Analysis of Unicorn Companies

- The dataset contains 1074 records of private companies in different countries with a valuation of over 1 billion as of March 2022.It includes the current valuation, funding, country of origin, industry, select investor and the years they were founded and year the became unicorns.

In [15]:
```
#import the necessary libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
unicorn_analysis = pd.read_csv
unicorn_analysis = pd.read_csv(r'C:\Users\adora\OneDrive\Desktop\Quantum Analytics'
unicorn_analysis
```

Out[15]:

	Company	Valuation	Date Joined	Industry	City	Country	Continent	Year Founded	Fu
0	Bytedance	$180B	07/04/2017	Artificial intelligence	Beijing	China	Asia	2012	
1	SpaceX	$100B	01/12/2012	Other	Hawthorne	United States	North America	2002	
2	SHEIN	$100B	03/07/2018	E-commerce & direct-to-consumer	Shenzhen	China	Asia	2008	
3	Stripe	$95B	23/01/2014	Fintech	San Francisco	United States	North America	2010	
4	Klarna	$46B	12/12/2011	Fintech	Stockholm	Sweden	Europe	2005	
...	...	...	...	...	...	...	...	...	
1069	Zhaogang	$1B	29/06/2017	E-commerce & direct-to-consumer	Shanghai	China	Asia	2012	
1070	Zhuan Zhuan	$1B	18/04/2017	E-commerce & direct-to-consumer	Beijing	China	Asia	2015	
1071	Zihaiguo	$1B	06/05/2021	Consumer & retail	Chongqing	China	Asia	2018	
1072	Zopa	$1B	19/10/2021	Fintech	London	United Kingdom	Europe	2005	
1073	Zwift	$1B	16/09/2020	E-commerce & direct-to-consumer	Long Beach	United States	North America	2014	

1074 rows × 10 columns

In [2]:

```
#import the necessary libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [12]:
```
unicorn_analysis.tail()
```

Out[12]:

	Company	Valuation	Date Joined	Industry	City	Country	Continent	Year Founded	Fundin
1069	Zhaogang	$1B	2017-06-29	E-commerce & direct-to-consumer	Shanghai	China	Asia	2012	$379N
1070	Zhuan Zhuan	$1B	2017-04-18	E-commerce & direct-to-consumer	Beijing	China	Asia	2015	$990N
1071	Zihaiguo	$1B	2021-05-06	Consumer & retail	Chongqing	China	Asia	2018	$80N
1072	Zopa	$1B	2021-10-19	Fintech	London	United Kingdom	Europe	2005	$792N
1073	Zwift	$1B	2020-09-16	E-commerce & direct-to-consumer	Long Beach	United States	North America	2014	$620N

Data cleaning and manipulation

In [13]:
```
#shape of the data
unicorn_analysis.shape
```

Out[13]:
```
(1074, 10)
```

In [15]:
```
#check the colums
unicorn_analysis.columns
```

Out[15]:
```
Index(['Company', 'Valuation', 'Date Joined', 'Industry', 'City', 'Country',
       'Continent', 'Year Founded', 'Funding', 'Select Investors'],
      dtype='object')
```

In [17]:
```
#check the data types
unicorn_analysis.dtypes
```

Out[17]:
```
Company             object
Valuation           object
Date Joined         object
Industry            object
City                object
Country             object
Continent           object
Year Founded         int64
Funding             object
Select Investors    object
dtype: object
```

In [18]:
```
#info about the data
unicorn_analysis.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1074 entries, 0 to 1073
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Company           1074 non-null   object
 1   Valuation         1074 non-null   object
 2   Date Joined       1074 non-null   object
 3   Industry          1074 non-null   object
 4   City              1058 non-null   object
 5   Country           1074 non-null   object
 6   Continent         1074 non-null   object
 7   Year Founded      1074 non-null   int64 
 8   Funding           1074 non-null   object
 9   Select Investors  1073 non-null   object
dtypes: int64(1), object(9)
memory usage: 84.0+ KB
```

In [19]:
```
#check if there are missing values in the data
unicorn_analysis.isnull()
```

Out[19]:

	Company	Valuation	Date Joined	Industry	City	Country	Continent	Year Founded	Funding	Se Inves
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	
1069	False	False	False	False	False	False	False	False	False	
1070	False	False	False	False	False	False	False	False	False	
1071	False	False	False	False	False	False	False	False	False	
1072	False	False	False	False	False	False	False	False	False	
1073	False	False	False	False	False	False	False	False	False	

1074 rows × 10 columns

In [20]:
```
# double check if there are missing values in the data
unicorn_analysis.isna()
```

Out[20]:

	Company	Valuation	Date Joined	Industry	City	Country	Continent	Year Founded	Funding	Se Inves
0	False	False	False	False	False	False	False	False	False	I
1	False	False	False	False	False	False	False	False	False	I
2	False	False	False	False	False	False	False	False	False	I
3	False	False	False	False	False	False	False	False	False	I
4	False	False	False	False	False	False	False	False	False	I
...	...	...	...	...	...	...	...	...	...	
1069	False	False	False	False	False	False	False	False	False	I
1070	False	False	False	False	False	False	False	False	False	I
1071	False	False	False	False	False	False	False	False	False	I
1072	False	False	False	False	False	False	False	False	False	I
1073	False	False	False	False	False	False	False	False	False	I

1074 rows × 10 columns

In [21]:

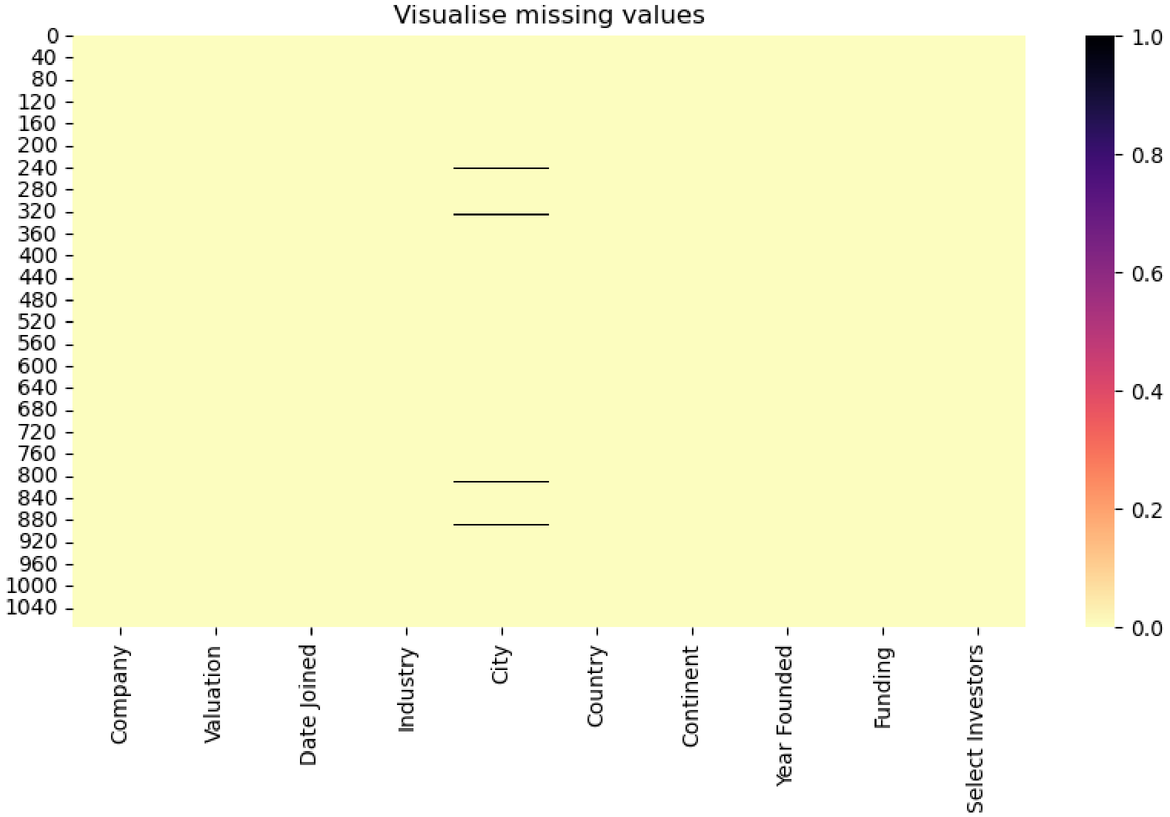
```
# refine check, look through all colums and sum any missing values
unicorn_analysis.isnull().sum()
```

Out[21]:

```
Company              0
Valuation            0
Date Joined          0
Industry             0
City                16
Country              0
Continent            0
Year Founded         0
Funding              0
Select Investors     1
dtype: int64
```

In [27]:

```
#visualise the patterns of missing values
plt.figure(figsize = (10, 5))
plt.title ('Visualise missing values')
sns.heatmap(unicorn_analysis.isnull(),cbar= True, cmap = 'magma_r')
plt.show()
```



In [29]:
```
#Descriptive statistical analysis (numerical variables only)
unicorn_analysis.describe().astype('int')
```

Out[29]:

	Year Founded
count	1074
mean	2012
std	5
min	1919
25%	2011
50%	2014
75%	2016
max	2021

Exploratory Data Analysis of Unicorn Companies

Univariate Analysis

In [38]:
```
# How many cities are industry hubs?
Count_city = unicorn_analysis.value_counts()
```

In [40]:
```
#plotting a bar chart
#Ax=count_city.plot(kind = 'bar', figsize = (10, 5), title = 'Cities and industry h
                   #ylabel = 'Count of industry hubs', legend = False)
#annotate chart
#ax.bar_label(ax.containers[0], label_type = 'edge')
# pad spacing between the number and edge of the figure
```

