Business Case: Funding in Startups

1. Problem Statement

Context:

Founding a startup with a revolutionary idea may seem very attractive, but having insights and knowledge about the condition of startups founded in the past can help in shaping future startup ideas and positioning it well in today's markets, which can ultimately lead to increasing its performance many folds, in the real world.

Objective:

1. Uncover trends and insights that guide strategic decision-making. Consider analyzing the distribution of funding across different categories, markets, and regions to identify sectors with higher investment potential.
2. Explore the correlation between a startup's founding characteristics and its funding success, examining factors such as the funding rounds, funding types, and geographical locations. Additionally, assess the impact of economic factors on funding, and propose strategies for startups to optimize their funding journeys. This project has the potential to offer valuable insights for both aspiring entrepreneurs and investors in the dynamic landscape of startup financing.

Importing Required Libraries

In [4]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from plotly import graph_objects as go
```

Loading the given data-set

In [5]:
```
df = pd.read_csv('D:\\Scaler\\Scaler\\Fintech Domain Course\\Business Case\\Code based\\Fundi
                 encoding = "ISO-8859-1")
```

Note: As this CSV file contains special characters that are not part of the standard ASCII set. ISO-8859-1 is chosen because it supports a wider range of characters compared to ASCII.

In [6]:
```
df.head(2)
```

Out[6]:

	permalink	name	homepage_url	category_list	market	funding_tc
0	/organization/waywire	#waywire	http://www.waywire.com	|Entertainment|Politics|Social Media|News|	News	1
1	/organization/tv-communications	&TV Communications	http://enjoyandtv.com	|Games|	Games	4

2 rows × 39 columns

2. Data Exploration/EDA

2.1 Understanding the Variables/Features

In [7]:
```
df.shape
```

Out[7]:
```
(54294, 39)
```

Observation: The dataset has a total of 54294 rows and 39 columns.

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

Out[8]:
```
Index(['permalink', 'name', 'homepage_url', 'category_list', ' market ',
       ' funding_total_usd ', 'status', 'country_code', 'state_code', 'region',
       'city', 'funding_rounds', 'founded_at', 'founded_month',
       'founded_quarter', 'founded_year', 'first_funding_at',
       'last_funding_at', 'seed', 'venture', 'equity_crowdfunding',
       'undisclosed', 'convertible_note', 'debt_financing', 'angel', 'grant',
       'private_equity', 'post_ipo_equity', 'post_ipo_debt',
       'secondary_market', 'product_crowdfunding', 'round_A', 'round_B',
       'round_C', 'round_D', 'round_E', 'round_F', 'round_G', 'round_H'],
      dtype='object')
```

Observations: A few column names have white spaces

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54294 entries, 0 to 54293
Data columns (total 39 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   permalink             49438 non-null  object 
 1   name                  49437 non-null  object 
 2   homepage_url          45989 non-null  object 
 3   category_list         45477 non-null  object 
 4    market               45470 non-null  object 
 5    funding_total_usd    49438 non-null  object 
 6   status                48124 non-null  object 
 7   country_code          44165 non-null  object 
 8   state_code            30161 non-null  object 
 9   region                44165 non-null  object 
 10  city                  43322 non-null  object 
 11  funding_rounds        49438 non-null  float64
 12  founded_at            38554 non-null  object 
 13  founded_month         38482 non-null  object 
 14  founded_quarter       38482 non-null  object 
 15  founded_year          38482 non-null  float64
 16  first_funding_at      49438 non-null  object 
 17  last_funding_at       49438 non-null  object 
 18  seed                  49438 non-null  float64
 19  venture               49438 non-null  float64
 20  equity_crowdfunding   49438 non-null  float64
 21  undisclosed           49438 non-null  float64
 22  convertible_note      49438 non-null  float64
 23  debt_financing        49438 non-null  float64
 24  angel                 49438 non-null  float64
 25  grant                 49438 non-null  float64
 26  private_equity        49438 non-null  float64
 27  post_ipo_equity       49438 non-null  float64
 28  post_ipo_debt         49438 non-null  float64
 29  secondary_market      49438 non-null  float64
 30  product_crowdfunding  49438 non-null  float64
 31  round_A               49438 non-null  float64
 32  round_B               49438 non-null  float64
 33  round_C               49438 non-null  float64
 34  round_D               49438 non-null  float64
 35  round_E               49438 non-null  float64
 36  round_F               49438 non-null  float64
 37  round_G               49438 non-null  float64
 38  round_H               49438 non-null  float64
dtypes: float64(23), object(16)
memory usage: 16.2+ MB
```

