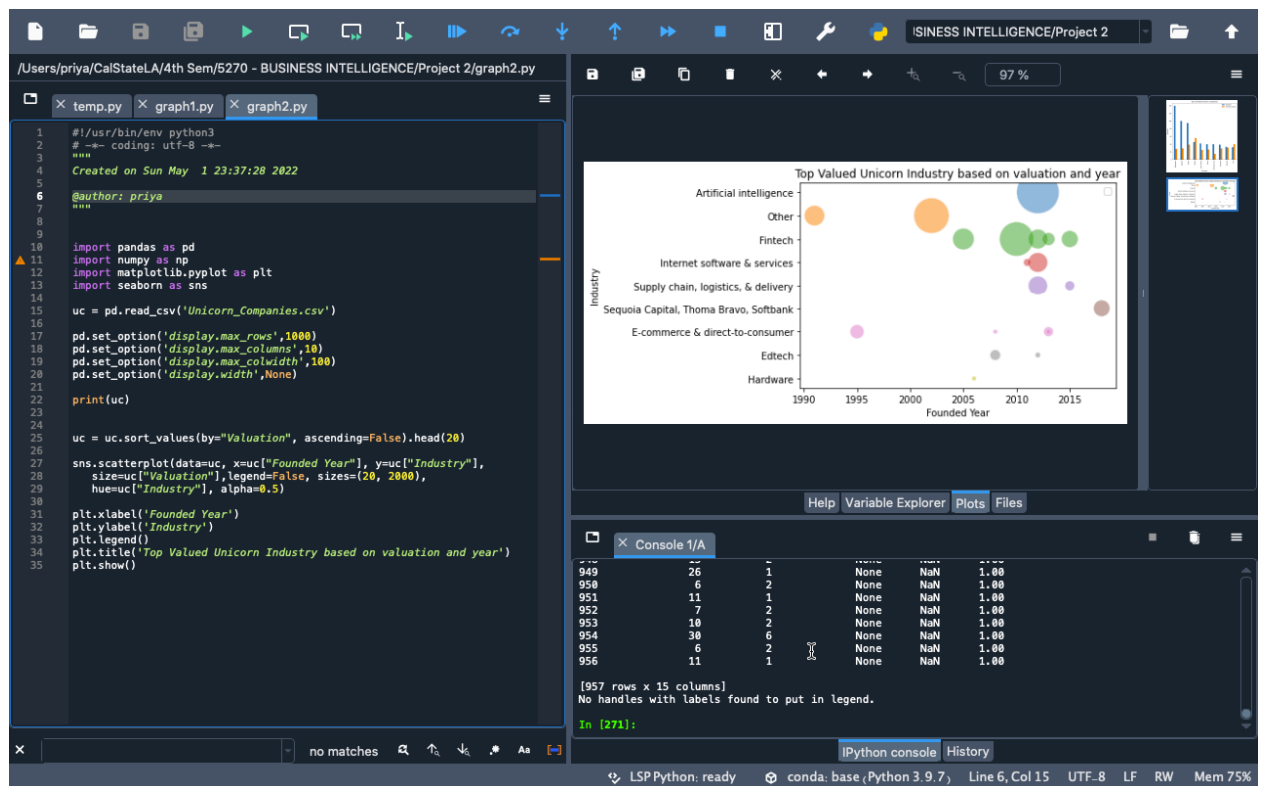


Figure 21: Top valued Unicorn Industry based on Company Valuation and year

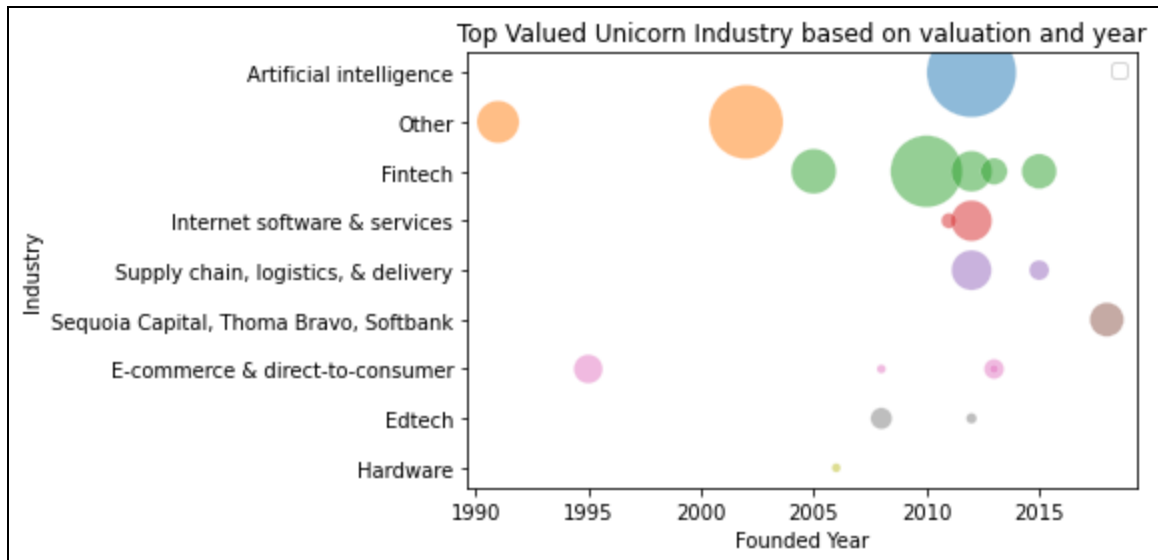


Figure 22: Top valued Unicorn Industry based on Company Valuation and year

#### Python Features used:

- **Packages/Libraries:** pandas, numpy, matplotlib, seaborn
- **read\_csv():** To read the dataset from the csv file.
- **set\_option():** To make changes for the number of rows and columns to display.
- **sort\_values():** To sort the data frame based on a specific column.
- **plot():** To plot the bar graph for the analysis.
- **xlabel():** To add the x-axis label.
- **ylabel():** To add the y-axis label.
- **title():** To add the title for the chart.
- **show():** To display the chart.
- **legend():** To show the legends for the chart.
- **sns.scatterplot():** To display the analysis in the scatterplot.

#### Code:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

uc = pd.read_csv('Unicorn_Companies.csv')

pd.set_option('display.max_rows',1000)

pd.set_option('display.max_columns',10)

pd.set_option('display.max_colwidth',100)

pd.set_option('display.width',None)

print(uc)

uc = uc.sort_values(by="Valuation", ascending=False).head(20)

sns.scatterplot(data=uc, x=uc["Founded Year"], y=uc["Industry"],

               size=uc["Valuation"],legend=False, sizes=(20, 2000),

               hue=uc["Industry"], alpha=0.5)

plt.xlabel('Founded Year')

plt.ylabel('Industry')

plt.legend()

plt.title('Top Valued Unicorn Industry based on valuation and year')

plt.show()
```

**Screenshot of the code**

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  """