```
#ax.margins(y = 0.1)
#plt.show()
```

In [41]:
```
#count the number of unique cities and the corresponding industries
#df = unicorn_analysis
#city_industry_counts = df.groupby('City')['Industry'].nunique().reset_index()
#city_industry_counts.columns = ['City', 'Number of Industries']
```

In [42]:
```
city_industry_counts = unicorn_analysis.groupby('City')['Industry'].nunique().rese
city_industry_counts.columns = ['City', 'Industry Hubs']
```

In [44]:
```
city_industry_counts = unicorn_analysis.groupby('City')['Industry'].nunique().rese
city_industry_counts.columns = ['City', 'Number of Industries']
```

In [45]:
```
# Group the data by city and industry and count the unique industries for each city
city_industry_counts = unicorn_analysis.groupby('City')['Industry'].nunique().rese
city_industry_counts.columns = ['City', 'Number of Industries']
```

In [47]:
```
city_industry_groups = unicorn_analysis.groupby('City')['Industry'].unique()
```

In [48]:
```
#check The first five rows
unicorn_analysis.head()
```

Out[48]:

	Company	Valuation	Date Joined	Industry	City	Country	Continent	Year Founded	Funding
0	Bytedance	$180B	2017-04-07	Artificial intelligence	Beijing	China	Asia	2012	$8B
1	SpaceX	$100B	2012-12-01	Other	Hawthorne	United States	North America	2002	$7B
2	SHEIN	$100B	2018-07-03	E-commerce & direct-to-consumer	Shenzhen	China	Asia	2008	$2B
3	Stripe	$95B	2014-01-23	Fintech	San Francisco	United States	North America	2010	$2B
4	Klarna	$46B	2011-12-12	Fintech	Stockholm	Sweden	Europe	2005	$4B

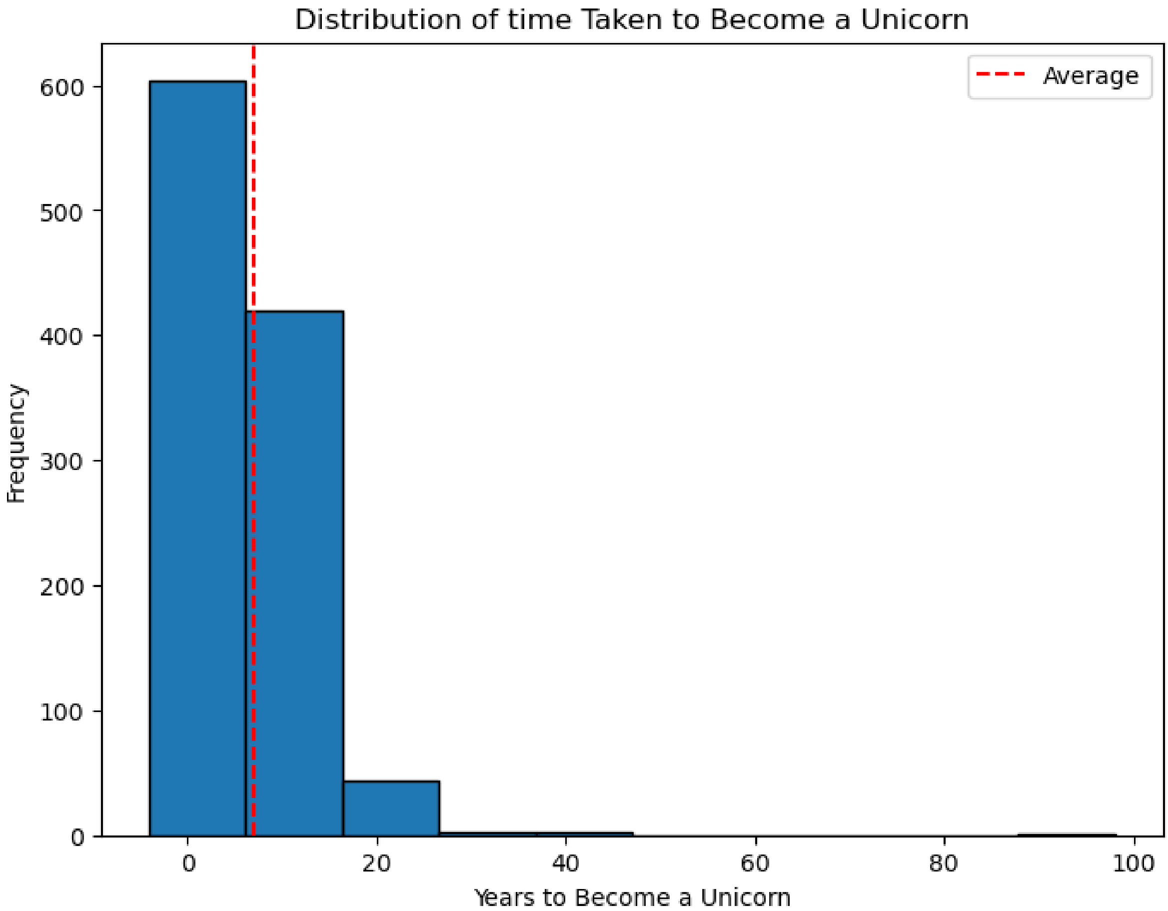
In [55]:
```
city_industry_groups = unicorn_analysis.groupby('City')['Industry'].unique()
industry_hubs = city_industry_groups[city_industry_groups.apply(lambda x: len(x) >
industry_hub_table = pd.DataFrame(industry_hubs).reset_index()
industry_hub_table.columns = ['City', 'Industries']
print("Cities that are industry hubs:")
print(industry_hub_table)
```

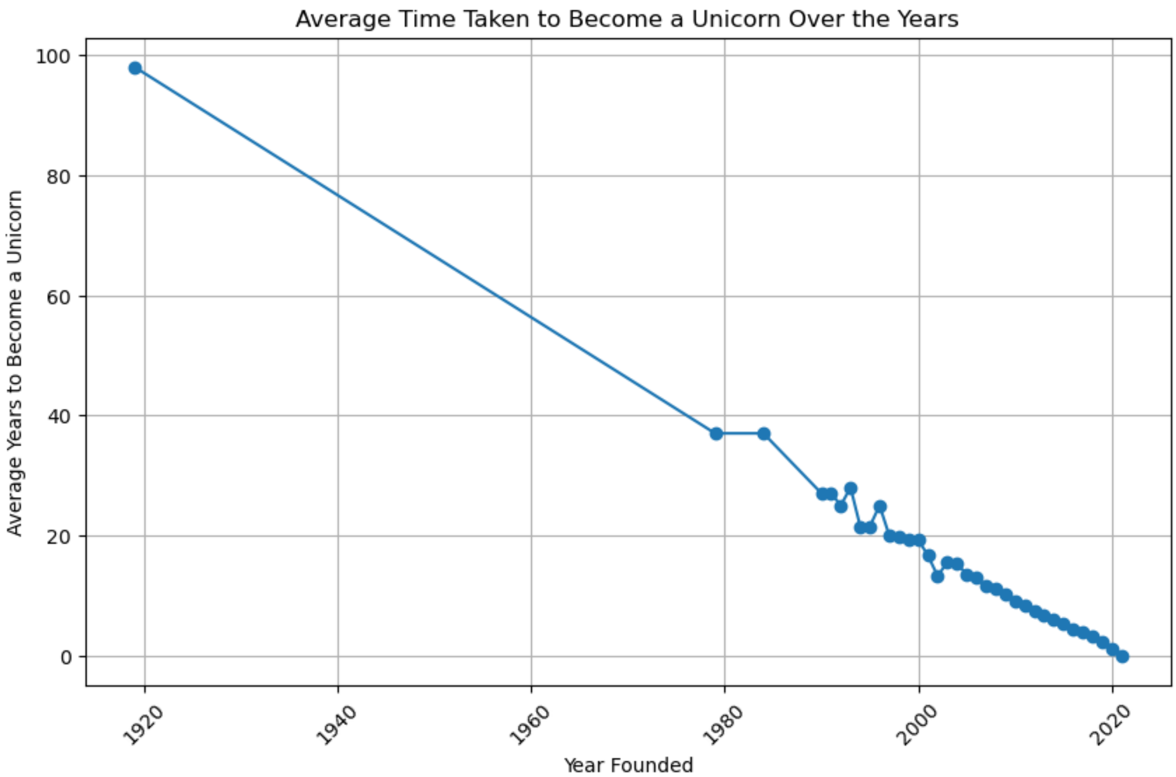
```
Cities that are industry hubs:
             City                                         Industries
0       Amsterdam  [Fintech, Mobile & telecommunications, Hardwar...
1         Atlanta  [Internet software & services, Hardware, Finte...
2          Austin  [Internet software & services, Fintech, E-comm...
3         Bangkok     [Fintech, Supply chain, logistics, & delivery]
4         Beijing  [Artificial intelligence, Edtech, Consumer & r...
..            ...                                                ...
90     Washington              [Other, Internet software & services]
91  Washington DC  [Artificial intelligence, Internet software & ...
92       Waterloo       [Cybersecurity, Internet software & services]
93          Wuhan  [Auto & transportation, E-commerce & direct-to...
94         Zurich     [Fintech, Supply chain, logistics, & delivery]

[95 rows x 2 columns]
```

In [1]:
```
## To check how long it takes a company to be come a unicorn and if the pattern has
```

In [13]:
```
unicorn_analysis['Date Joined'] = pd.to_datetime(unicorn_analysis['Date Joined'], e
unicorn_analysis['Years to Unicorn'] = unicorn_analysis['Date Joined'].dt.year - un
average_years_to_unicorn = unicorn_analysis['Years to Unicorn'].mean()
```

In [17]:
```
# Plot the distribution of years taken to become a unicorn
unicorn_analysis['Date Joined'] = pd.to_datetime(unicorn_analysis['Date Joined'], e
unicorn_analysis['Years to Unicorn'] = unicorn_analysis['Date Joined'].dt.year - un
average_years_to_unicorn = unicorn_analysis['Years to Unicorn'].mean()
plt.figure(figsize=(8, 6))
plt.hist(unicorn_analysis['Years to Unicorn'], bins=10, edgecolor='black')
plt.axvline(average_years_to_unicorn, color='red', linestyle='--', label='Average'
plt.xlabel('Years to Become a Unicorn')
plt.ylabel('Frequency')
plt.title('Distribution of time Taken to Become a Unicorn')
plt.legend()
plt.show()
# Group the data by the year founded and calculate the average time taken to become
average_years_by_year = unicorn_analysis.groupby('Year Founded')['Years to Unicorn

# Plot the average time taken to become a unicorn over the years
plt.figure(figsize=(10, 6))
plt.plot(average_years_by_year['Year Founded'], average_years_by_year['Years to Uni
plt.xlabel('Year Founded')
plt.ylabel('Average Years to Become a Unicorn')
plt.title('Average Time Taken to Become a Unicorn Over the Years')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```





In [ ]:
```
##This code first converts the "Date Joined" column to a datetime format. It then c

## Next, it plots a histogram to show the distribution of the years taken to become

## Finally, the code groups the data by the year founded and calculates the average

##These visualizations and calculations will provide insights into how long it typ
## a unicorn
```