Observation: A few columns representing date is not of date type

In [10]:
```
df.describe()
```

Out[10]:

	funding_rounds	founded_year	seed	venture	equity_crowdfunding	undisclosed	conver
count	49438.000000	38482.000000	4.943800e+04	4.943800e+04	4.943800e+04	4.943800e+04	4.94
mean	1.696205	2007.359129	2.173215e+05	7.501051e+06	6.163322e+03	1.302213e+05	2.33
std	1.294213	7.579203	1.056985e+06	2.847112e+07	1.999048e+05	2.981404e+06	1.43
min	1.000000	1902.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.00
25%	1.000000	2006.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.00
50%	1.000000	2010.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.00
75%	2.000000	2012.000000	2.500000e+04	5.000000e+06	0.000000e+00	0.000000e+00	0.00
max	18.000000	2014.000000	1.300000e+08	2.351000e+09	2.500000e+07	2.924328e+08	3.00

8 rows × 23 columns

2.2 Identifying and Handling Missing Values

In [11]:
```
missing_data = round((df.isnull().sum()* 100/df.shape[0]),2).sort_values(ascending=False)
missing_data
```

Out[11]:
```
state_code              44.45
founded_month           29.12
founded_year            29.12
founded_quarter         29.12
founded_at              28.99
city                    20.21
country_code            18.66
region                  18.66
 market                 16.25
category_list           16.24
homepage_url            15.30
status                  11.36
name                     8.95
post_ipo_debt            8.94
secondary_market         8.94
product_crowdfunding     8.94
round_A                  8.94
round_B                  8.94
permalink                8.94
round_C                  8.94
round_D                  8.94
round_E                  8.94
private_equity           8.94
round_F                  8.94
round_G                  8.94
post_ipo_equity          8.94
venture                  8.94
grant                    8.94
angel                    8.94
debt_financing           8.94
convertible_note         8.94
undisclosed              8.94
equity_crowdfunding      8.94
seed                     8.94
last_funding_at          8.94
first_funding_at         8.94
funding_rounds           8.94
 funding_total_usd       8.94
round_H                  8.94
dtype: float64
```

Observations:

- State code and various founding-related columns like founded_month, founded_year, and founded_quarter have the highest missing value percentages, suggesting potential challenges in data collection or recording for these attributes.
- Funding-related columns such as funding_rounds and funding_total_usd also exhibit notable missing values, indicating potential gaps in financial data or reporting.
- Other columns like market, category_list, and homepage_url have moderate missing value percentages, suggesting potential variations in data availability or completeness across different attributes.

Dropping missing values of 'name' columns

In [12]:
```
df = df[df['name'].notna()]
```

In [13]:
```
df.head(2)
```

Out[13]:

	permalink	name	homepage_url	category_list	market	funding_tc
0	/organization/waywire	#waywire	http://www.waywire.com	|Entertainment|Politics|Social Media|News|	News	1
1	/organization/tv-communications	&TV Communications	http://enjoyandtv.com	|Games|	Games	4

2 rows × 39 columns

In [14]:
```
round((df.isnull().sum()* 100/df.shape[0]),2).sort_values(ascending=False)
```

Out[14]:
```
state_code              38.99
founded_year            22.16
founded_quarter         22.16
founded_month           22.16
founded_at              22.02
city                    12.37
country_code            10.66
region                  10.66
 market                  8.03
category_list            8.01
homepage_url             6.98
status                   2.66
round_C                  0.00
post_ipo_debt            0.00
secondary_market         0.00
product_crowdfunding     0.00
round_A                  0.00
round_B                  0.00
permalink                0.00
round_D                  0.00
round_E                  0.00
private_equity           0.00
round_F                  0.00
round_G                  0.00
post_ipo_equity          0.00
venture                  0.00
grant                    0.00
angel                    0.00
debt_financing           0.00
convertible_note         0.00
undisclosed              0.00
equity_crowdfunding      0.00
name                     0.00
seed                     0.00
last_funding_at          0.00
first_funding_at         0.00
funding_rounds           0.00
 funding_total_usd       0.00
round_H                  0.00
dtype: float64
```