4  Created on Sun May 1 23:37:28 2022
5
6  @author: priya
7  """
8
9
10 import pandas as pd
11 import numpy as np
12 import matplotlib.pyplot as plt
13 import seaborn as sns
14
15 uc = pd.read_csv('Unicorn_Companies.csv')
16
17 pd.set_option('display.max_rows',1000)
18 pd.set_option('display.max_columns',10)
19 pd.set_option('display.max_colwidth',100)
20 pd.set_option('display.width',None)
21
22 print(uc)
23
24
25 uc = uc.sort_values(by="Valuation", ascending=False).head(20)
26 sns.scatterplot(data=uc, x=uc["Founded Year"], y=uc["Industry"],
27                 size=uc["Valuation"], legend=False, sizes=(20, 2000),
28                 hue=uc["Industry"], alpha=0.5)
29
30 plt.xlabel('Founded Year')
31 plt.ylabel('Industry')
32 plt.legend()
33 plt.title('Top Valued Unicorn Industry based on valuation and year')
34 plt.show()

```

*Figure 23: Screenshot of the Code*

### **Insight:**

The above analysis represents the Top valued Unicorn Industry based on Company Valuation and year. We have used the scatter plot to show the Bubble chart representation for this analysis. The x-axis represents the Year the company was founded, and the y-axis represents the top industries in demand. As we can observe from the above result, Artificial intelligence is on the top based on the valuation, and we can say that the value is showing a positive relationship followed by the other technology companies, the Fintech industry, which has grown over the year, Internet & Services and so on. From the analysis, we can also state that the software industry has more valuation than the hardware industry.