In [ ]:
```
## To count the number of unicorn companies in each country, i use value_counts()
```

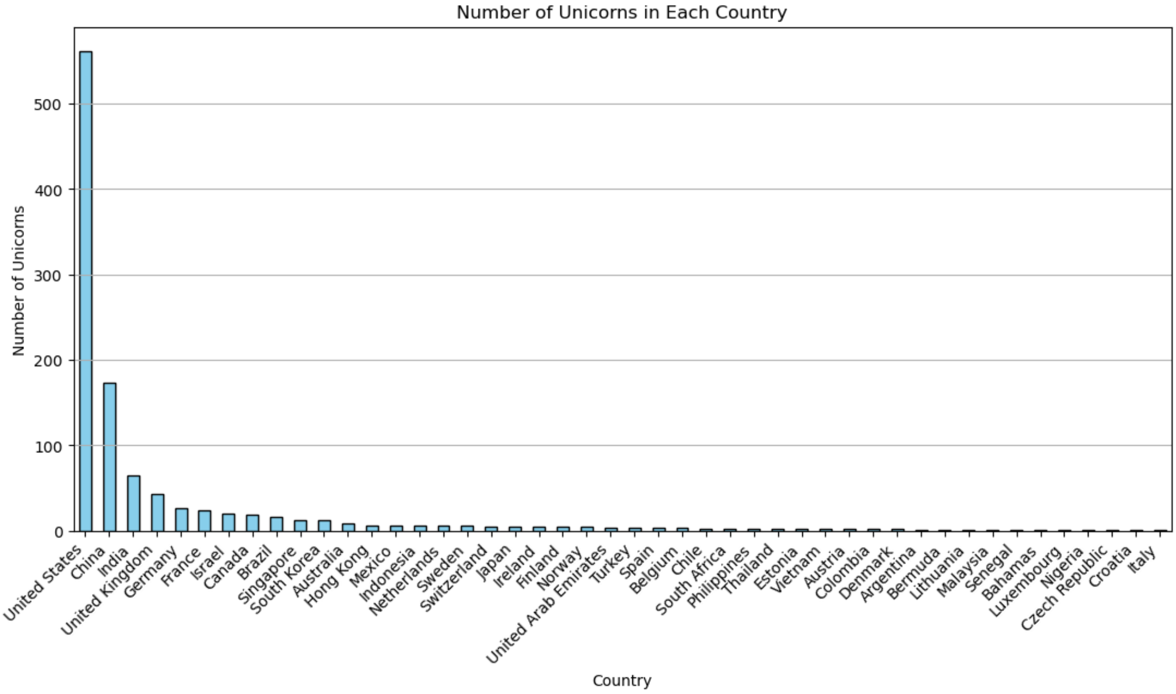
In [11]:
```
unicorn_analysis.head()
```

Out[11]:

	Company	Valuation	Date Joined	Industry	City	Country	Continent	Year Founded	Funding
0	Bytedance	$180B	2017-04-07	Artificial intelligence	Beijing	China	Asia	2012	$8B
1	SpaceX	$100B	2012-12-01	Other	Hawthorne	United States	North America	2002	$7B
2	SHEIN	$100B	2018-07-03	E-commerce & direct-to-consumer	Shenzhen	China	Asia	2008	$2B
3	Stripe	$95B	2014-01-23	Fintech	San Francisco	United States	North America	2010	$2B
4	Klarna	$46B	2011-12-12	Fintech	Stockholm	Sweden	Europe	2005	$4B

In [ ]:

In [16]:
```
unicorns_by_country = unicorn_analysis['Country'].value_counts()
plt.figure(figsize=(13, 6))
unicorns_by_country.plot(kind='bar', color='skyblue', edgecolor='black')
plt.xlabel('Country')
plt.ylabel('Number of Unicorns')
plt.title('Number of Unicorns in Each Country')
plt.xticks(rotation=45, ha='right', fontsize=10)
plt.grid(axis='y')
plt.show()
```



In [ ]:

```
#Cities that are industry hubs
```

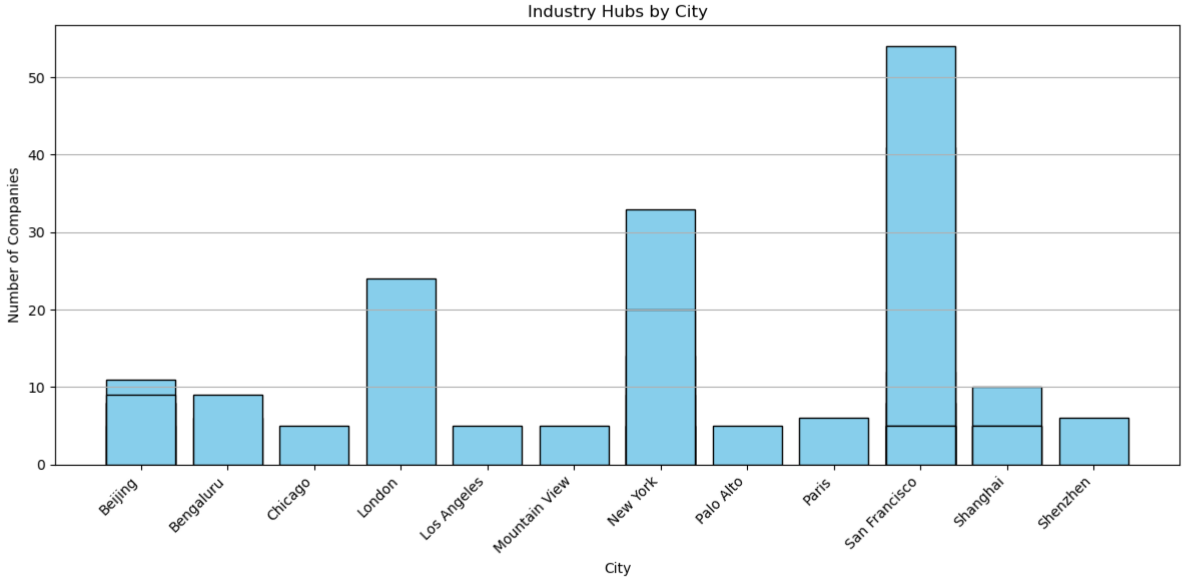
In [17]:

```
industry_hubs = unicorn_analysis.groupby(['City', 'Industry']).size().reset_index(
threshold = 5
industry_hubs = industry_hubs[industry_hubs['Count'] >= threshold]
print(industry_hubs[['City', 'Industry', 'Count']])
```

```
                City                             Industry  Count
28           Beijing              Artificial intelligence      8
31           Beijing        E-commerce & direct-to-consumer     11
32           Beijing                               Edtech      8
36           Beijing          Internet software & services      5
37           Beijing         Mobile & telecommunications      9
48         Bengaluru                              Fintech      6
50         Bengaluru          Internet software & services      9
122          Chicago          Internet software & services      5
252           London                              Fintech     24
264      Los Angeles                              Fintech      5
296    Mountain View              Artificial intelligence      5
321         New York              Artificial intelligence      5
324         New York                        Cybersecurity      9
326         New York       E-commerce & direct-to-consumer      5
328         New York                              Fintech     33
330         New York                               Health     14
331         New York          Internet software & services     20
353        Palo Alto          Internet software & services      5
358            Paris       E-commerce & direct-to-consumer      6
404    San Francisco              Artificial intelligence      8
405    San Francisco                    Consumer & retail      5
406    San Francisco                        Cybersecurity      8
408    San Francisco       E-commerce & direct-to-consumer      5
410    San Francisco                              Fintech     41
412    San Francisco                               Health     12
413    San Francisco          Internet software & services     54
414    San Francisco         Mobile & telecommunications      5
416    San Francisco  Supply chain, logistics, & delivery      5
463         Shanghai            Auto & transportation     10
465         Shanghai       E-commerce & direct-to-consumer      5
472         Shanghai  Supply chain, logistics, & delivery      5
476         Shenzhen                             Hardware      6
```

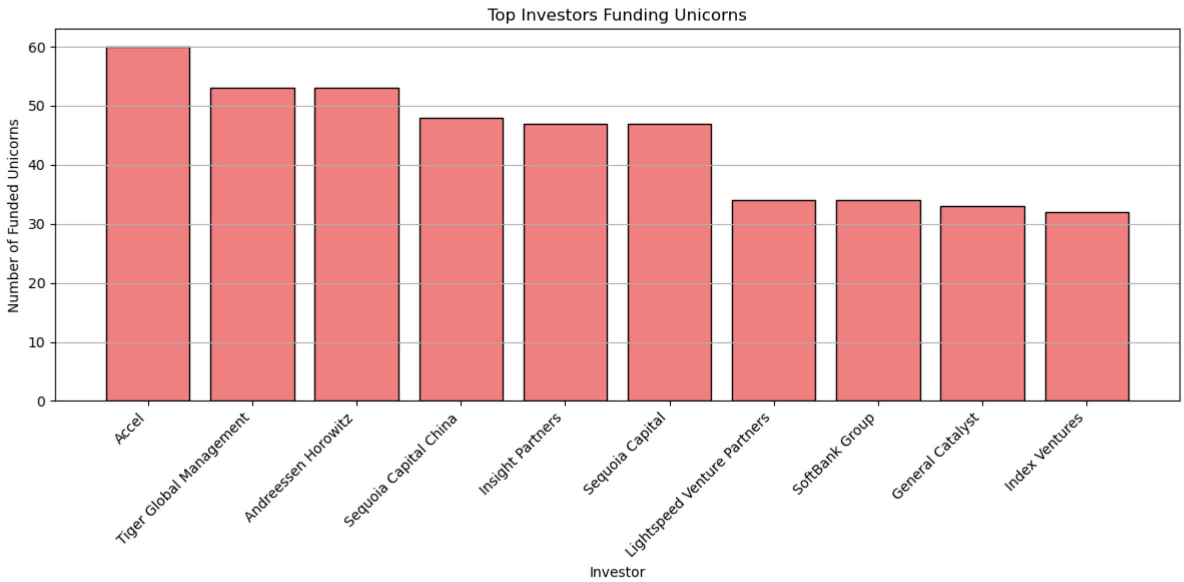
In [18]:
```python
industry_hubs = unicorn_analysis.groupby(['City', 'Industry']).size().reset_index(
threshold = 5
industry_hubs = industry_hubs[industry_hubs['Count'] >= threshold]
print(industry_hubs[['City', 'Industry', 'Count']])
plt.figure(figsize=(12, 6))  # Adjust the figure size for better visualization
plt.bar(industry_hubs['City'], industry_hubs['Count'], color='skyblue', edgecolor=
plt.xlabel('City')
plt.ylabel('Number of Companies')
plt.title('Industry Hubs by City')
plt.xticks(rotation=45, ha='right', fontsize=10)
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```