3. Data Cleaning

Dropping 'state_code' column

In [15]:
```
df.drop('state_code', axis=1,inplace=True)
```

Dropping missing values of 'founded_year' columns

In [16]:
```
df = df[df['founded_year'].notna()]
```

In [17]:
```
df.columns
```

Out[17]:
```
Index(['permalink', 'name', 'homepage_url', 'category_list', ' market ',
       ' funding_total_usd ', 'status', 'country_code', 'region', 'city',
       'funding_rounds', 'founded_at', 'founded_month', 'founded_quarter',
       'founded_year', 'first_funding_at', 'last_funding_at', 'seed',
       'venture', 'equity_crowdfunding', 'undisclosed', 'convertible_note',
       'debt_financing', 'angel', 'grant', 'private_equity', 'post_ipo_equity',
       'post_ipo_debt', 'secondary_market', 'product_crowdfunding', 'round_A',
       'round_B', 'round_C', 'round_D', 'round_E', 'round_F', 'round_G',
       'round_H'],
      dtype='object')
```

Renaming the odd columns by removing whitespace

In [18]:
```
df.rename(columns=lambda x: x.strip(), inplace=True)
df.columns
```

Out[18]:
```
Index(['permalink', 'name', 'homepage_url', 'category_list', 'market',
       'funding_total_usd', 'status', 'country_code', 'region', 'city',
       'funding_rounds', 'founded_at', 'founded_month', 'founded_quarter',
       'founded_year', 'first_funding_at', 'last_funding_at', 'seed',
       'venture', 'equity_crowdfunding', 'undisclosed', 'convertible_note',
       'debt_financing', 'angel', 'grant', 'private_equity', 'post_ipo_equity',
       'post_ipo_debt', 'secondary_market', 'product_crowdfunding', 'round_A',
       'round_B', 'round_C', 'round_D', 'round_E', 'round_F', 'round_G',
       'round_H'],
      dtype='object')
```

Dropping columns not important for data analysis

In [19]:
```
df.drop(['homepage_url', 'category_list'], axis=1, inplace= True)

df.columns
```

Out[19]:
```
Index(['permalink', 'name', 'market', 'funding_total_usd', 'status',
       'country_code', 'region', 'city', 'funding_rounds', 'founded_at',
       'founded_month', 'founded_quarter', 'founded_year', 'first_funding_at',
       'last_funding_at', 'seed', 'venture', 'equity_crowdfunding',
       'undisclosed', 'convertible_note', 'debt_financing', 'angel', 'grant',
       'private_equity', 'post_ipo_equity', 'post_ipo_debt',
       'secondary_market', 'product_crowdfunding', 'round_A', 'round_B',
       'round_C', 'round_D', 'round_E', 'round_F', 'round_G', 'round_H'],
      dtype='object')
```

deleting duplicate rows.

In [20]:
```
df = df.drop_duplicates()
```

In [21]:
```
df.shape
```

Out[21]:
```
(38481, 36)
```

Dropping missing values of 'status' column

In [22]:
```
df.status.value_counts(dropna=False)
```

Out[22]:
```
operating    32597
acquired      2971
closed        1995
NaN            918
Name: status, dtype: int64
```

In [23]:
```
df = df[df['status'].notna()]
```

In [24]:
```
round((df.isnull().sum()* 100/df.shape[0]),2).sort_values(ascending=False)
```

Out[24]:
```
city                    8.94
country_code            7.82
region                  7.82
market                  4.79
permalink               0.00
round_A                 0.00
grant                   0.00
private_equity          0.00
post_ipo_equity         0.00
post_ipo_debt           0.00
secondary_market        0.00
product_crowdfunding    0.00
round_C                 0.00
round_B                 0.00
debt_financing          0.00
round_D                 0.00
round_E                 0.00
round_F                 0.00
round_G                 0.00
angel                   0.00
undisclosed             0.00
convertible_note        0.00
name                    0.00
equity_crowdfunding     0.00
venture                 0.00
seed                    0.00
last_funding_at         0.00
first_funding_at        0.00
founded_year            0.00
founded_quarter         0.00
founded_month           0.00
founded_at              0.00
funding_rounds          0.00
status                  0.00
funding_total_usd       0.00
round_H                 0.00
dtype: float64
```