### 3. Valuation based on top 10 Country and top 10 City

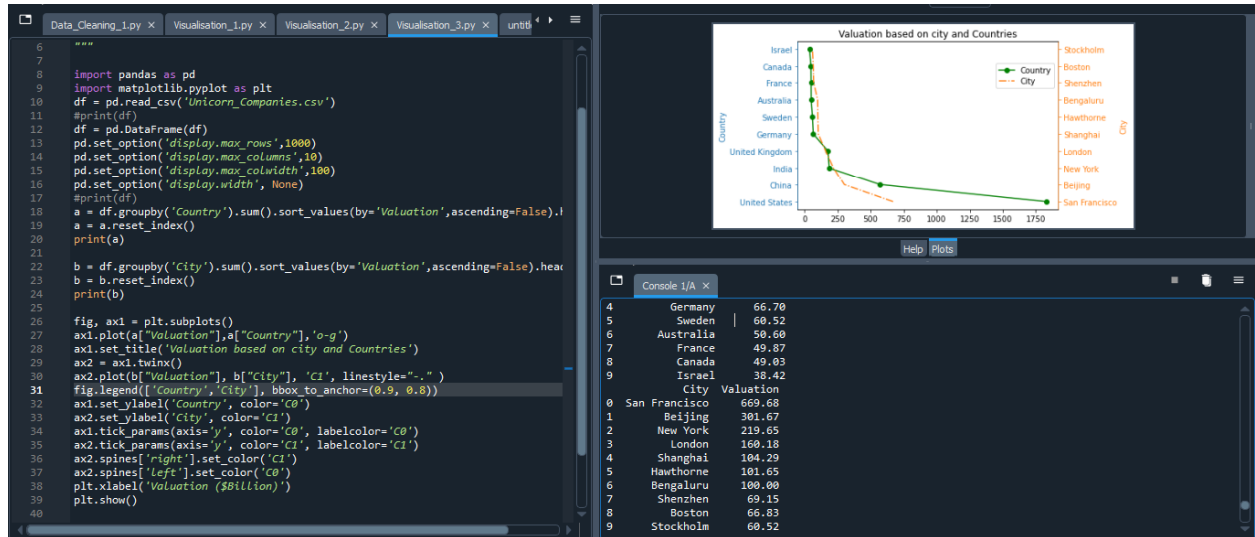


Figure 24: Valuation based on top ten countries and cities

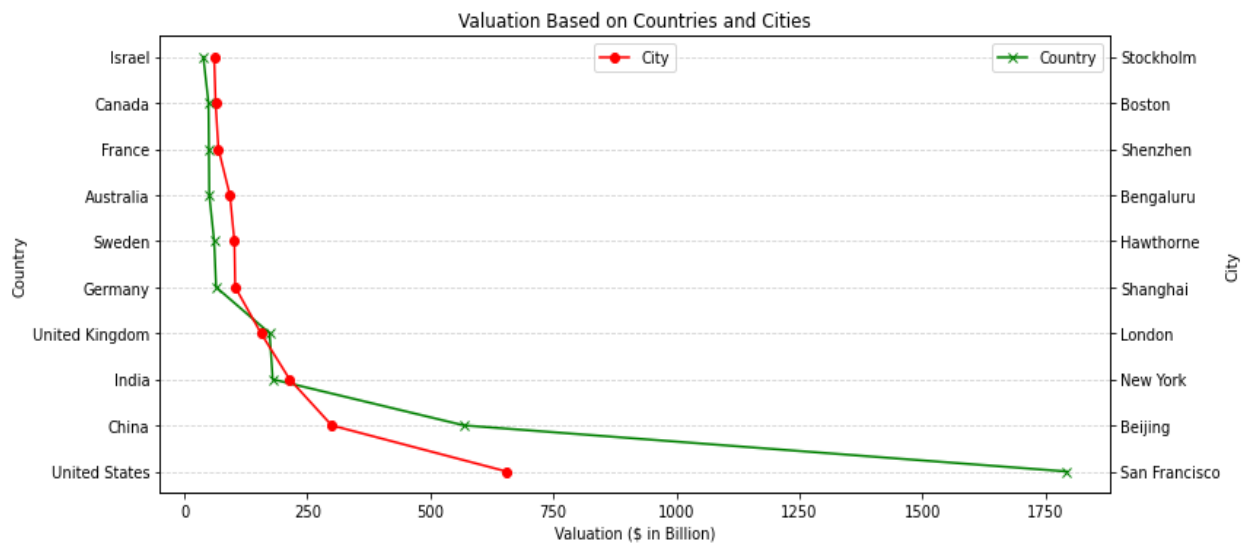


Figure 25: Dual axis chart shows the valuation based on top ten countries and cities