```
              City                            Industry  Count
28         Beijing             Artificial intelligence      8
31         Beijing      E-commerce & direct-to-consumer     11
32         Beijing                              Edtech      8
36         Beijing        Internet software & services      5
37         Beijing         Mobile & telecommunications      9
48       Bengaluru                             Fintech      6
50       Bengaluru        Internet software & services      9
122        Chicago        Internet software & services      5
252         London                             Fintech     24
264    Los Angeles                             Fintech      5
296  Mountain View             Artificial intelligence      5
321       New York             Artificial intelligence      5
324       New York                       Cybersecurity      9
326       New York      E-commerce & direct-to-consumer      5
328       New York                             Fintech     33
330       New York                              Health     14
331       New York        Internet software & services     20
353      Palo Alto        Internet software & services      5
358          Paris      E-commerce & direct-to-consumer      6
404  San Francisco             Artificial intelligence      8
405  San Francisco                   Consumer & retail      5
406  San Francisco                       Cybersecurity      8
408  San Francisco      E-commerce & direct-to-consumer      5
410  San Francisco                             Fintech     41
412  San Francisco                              Health     12
413  San Francisco        Internet software & services     54
414  San Francisco         Mobile & telecommunications      5
416  San Francisco  Supply chain, logistics, & delivery      5
463       Shanghai              Auto & transportation     10
465       Shanghai      E-commerce & direct-to-consumer      5
472       Shanghai  Supply chain, logistics, & delivery      5
476       Shenzhen                            Hardware      6
```



In [23]:
```
unicorn_analysis.dropna(subset=['Select Investors'], inplace=True)
unicorn_analysis['Select Investors'] = unicorn_analysis['Select Investors'].apply(:
investor_lists = unicorn_analysis['Select Investors'].explode().reset_index(drop=Tr
investor_counts = investor_lists.value_counts().reset_index()
investor_counts.columns = ['Investor', 'Number of Unicorns']
print(investor_counts.head())
plt.figure(figsize=(12, 6))
plt.bar(investor_counts['Investor'][:10], investor_counts['Number of Unicorns'][:10
plt.xlabel('Investor')
plt.ylabel('Number of Funded Unicorns')
plt.title('Top Investors Funding Unicorns')
plt.xticks(rotation=45, ha='right', fontsize=10)
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```

```
                 Investor  Number of Unicorns
0                   Accel                  60
1  Tiger Global Management                  53
2      Andreessen Horowitz                  53
3      Sequoia Capital China                  48
4         Insight Partners                  47
```



In [ ]:
```
# Drop rows with missing "Select Investors" data
unicorn_analysis.dropna(subset=['Select Investors'], inplace=True)
```

```
# Explode the lists to create a flat list of all investors
investor_lists = unicorn_analysis['Select Investors'].explode().reset_index(drop=Tr

# Count the occurrences of each investor
investor_counts = investor_lists.value_counts().reset_index()
investor_counts.columns = ['Investor', 'Number of Unicorns']

# Display the investors with the most funded unicorns
print(investor_counts.head())

# Plot the result as a pie chart
plt.figure(figsize=(8, 8))
plt.pie(investor_counts['Number of Unicorns'][:10], labels=investor_counts['Investo
plt.axis('equal')
plt.title('Top Investors Funding Unicorns')
plt.tight_layout()
plt.show()
```

In [27]:
```
# Descriptive statistics for numerical columns
numerical_stats = unicorn_analysis.describe()
print(numerical_stats)
```

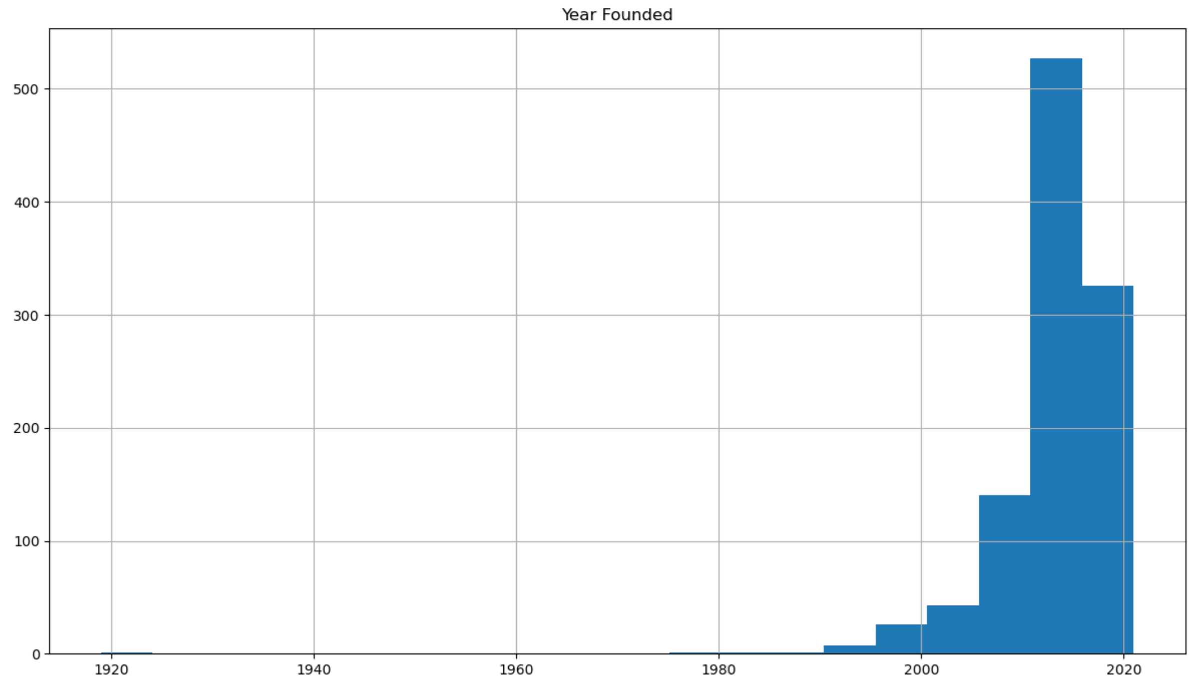
```
       Year Founded
count    1073.00000
mean     2012.89562
std         5.70123
min      1919.00000
25%      2011.00000
50%      2014.00000
75%      2016.00000
max      2021.00000
```

Observation

- The oldest company was founded in 1919

In [28]:
```
# Histograms for numerical columns
unicorn_analysis.hist(figsize=(12, 8), bins=20)
plt.suptitle("Univariate Analysis: Histograms", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```

Univariate Analysis: Histograms



In [29]:
```
# Value counts for categorical columns
categorical_columns = ['Industry', 'City', 'Country', 'Continent']
for column in categorical_columns:
    value_counts = unicorn_analysis[column].value_counts()
    print(f"\nValue counts for {column}:")
    print(value_counts)
```

```
Value counts for Industry:
Fintech                                224
Internet software & services           205
E-commerce & direct-to-consumer        111
Health                                  74
Artificial intelligence                 73
Other                                   58
Supply chain, logistics, & delivery     57
Cybersecurity                           50
Data management & analytics             41
Mobile & telecommunications             37
Hardware                                34
Auto & transportation                   31
Edtech                                  28
Consumer & retail                       25
Travel                                  14
Artificial Intelligence                 11
Name: Industry, dtype: int64

Value counts for City:
San Francisco    152
New York         103
Beijing           63
Shanghai          43
London            34
                ...
Santa Barbara      1
Altrincham         1
Northbrook         1
Cincinnati         1
Milpitas           1
Name: City, Length: 256, dtype: int64

Value counts for Country:
United States           562
China                   172
India                    65
United Kingdom           43
Germany                  26
France                   24
Israel                   20
Canada                   19
Brazil                   16
Singapore                12
South Korea              12
Australia                 8
Hong Kong                 6
Mexico                    6
Indonesia                 6
Netherlands               6
Sweden                    6
Switzerland               5
Japan                     5
Ireland                   5
Finland                   4
Norway                    4
United Arab Emirates      3
Turkey                    3
Spain                     3
Belgium                   3
Chile                     2
South Africa              2
Philippines               2
Thailand                  2
```

```
Estonia                 2
Vietnam                 2
Austria                 2
Colombia                2
Denmark                 2
Argentina               1
Bermuda                 1
Lithuania               1
Malaysia                1
Senegal                 1
Bahamas                 1
Luxembourg              1
Nigeria                 1
Czech Republic          1
Croatia                 1
Italy                   1
Name: Country, dtype: int64

Value counts for Continent:
North America    589
Asia             309
Europe           143
South America     21
Oceania            8
Africa             3
Name: Continent, dtype: int64
```

Observation

- United States has the largest Unicorns while 20% of comapnies listed are in the Fintech industry

In [6]:
```
unicorn_analysis = pd.read_csv(r'C:\Users\adora\OneDrive\Desktop\Quantum Analytics\
unicorn_analysis
```

Out[6]:

	Company	Valuation	Date Joined	Industry	City	Country	Continent	Year Founded	Fundir
0	Bytedance	$180B	2017-04-07	Artificial intelligence	Beijing	China	Asia	2012	$8
1	SpaceX	$100B	2012-12-01	Other	Hawthorne	United States	North America	2002	$7
2	SHEIN	$100B	2018-07-03	E-commerce & direct-to-consumer	Shenzhen	China	Asia	2008	$2
3	Stripe	$95B	2014-01-23	Fintech	San Francisco	United States	North America	2010	$2
4	Klarna	$46B	2011-12-12	Fintech	Stockholm	Sweden	Europe	2005	$4
...	...	...	...	...	...	...	...	...	
1069	Zhaogang	$1B	2017-06-29	E-commerce & direct-to-consumer	Shanghai	China	Asia	2012	$379
1070	Zhuan Zhuan	$1B	2017-04-18	E-commerce & direct-to-consumer	Beijing	China	Asia	2015	$990
1071	Zihaiguo	$1B	2021-05-06	Consumer & retail	Chongqing	China	Asia	2018	$80
1072	Zopa	$1B	2021-10-19	Fintech	London	United Kingdom	Europe	2005	$792
1073	Zwift	$1B	2020-09-16	E-commerce & direct-to-consumer	Long Beach	United States	North America	2014	$620

1074 rows × 10 columns

Observation

- United States has the largest Unicorns

In [11]:

```
# Insight 1: Total number of unicorn companies in the dataset
total_unicorns = len(unicorn_analysis)
print(f"Insight 1: Total number of unicorn companies: {total_unicorns}")
```

```
Insight 1: Total number of unicorn companies: 1074
```

In [13]:

```
# Insight 2: Countries with the most unicorn companies
most_unicorns_countries = unicorn_analysis["Country"].value_counts().head(3)
print("Insight 3: Countries with the most unicorn companies:")
print(most_unicorns_countries)
```

```
Insight 3: Countries with the most unicorn companies:
United States    562
China            173
India             65
Name: Country, dtype: int64
```

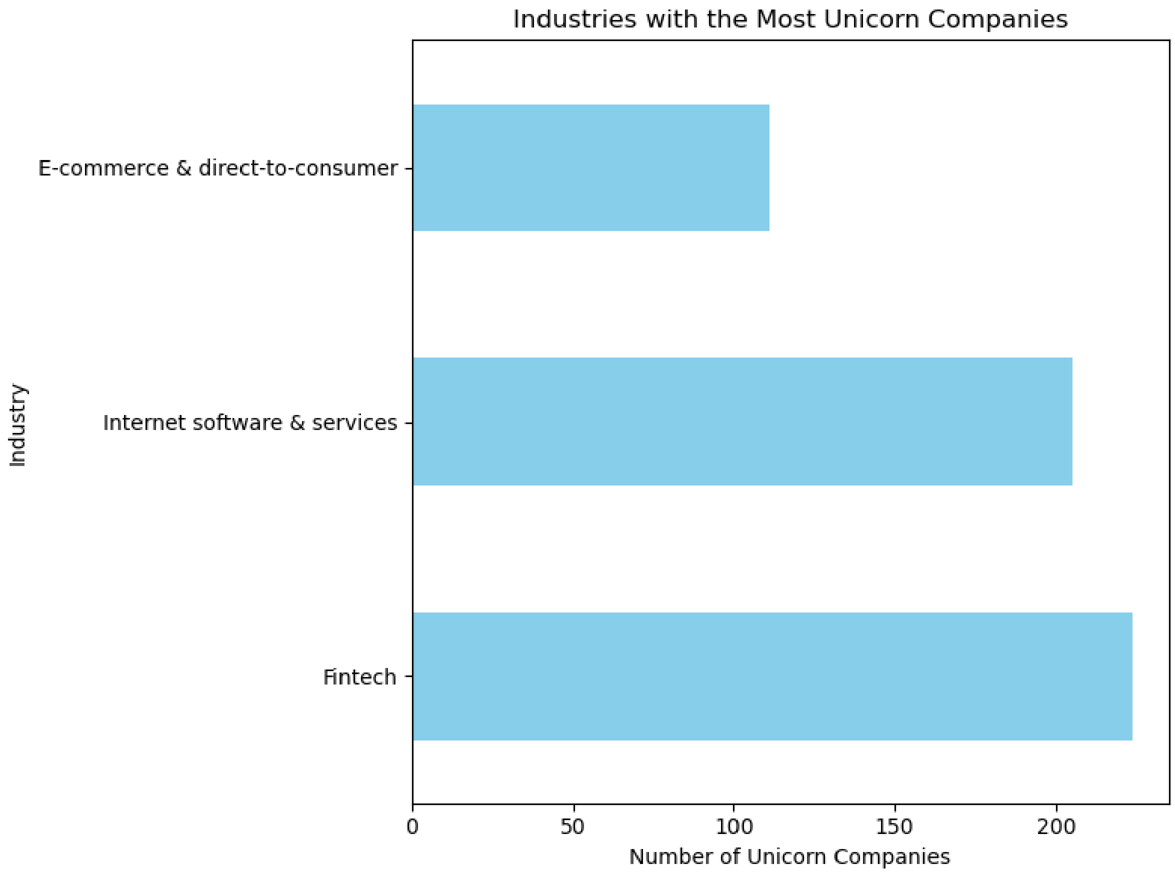
In [16]:

```
# Insight 3: Industries with the most unicorn companies
most_unicorns_industries = unicorn_analysis["Industry"].value_counts().head(3)
print("Insight 3: Industries with the most unicorn companies:")
print(most_unicorns_industries)
```

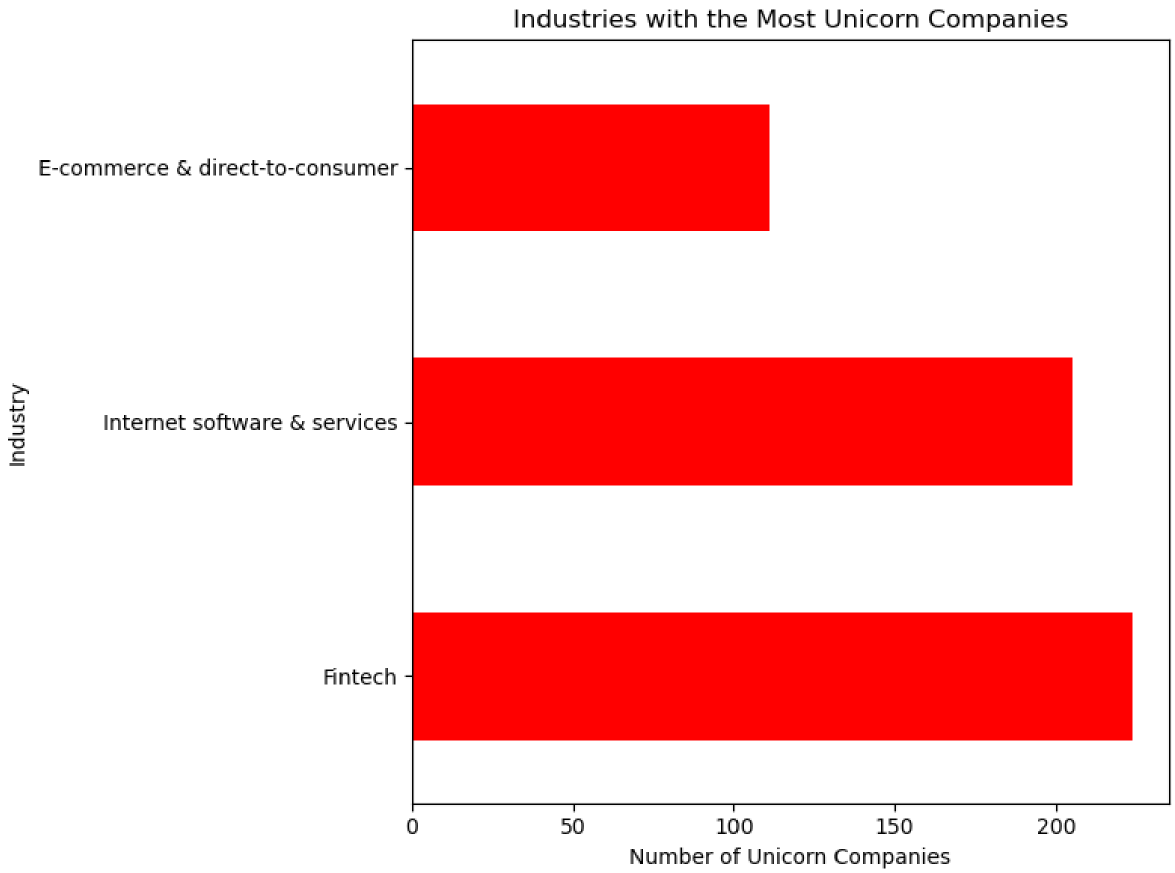
```
Insight 3: Industries with the most unicorn companies:
Fintech                               224
Internet software & services          205
E-commerce & direct-to-consumer       111
Name: Industry, dtype: int64
```

In [19]:

```
# Create a horizontal bar chart
plt.figure(figsize=(8, 6))
most_unicorns_industries.plot(kind='barh', color='skyblue')
plt.title("Industries with the Most Unicorn Companies")
plt.xlabel("Number of Unicorn Companies")
plt.ylabel("Industry")
plt.tight_layout()

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



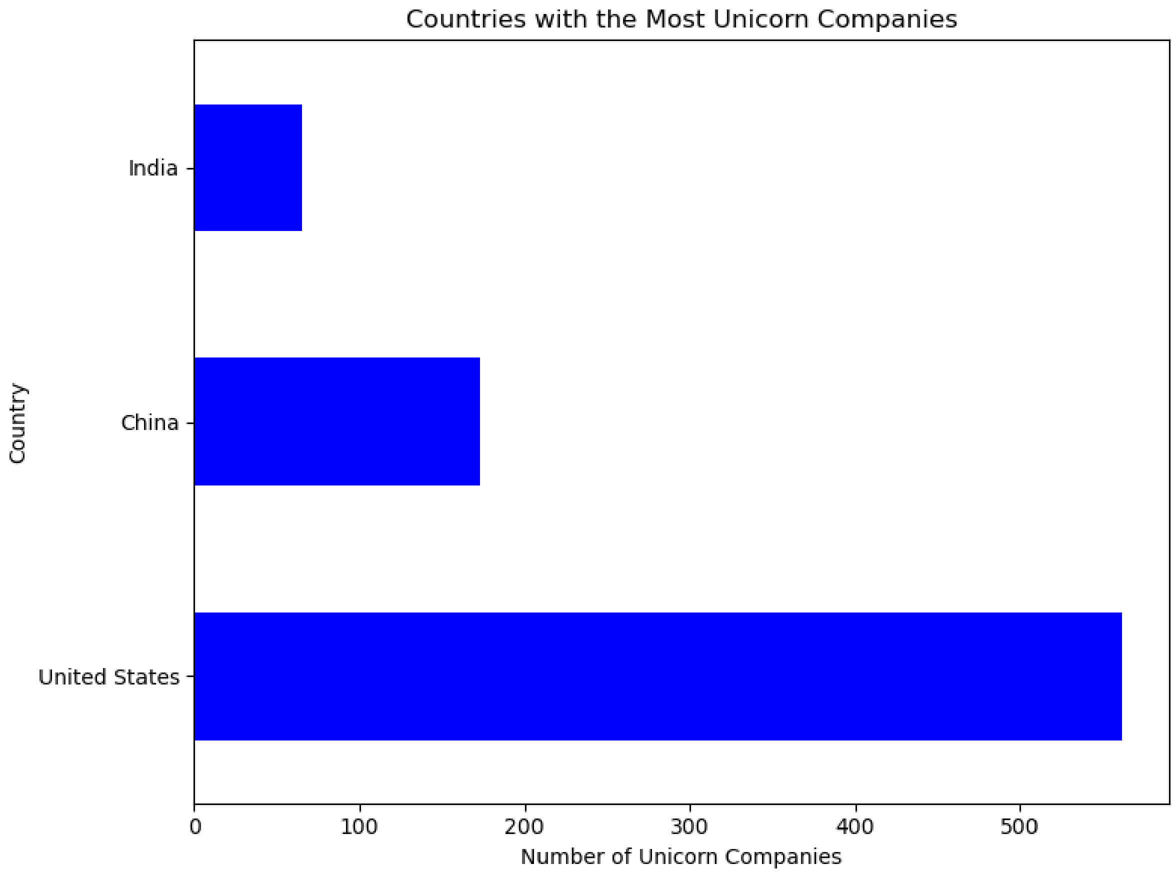
In [21]:
```
# Create a horizontal bar chart
plt.figure(figsize=(8, 6))
most_unicorns_industries.plot(kind='barh', color='red')
plt.title("Industries with the Most Unicorn Companies")
plt.xlabel("Number of Unicorn Companies")
plt.ylabel("Industry")
plt.tight_layout()

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



In [22]:
```
# Insight 5: Year with the highest number of unicorn company formations
year_with_most_unicorns = unicorn_analysis["Year Founded"].value_counts().idxmax()
print(f"Insight 5: Year with the highest number of unicorn company formations: {ye
```