Data Type Conversion

Changing the values in column "funding_total_usd" from string to float

In [25]:
```
df['funding_total_usd'] = df['funding_total_usd'].str.strip().str.replace(",","")
df['funding_total_usd'] = df['funding_total_usd'].replace("-",0).astype("float")
```

In [26]:
```
df = pd.concat([df, pd.get_dummies(df['market'])], axis=1)
```

In [27]:
```
df.head()
```

Out[27]:

	permalink	name	market	funding_total_usd	status	country_code	region	city	fun
0	/organization/waywire	#waywire	News	1750000.0	acquired	USA	New York City	New York	
2	/organization/rock-your-paper	'Rock' Your Paper	Publishing	40000.0	operating	EST	Tallinn	Tallinn	
3	/organization/in-touch-network	(In)Touch Network	Electronics	1500000.0	operating	GBR	London	London	
4	/organization/r-ranch-and-mine	-R- Ranch and Mine	Tourism	60000.0	operating	USA	Dallas	Fort Worth	
7	/organization/0-6-com	0-6.com	Curated Web	2000000.0	operating	NaN	NaN	NaN	

5 rows × 769 columns

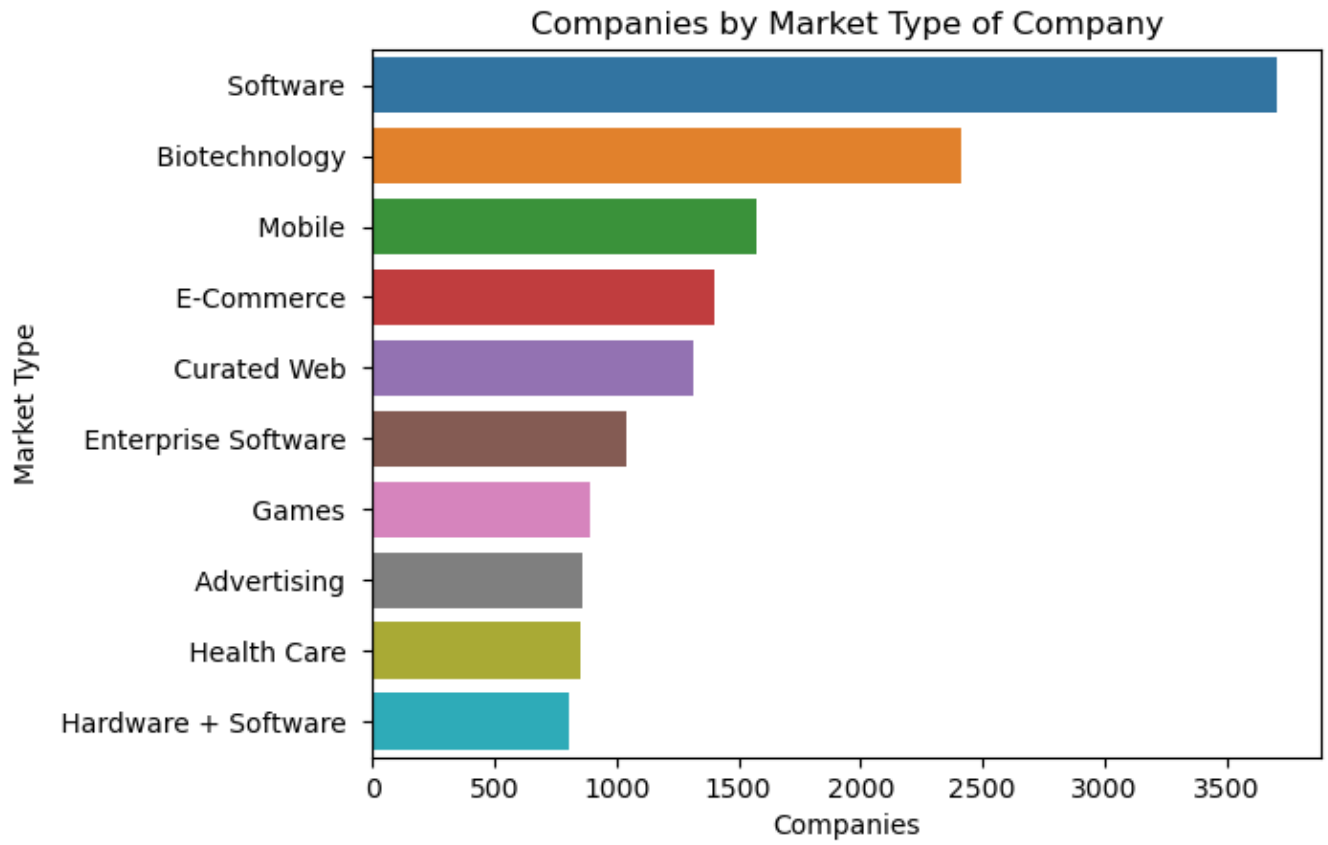
4. Data Analysis/Visualization

Q1. What is the relation of different startups with different markets/sectors?

The Top 10 Markets/Sectors that are generating the most startups

In [28]:
```
top_market = sns.countplot(y="market",data=df,
             order=df.market.value_counts().iloc[:10].index)
top_market.set_ylabel("Market Type")
top_market.set_xlabel("Companies")