#### Python Features used:

- **Packages/Libraries:** pandas, matplotlib

- **read\_csv():** To read the dataset from the CSV file.
- **set\_option():** To make changes to the number of rows and columns to display.
- **plot():** To plot the bar graph for the analysis.
- **groupby():** To group the data frame based on a specific column.
- **sort\_value:** To sort the data frame based on a specific column.
- **xlabel():** To add the x-axis label.
- **ylabel():** To add the y-axis label.
- **title():** To add the title for the chart.
- **show():** To display the chart
- **legend():** To show the legends for the chart.

**Code:**

```
import pandas as pd

import matplotlib.pyplot as plt

df = pd.read_csv('Unicorn_Companies.csv')

#print(df)

df = pd.DataFrame(df)

pd.set_option('display.max_rows',1000)

pd.set_option('display.max_columns',10)

pd.set_option('display.max_colwidth',100)

pd.set_option('display.width', None)

#print(df)

a = df.groupby('Country').sum().sort_values(by='Valuation',ascending=False).head(10)
```

```
a = a.reset_index()

print(a)

b = df.groupby('City').sum().sort_values(by='Valuation',ascending=False).head(10)
b = b.reset_index()
print(b)

fig, ax = plt.subplots(figsize=(12,5))
ax2 = ax.twinx()
ax.set_title('Valuation Based on Countries and Cities')
ax.set_xlabel('Valuation ($ in Billion)')
ax.plot(a['Valuation'], a['Country'], color='green', marker='x')
ax2.plot(b['Valuation'], b['City'], color='red', marker='o')
ax.set_ylabel('Country')
ax2.set_ylabel('City')
ax.legend(['Country'])
ax2.legend(['City'], loc='upper center')
ax.yaxis.grid(color='lightgray', linestyle='dashed')

plt.tight_layout()

plt.show()
```

### Screenshot of the code:

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('Unicorn_Companies.csv')
#print(df)
df = pd.DataFrame(df)
pd.set_option('display.max_rows',1000)
pd.set_option('display.max_columns',10)
pd.set_option('display.max_colwidth',100)
pd.set_option('display.width', None)
#print(df)
a = df.groupby('Country').sum().sort_values(by='Valuation',ascending=False).head(10)
a = a.reset_index()
print(a)

b = df.groupby('City').sum().sort_values(by='Valuation',ascending=False).head(10)
b = b.reset_index()
print(b)