```
Insight 5: Year with the highest number of unicorn company formations: 2015
```

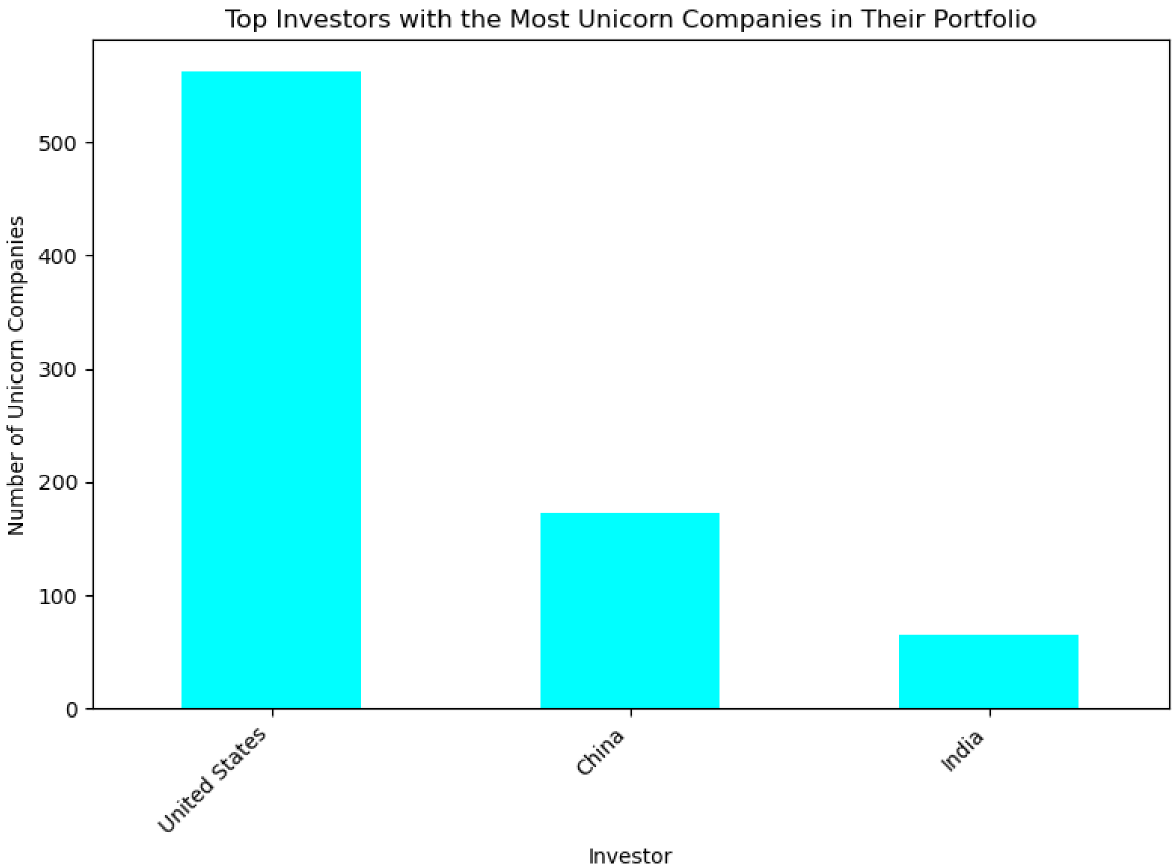
In [24]:
```
# Define the color for the bar chart (dark blue)
bar_color = "blue"

# Create a horizontal bar chart
plt.figure(figsize=(8, 6))
most_unicorns_countries.plot(kind='barh', color=bar_color)
plt.title("Countries with the Most Unicorn Companies")
plt.xlabel("Number of Unicorn Companies")
plt.ylabel("Country")
plt.tight_layout()

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



In [25]:
```
# Insight 6: Who are the top investors with the most unicorn companies in their po
top_investors = unicorn_analysis["Select Investors"].str.split(", ").explode().val
print("Insight 6: Top investors with the most unicorn companies in their portfolio
print(top_investors)
```

```
Insight 6: Top investors with the most unicorn companies in their portfolio:
Accel                      60
Tiger Global Management    53
Andreessen Horowitz        53
Name: Select Investors, dtype: int64
```

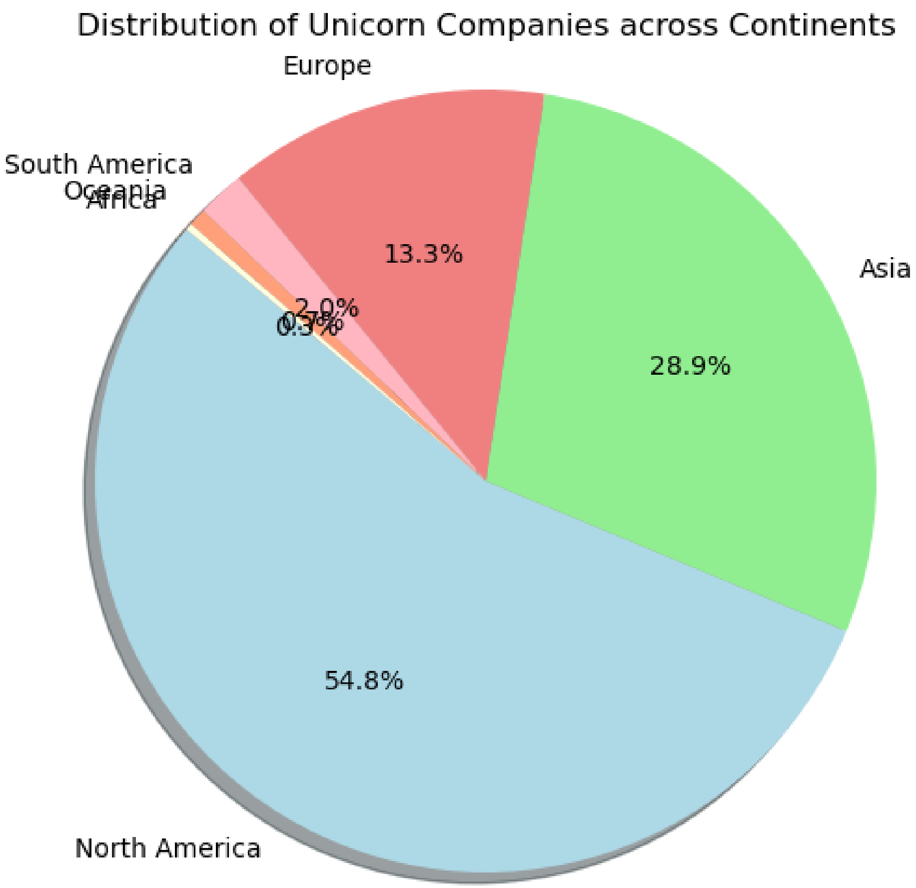
In [27]:
```
# Create a horizontal bar chart
plt.figure(figsize=(8, 6))
most_unicorns_countries.plot(kind='bar', color='cyan')
plt.title("Top Investors with the Most Unicorn Companies in Their Portfolio")
plt.xlabel("Investor")
plt.ylabel("Number of Unicorn Companies")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

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



In [30]:
```
# Insight 9: Distribution of unicorn companies across continents
continent_distribution = unicorn_analysis["Continent"].value_counts()
print("Insight 9: Distribution of unicorn companies across continents:")
print(continent_distribution)
```

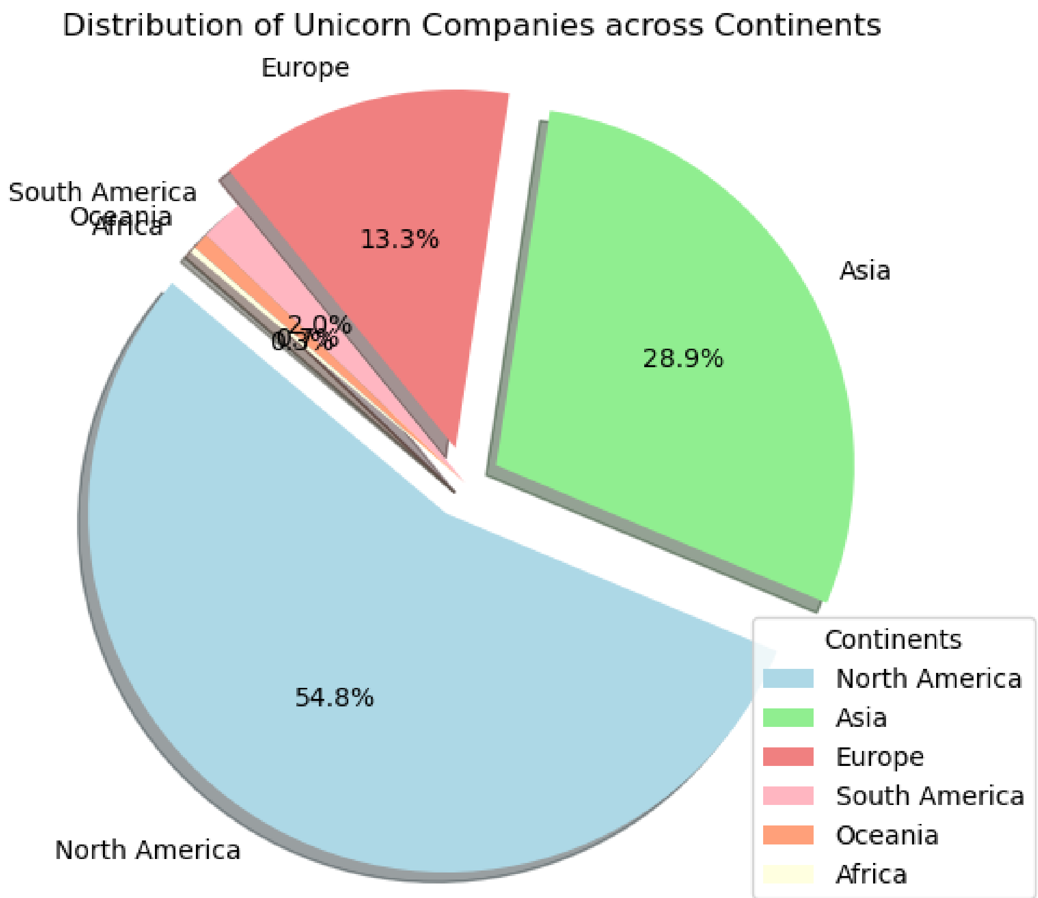
```
Insight 9: Distribution of unicorn companies across continents:
North America    589
Asia             310
Europe           143
South America     21
Oceania            8
Africa             3
Name: Continent, dtype: int64
```

In [35]:
```
# Create a pie chart
plt.figure(figsize=(8, 6))
colors = ['lightblue', 'lightgreen', 'lightcoral', 'lightpink', 'lightsalmon', 'li
plt.pie(continent_distribution, labels=continent_distribution.index, colors=colors
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.title("Distribution of Unicorn Companies across Continents")