top_market.set_title("Companies by Market Type of Company")
```

Out[28]:
```
Text(0.5, 1.0, 'Companies by Market Type of Company')
```

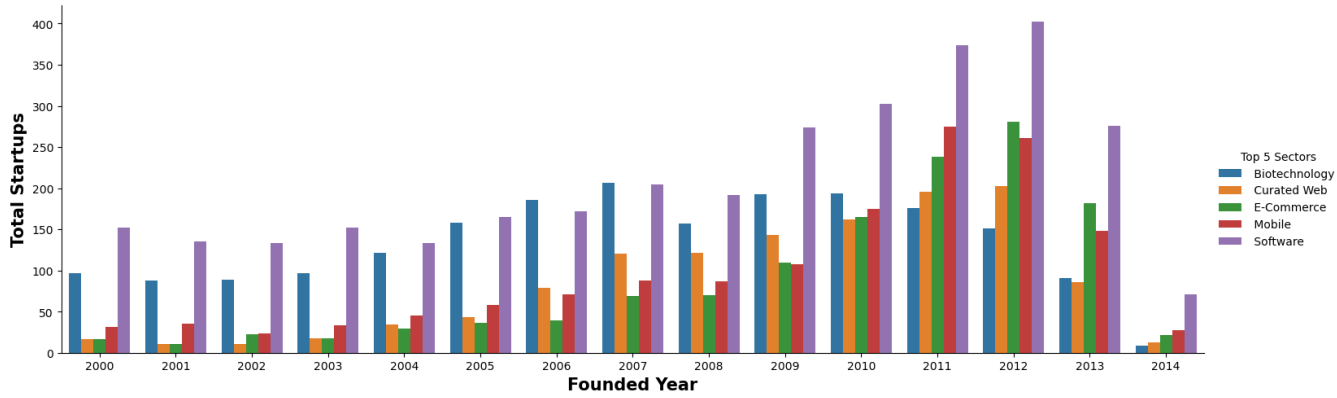


Observation: The software sector emerges as the frontrunner in startup funding, with biotechnology and mobile markets closely trailing behind

Distribution of top-5 startup sector after year 2020

In [29]:

```
strtup_invest_q4 = df[['founded_year','market','permalink']].dropna(axis=0,how='any')
strtup_invest_q4['founded_year'] = strtup_invest_q4['founded_year'].astype('int')
strtup_invest_q4 = strtup_invest_q4.groupby(['founded_year','market'],as_index='False').count

strtup_invest_q4.reset_index(inplace=True)
top_5_sectors = strtup_invest_q4.groupby('market').agg(total=pd.NamedAgg(column="permalink",
strtup_invest_q4 = strtup_invest_q4[strtup_invest_q4['market'].isin(top_5_sectors)]

g= sns.catplot(data=strtup_invest_q4.query('founded_year>1999'),x='founded_year',y='permalink
            hue='market',kind='bar',legend='Hello' )
g.set_xlabels('Founded Year',size=15,weight='bold')
g.set_ylabels('Total Startups',size=15,weight='bold')
g._legend.set_title(title = 'Top 5 Sectors')
plt.show()
```



Observation: Post-2020, the top five sectors driving funding and startup activity comprise biotechnology, curated web, e-commerce, mobile, and software

Q2. What is the relation of different startups with funding received?