fig, ax = plt.subplots(figsize=(12,5))
ax2 = ax.twinx()
ax.set_title('Valuation Based on Countries and Cities')
ax.set_xlabel('Valuation ($ in Billion)')
ax.plot(a['Valuation'], a['Country'], color='green', marker='x')
ax2.plot(b['Valuation'], b['City'], color='red', marker='o')
ax.set_ylabel('Country')
ax2.set_ylabel('City')
ax.legend(['Country'])
ax2.legend(['City'], loc='upper center')
ax.yaxis.grid(color='lightgray', linestyle='dashed')
plt.tight_layout()
plt.show()
```

Figure 26: Screenshot of the Code

### Insight:

The above analysis represents the valuation based on the top ten countries and top 10 cities. We have used a dual-axis chart to represent this analysis. The x-axis represents the valuation in billions. The left y-axis represents the names of countries and the right y-axis represents the names of cities. From the result, we can observe that the United States followed by China and India has the highest valuation in terms of country. Of the cities, San Francisco has the highest valuation. The top ten countries with the highest valuation are the United States, China, India,



United Kingdom, Germany, Sweden, Australia, France, Canada, and Israel. The top cities with the highest valuation are San Francisco, Beijing, New York, London, Shanghai, Hawthorne, Bengaluru, Shenzhen, Boston, and Stockholm. From the analysis, we can state that the United States has the highest valuation of other countries.

#### 4. Top five cities with the highest number of Unicorn Companies

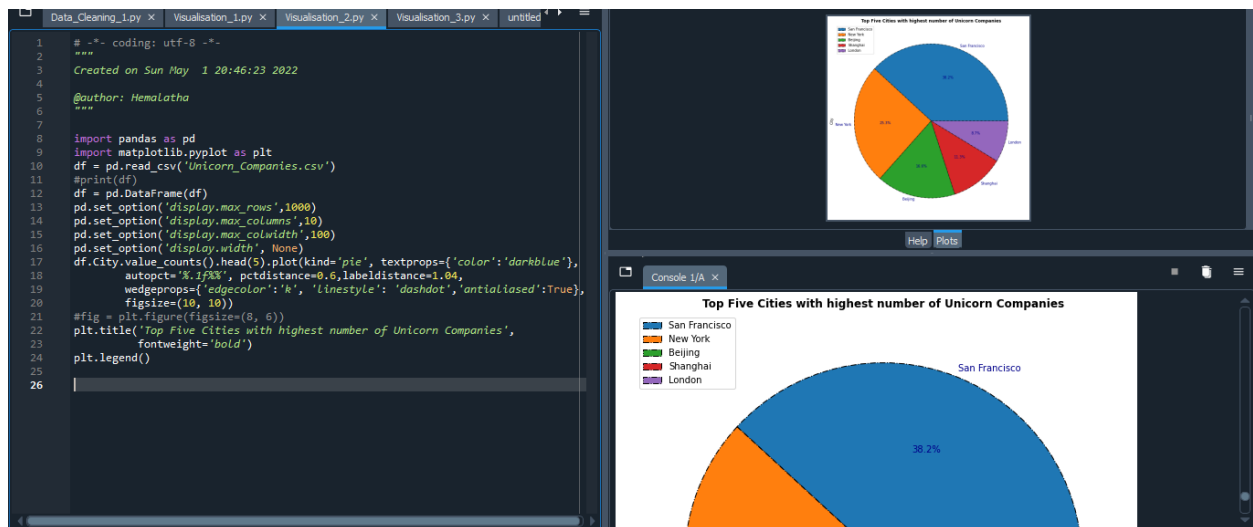
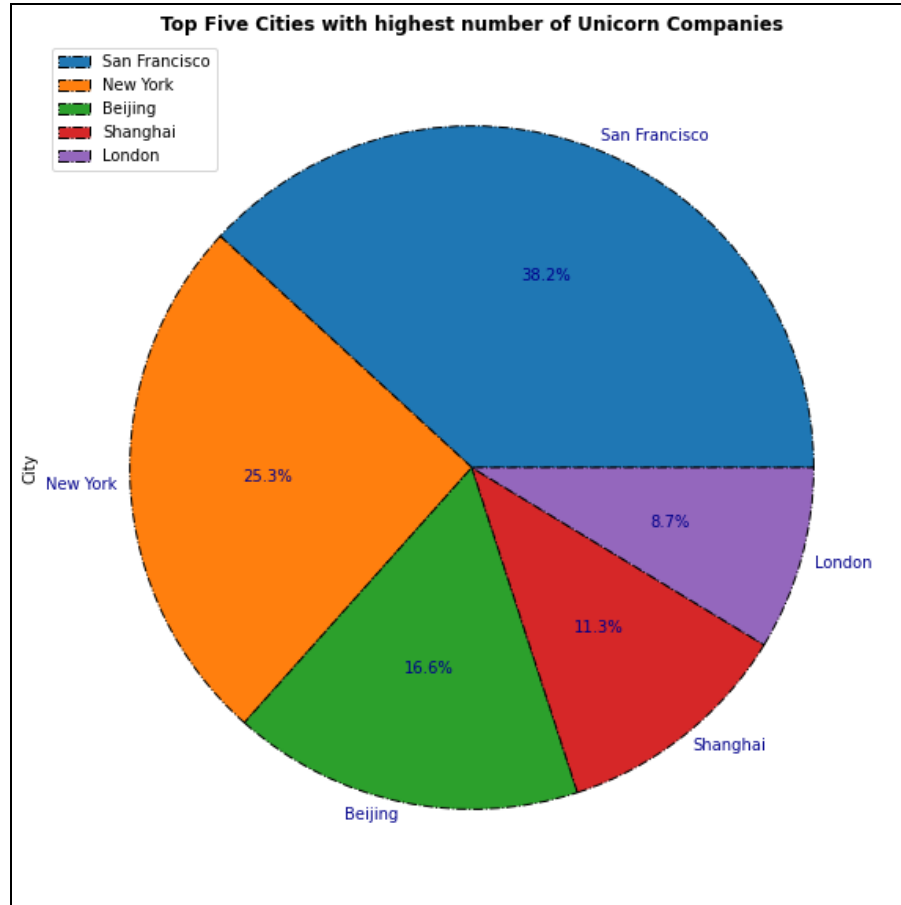


Figure 27: The top five cities with the highest number of Unicorn Companies



*Figure 28: Bar graph shows the top five cities with highest number of Unicorn Companies*

#### **Python Features used:**

- **Packages/Libraries:** pandas, matplotlib
- **read\_csv():** To read the dataset from the CSV file
- **set\_option():** To make changes to the number of rows and columns to display
- **plot():** To plot the bar graph for the analysis
- **title():** To add the title for the chart
- **show():** To display the chart

- **legend():** To show the legends for the chart

**Code:**