# Set the aspect ratio to ensure the pie chart is drawn properly
plt.subplots_adjust(wspace=0, hspace=0)

plt.show()
```



In [36]:

```
# Create a pie chart
plt.figure(figsize=(8, 6))
colors = ['lightblue', 'lightgreen', 'lightcoral', 'lightpink', 'lightsalmon', 'li

# Explode the slices to create distance from the center for South America, Oceania,
explode = (0.1, 0.1, 0.1, 0, 0, 0)

plt.pie(continent_distribution, labels=continent_distribution.index, colors=colors
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.title("Distribution of Unicorn Companies across Continents")

# Add a legend
plt.legend(continent_distribution.index, title="Continents", loc="best")

plt.show()
```



In [31]:
```
# Insight 10: Oldest and newest unicorn companies
oldest_unicorn = unicorn_analysis.nsmallest(1, "Year Founded")[["Company", "Year Fo
newest_unicorn = unicorn_analysis.nlargest(1, "Year Founded")[["Company", "Year Fou
print("Insight 10: Oldest and newest unicorn companies:")
print(oldest_unicorn)
print(newest_unicorn)
```

```
Insight 10: Oldest and newest unicorn companies:
                    Company  Year Founded
189  Otto Bock HealthCare          1919
        Company  Year Founded
238  Yuga Labs          2021
```

In [9]:
```
unicorn_analysis = pd.read_csv
unicorn_analysis = pd.read_csv(r'C:\Users\adora\OneDrive\Desktop\Quantum Analytics\
unicorn_analysis
```

Out[9]:

	Company	Valuation	Date Joined	Industry	City	Country	Continent	Year Founded	Fu
0	Bytedance	$180B	07/04/2017	Artificial intelligence	Beijing	China	Asia	2012	
1	SpaceX	$100B	01/12/2012	Other	Hawthorne	United States	North America	2002	
2	SHEIN	$100B	03/07/2018	E-commerce & direct-to-consumer	Shenzhen	China	Asia	2008	
3	Stripe	$95B	23/01/2014	Fintech	San Francisco	United States	North America	2010	
4	Klarna	$46B	12/12/2011	Fintech	Stockholm	Sweden	Europe	2005	
...	...	...	...	...	...	...	...	...	
1069	Zhaogang	$1B	29/06/2017	E-commerce & direct-to-consumer	Shanghai	China	Asia	2012	
1070	Zhuan Zhuan	$1B	18/04/2017	E-commerce & direct-to-consumer	Beijing	China	Asia	2015	
1071	Zihaiguo	$1B	06/05/2021	Consumer & retail	Chongqing	China	Asia	2018	
1072	Zopa	$1B	19/10/2021	Fintech	London	United Kingdom	Europe	2005	
1073	Zwift	$1B	16/09/2020	E-commerce & direct-to-consumer	Long Beach	United States	North America	2014	

1074 rows × 10 columns

In [2]:

```
unicorn_analysis = pd.read_csv
unicorn_analysis = pd.read_csv(r'C:\Users\adora\OneDrive\Desktop\Quantum Analytics'
unicorn_analysis
```

Out[2]:

	Company	Valuation	Date Joined	Industry	City	Country	Continent	Year Founded	Fu
0	Bytedance	$180B	07/04/2017	Artificial intelligence	Beijing	China	Asia	2012	
1	SpaceX	$100B	01/12/2012	Other	Hawthorne	United States	North America	2002	
2	SHEIN	$100B	03/07/2018	E-commerce & direct-to-consumer	Shenzhen	China	Asia	2008	
3	Stripe	$95B	23/01/2014	Fintech	San Francisco	United States	North America	2010	
4	Klarna	$46B	12/12/2011	Fintech	Stockholm	Sweden	Europe	2005	
...	...	...	...	...	...	...	...	...	
1069	Zhaogang	$1B	29/06/2017	E-commerce & direct-to-consumer	Shanghai	China	Asia	2012	
1070	Zhuan Zhuan	$1B	18/04/2017	E-commerce & direct-to-consumer	Beijing	China	Asia	2015	
1071	Zihaiguo	$1B	06/05/2021	Consumer & retail	Chongqing	China	Asia	2018	
1072	Zopa	$1B	19/10/2021	Fintech	London	United Kingdom	Europe	2005	
1073	Zwift	$1B	16/09/2020	E-commerce & direct-to-consumer	Long Beach	United States	North America	2014	

1074 rows × 10 columns

In [ ]:

```
##insights - to calculate the time it takes a company to become a unicorn
## First calculate the the time it took each company to achieve a valuation of $1 b
## Then calculate the average time across all companies and check if there are any
```

In [3]:

```
import pandas as pd
from datetime import datetime
```

```
unicorn_analysis
```

Out[3]:

	Company	Valuation	Date Joined	Industry	City	Country	Continent	Year Founded	Fu
0	Bytedance	$180B	07/04/2017	Artificial intelligence	Beijing	China	Asia	2012	
1	SpaceX	$100B	01/12/2012	Other	Hawthorne	United States	North America	2002	
2	SHEIN	$100B	03/07/2018	E-commerce & direct-to-consumer	Shenzhen	China	Asia	2008	
3	Stripe	$95B	23/01/2014	Fintech	San Francisco	United States	North America	2010	
4	Klarna	$46B	12/12/2011	Fintech	Stockholm	Sweden	Europe	2005	
...	...	...	...	...	...	...	...	...	
1069	Zhaogang	$1B	29/06/2017	E-commerce & direct-to-consumer	Shanghai	China	Asia	2012	
1070	Zhuan Zhuan	$1B	18/04/2017	E-commerce & direct-to-consumer	Beijing	China	Asia	2015	
1071	Zihaiguo	$1B	06/05/2021	Consumer & retail	Chongqing	China	Asia	2018	
1072	Zopa	$1B	19/10/2021	Fintech	London	United Kingdom	Europe	2005	
1073	Zwift	$1B	16/09/2020	E-commerce & direct-to-consumer	Long Beach	United States	North America	2014	

1074 rows × 10 columns

In [14]:

```python
import pandas as pd
from datetime import datetime
# Convert valuation to numeric values (remove $ sign and 'B')
unicorn_analysis["Valuation"] = unicorn_analysis["Valuation"].replace({"\$": "", "
```

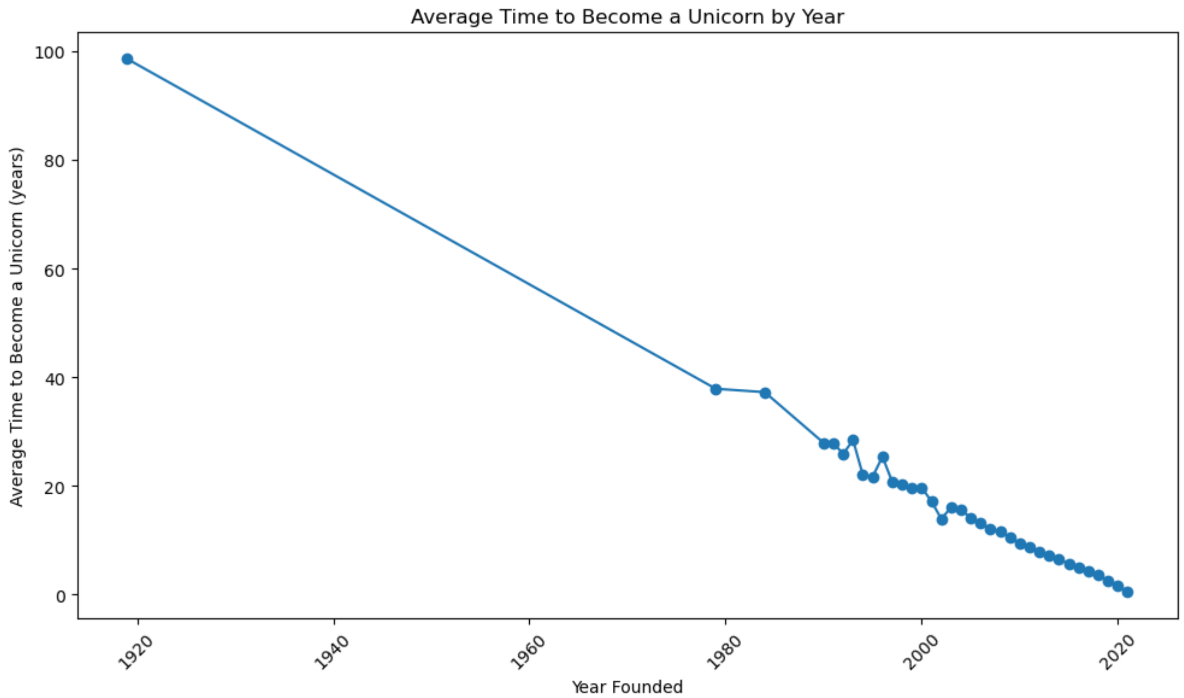
```
# Convert 'Date Joined' to datetime format
unicorn_analysis["Date Joined"] = pd.to_datetime(unicorn_analysis["Date Joined"], 

# Calculate the time taken to become a unicorn in years
unicorn_analysis["Time to Unicorn"] = (unicorn_analysis["Date Joined"] - pd.to_dat

# Calculate the average time taken to become a unicorn
average_time_to_unicorn = unicorn_analysis["Time to Unicorn"].mean()

print("Average Time to Become a Unicorn (in years):", average_time_to_unicorn)

# Group the data by 'Year Founded' and calculate the average time to become a unic
average_time_by_year = unicorn_analysis.groupby("Year Founded")["Time to Unicorn"]

print("\nAverage Time to Become a Unicorn by Year:")
print(average_time_by_year)
```

```
Average Time to Become a Unicorn (in years): 7.490798704114691

Average Time to Become a Unicorn by Year:
Year Founded
1919    98.545205
1979    37.879452
1984    37.257534
1990    27.923288
1991    27.835616
1992    25.915068
1993    28.435616
1994    22.010959
1995    21.719178
1996    25.378082
1997    20.734247
1998    20.401096
1999    19.702055
2000    19.701619
2001    17.157686
2002    13.927397
2003    16.076370
2004    15.620890
2005    14.145597
2006    13.248584
2007    12.081507
2008    11.703095
2009    10.586865
2010     9.481438
2011     8.859305
2012     7.933324
2013     7.174429
2014     6.512379
2015     5.675669
2016     4.964309
2017     4.316735
2018     3.595284
2019     2.639269
2020     1.658521
2021     0.622914
Name: Time to Unicorn, dtype: float64
```

In [17]:

```
## To understand the trend over time , plot an average time graph
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime

# Convert valuation to numeric values (remove $ sign and 'B')
```

```
unicorn_analysis["Valuation"] = unicorn_analysis["Valuation"].replace({"\$": "", "

# Convert 'Date Joined' to datetime format
unicorn_analysis["Date Joined"] = pd.to_datetime(unicorn_analysis["Date Joined"], 

# Calculate the time taken to become a unicorn in years
unicorn_analysis["Time to Unicorn"] = (unicorn_analysis["Date Joined"] - pd.to_dat

# Group the data by 'Year Founded' and calculate the average time to become a unic
average_time_by_year = unicorn_analysis.groupby("Year Founded")["Time to Unicorn"]

# Plot the average time to become a unicorn by year
plt.figure(figsize=(10, 6))
plt.plot(average_time_by_year.index, average_time_by_year.values, marker='o', line
plt.xlabel("Year Founded")
plt.ylabel("Average Time to Become a Unicorn (years)")
plt.title("Average Time to Become a Unicorn by Year")
plt.grid(False)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



In [ ]:
```
## It takes an average of 7.5 years to become a Unicorn  company
```

In [ ]:
```
## the chart shows that average time was on a steady decline fro, the 1920s until 
```

In [ ]:
```
##Calculating the ROI for Unicorn companies
import pandas as pd
```

In [21]:
```
# Define a function to convert the funding values to numeric values
def funding_to_numeric(funding_str):
    if funding_str == 'Unknown':
        return None
    multiplier = 1
    if funding_str[-1] == 'B':
        multiplier = 1000
    funding_str = funding_str.replace('$', '').replace('B', '').replace('M', '')
    return float(funding_str) * multiplier

# Convert valuation to numeric values (remove $ sign and 'B')
```

```
unicorn_analysis["Valuation"] = unicorn_analysis["Valuation"].replace({"\$": "", "I

# Convert funding to numeric values
unicorn_analysis["Funding"] = unicorn_analysis["Funding"].apply(funding_to_numeric

# Calculate ROI for each company (ignoring rows with 'Unknown' funding or valuatio
unicorn_analysis["ROI"] = ((unicorn_analysis["Valuation"] - unicorn_analysis["Fund

print(unicorn_analysis[["Company", "Valuation", "Funding", "ROI"]])
```

```
           Company  Valuation  Funding        ROI
0        Bytedance      180.0   8000.0 -97.750000
1           SpaceX      100.0   7000.0 -98.571429
2            SHEIN      100.0   2000.0 -95.000000
3           Stripe       95.0   2000.0 -95.250000
4           Klarna       46.0   4000.0 -98.850000
...            ...        ...      ...        ...
1069      Zhaogang        1.0    379.0 -99.736148
1070  Zhuan Zhuan         1.0    990.0 -99.898990
1071      Zihaiguo        1.0     80.0 -98.750000
1072          Zopa        1.0    792.0 -99.873737
1073         Zwift        1.0    620.0 -99.838710

[1074 rows x 4 columns]
```

In [5]:
```
unicorn_analysis['Funding'] = pd.to_numeric(unicorn_analysis['Funding'], errors='c
# Insight: Companies with the highest funding
highest_funded_companies = unicorn_analysis.nlargest(3, "Funding")[["Company", "Fu
print("Insight: Companies with the highest funding:")
print(highest_funded_companies)
```

```
Insight: Companies with the highest funding:
     Company  Funding
0  Bytedance      NaN
1     SpaceX      NaN
2      SHEIN      NaN
```

In [ ]:
```
##Bytedance, Space X and Shein have the largest funding
```

In [7]:
```
## To calculate the ROI
## ROI = (Current Valuation - Initial Funding) / Initial Funding * 100
# Convert valuation and funding to numeric values (remove $ sign and 'B')
unicorn_analysis["Valuation"] = unicorn_analysis["Valuation"].replace({"\$": "", "I
unicorn_analysis["Funding"] = unicorn_analysis["Funding"].replace({"\$": "", "B":

# Calculate ROI for each company
unicorn_analysis["ROI"] = ((unicorn_analysis["Valuation"] - unicorn_analysis["Fund

print(unicorn_analysis[["Company", "Valuation", "Funding", "ROI"]])
```

```
           Company  Valuation  Funding  ROI
0        Bytedance      180.0      NaN  NaN
1           SpaceX      100.0      NaN  NaN
2            SHEIN      100.0      NaN  NaN
3           Stripe       95.0      NaN  NaN
4           Klarna       46.0      NaN  NaN
...            ...        ...      ...  ...
1069      Zhaogang        1.0      NaN  NaN
1070  Zhuan Zhuan         1.0      NaN  NaN
1071      Zihaiguo        1.0      NaN  NaN
1072          Zopa        1.0      NaN  NaN
1073         Zwift        1.0      NaN  NaN

[1074 rows x 4 columns]
```

In [8]:
```
# Convert valuation and funding to numeric values (remove $ sign and 'B' or 'M')
def convert_to_numeric(value_str):
    if isinstance(value_str, str):
        value_str = value_str.replace("$", "").replace("B", "e9").replace("M", "e6
    return pd.to_numeric(value_str, errors="coerce")

unicorn_analysis["Valuation"] = unicorn_analysis["Valuation"].apply(convert_to_num
unicorn_analysis["Funding"] = unicorn_analysis["Funding"].apply(convert_to_numeric

# Calculate ROI for each company
unicorn_analysis["ROI"] = ((unicorn_analysis["Valuation"] - unicorn_analysis["Fund

print(unicorn_analysis[["Company", "Valuation", "Funding", "ROI"]])
```

```
          Company  Valuation  Funding  ROI
0       Bytedance      180.0      NaN  NaN
1          SpaceX      100.0      NaN  NaN
2           SHEIN      100.0      NaN  NaN
3          Stripe       95.0      NaN  NaN
4          Klarna       46.0      NaN  NaN
...           ...        ...      ...  ...
1069     Zhaogang        1.0      NaN  NaN
1070  Zhuan Zhuan        1.0      NaN  NaN
1071     Zihaiguo        1.0      NaN  NaN
1072         Zopa        1.0      NaN  NaN
1073        Zwift        1.0      NaN  NaN

[1074 rows x 4 columns]
```

In [10]:
```
# Convert "Valuation" and "Funding" columns to strings before data cleaning
unicorn_analysis["Valuation"] = unicorn_analysis["Valuation"].astype(str)
unicorn_analysis["Funding"] = unicorn_analysis["Funding"].astype(str)

# Data cleaning: Remove unwanted characters from "Valuation" and "Funding" columns
unicorn_analysis["Valuation"] = unicorn_analysis["Valuation"].str.replace('[\$\,]'
unicorn_analysis["Funding"] = unicorn_analysis["Funding"].str.replace('[\$\,]', ''

# Convert "Valuation" and "Funding" to numeric values (remove $ sign and 'B' or 'M
unicorn_analysis["Valuation"] = pd.to_numeric(unicorn_analysis["Valuation"], error
unicorn_analysis["Funding"] = pd.to_numeric(unicorn_analysis["Funding"], errors='c

# Calculate ROI for each company (ignoring rows with NaN in "Funding" or "Valuatio
unicorn_analysis["ROI"] = ((unicorn_analysis["Valuation"] - unicorn_analysis["Fund

print(unicorn_analysis[["Company", "Valuation", "Funding", "ROI"]])
```

```
          Company  Valuation  Funding  ROI
0       Bytedance      180.0      NaN  NaN
1          SpaceX      100.0      NaN  NaN
2           SHEIN      100.0      NaN  NaN
3          Stripe       95.0      NaN  NaN
4          Klarna       46.0      NaN  NaN
...           ...        ...      ...  ...
1069     Zhaogang        1.0      NaN  NaN
1070  Zhuan Zhuan        1.0      NaN  NaN
1071     Zihaiguo        1.0      NaN  NaN
1072         Zopa        1.0      NaN  NaN
1073        Zwift        1.0      NaN  NaN

[1074 rows x 4 columns]
```

In [11]:
```
## Tried to trouble shoot why I get non numeric values but cannot understand this
```

In [ ]:

```

```

In [ ]:

```

```