```
import pandas as pd

import matplotlib.pyplot as plt

df = pd.read_csv('Unicorn_Companies.csv')

#print(df)

df = pd.DataFrame(df)

pd.set_option('display.max_rows',1000)

pd.set_option('display.max_columns',10)

pd.set_option('display.max_colwidth',100)

pd.set_option('display.width', None)

df.City.value_counts().head(5).plot(kind='pie', textprops={'color':'darkblue'},

    autopct='%0.1f%%', pctdistance=0.6,labeldistance=1.04,

    wedgeprops={'edgecolor':'k', 'linestyle': 'dashdot','antialiased':True},

    figsize=(10, 10))

plt.title('Top Five Cities with highest number of Unicorn Companies',

    fontweight='bold')

plt.legend()
```

### Screenshot of the Code:

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('Unicorn_Companies.csv')
#print(df)
df = pd.DataFrame(df)
pd.set_option('display.max_rows',1000)
pd.set_option('display.max_columns',10)
pd.set_option('display.max_colwidth',100)
pd.set_option('display.width', None)
df.City.value_counts().head(5).plot(kind='pie', textprops={'color':'darkblue'},
    autopct='%.1f%%', pctdistance=0.6,labeldistance=1.04,
    wedgeprops={'edgecolor':'k', 'linestyle': 'dashdot','antialiased':True},
    figsize=(10, 10))
plt.title('Top Five Cities with highest number of Unicorn Companies',
    fontweight='bold')
plt.legend()
```

*Figure 29: Screenshot of the Code*

### Insight:

The above analysis represents the top five cities with the highest number of Unicorn companies. We have used pie charts to represent this analysis. From the result, we can observe that San Francisco has the highest number of unicorn companies. The top five cities with the highest number of unicorn companies are San Francisco, New York, Beijing, Shanghai, and London. From the analysis, we can state that San Francisco has the highest number of unicorn companies. San Francisco has become the hub for Unicorn companies.

## Reference

---

1. **Article title:** What is a unicorn company? **URL:**  
<https://pitchbook.com/blog/what-is-a-unicorn>, **Date accessed:** May 1, 2022.
2. **Article title:** What Are Unicorn Companies? **URL:**  
<https://www.sofi.com/learn/content/what-is-a-unicorn-company/>, **Date published:** November 08, 2021, **Date accessed:** May 1, 2022.
3. **Article title:** What Makes Unicorns Special? These Numbers May Hold the Answers.,  
**URL:**<https://www.gsb.stanford.edu/insights/what-makes-unicorns-special-these-numbers-may-hold-answers>, **Date published:** December 10, 2021, **Date accessed:** May 1, 2022.
4. **Article title:** Guide To Data Cleaning: Definition, Benefits, Components, And How To Clean Your Data, **URL:** <https://www.tableau.com/learn/articles/what-is-data-cleaning>, **Date published:** 2021, December 14