Highest funded sectors

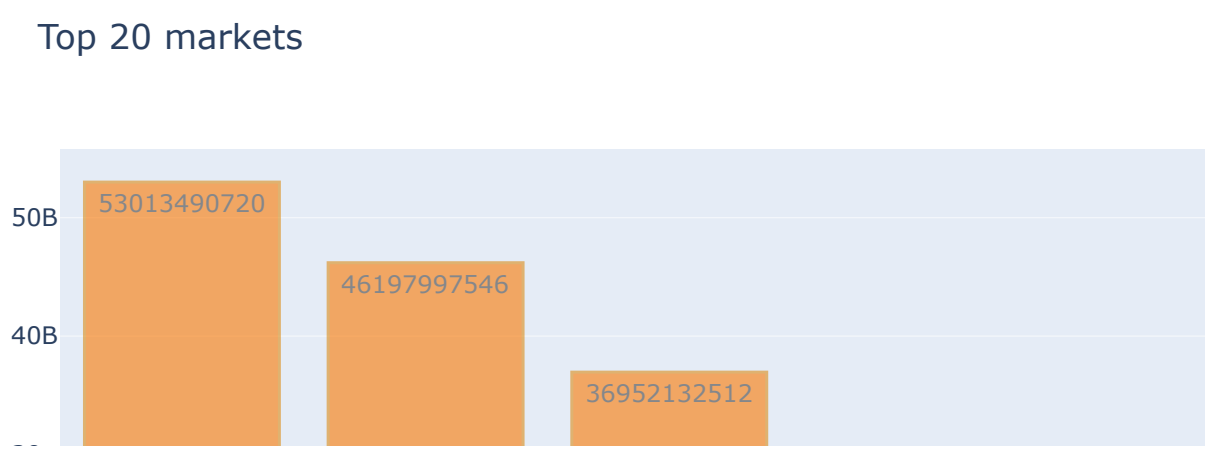
In [30]:
```
top_funded_market=pd.DataFrame(df[['market','funding_total_usd']])
top_funded_market=top_funded_market.groupby('market')[['funding_total_usd']].sum().sort_value

top_funded_market
```

Out[30]:

	market	funding_total_usd
0	Biotechnology	5.301349e+10
1	Mobile	4.619800e+10
2	Software	3.695213e+10
3	Clean Technology	2.706090e+10
4	Health Care	2.239594e+10
...	...	...
728	Social Innovation	0.000000e+00
729	Debt Collecting	0.000000e+00
730	ICT	0.000000e+00
731	Elder Care	0.000000e+00
732	Dietary Supplements	0.000000e+00

733 rows × 2 columns

In [31]:
```
data = top_funded_market
y=top_funded_market['funding_total_usd'][:10]
x=top_funded_market['market'][:20]

fig_2 = go.Figure(data=[
    go.Bar(name='total funding', x=x, y=y, text= y,textposition='auto')])

# Customize aspect
fig_2.update_traces(marker_color='rgb(250,120,2)', marker_line_color='rgb(220,140,23)',
                  marker_line_width=1.5, opacity=0.6)
fig_2.update_layout(title_text='Top 20 markets')

fig_2.show()
```



Observation:

- The highest funded sectors, namely Biotechnology, Mobile, and Software, have secured substantial investment amounts, with Biotechnology leading the pack, followed closely by Mobile and Software.
- 'Clean Technology' and Health Care also feature prominently among the top funded sectors, indicating significant investor interest and financial support in these industries.

Q3. What is the relation of different startups with status (operating, closed, etc.)?

Status of different startups

In [32]:
```
status=pd.DataFrame(df['status'].value_counts().reset_index())
status
```

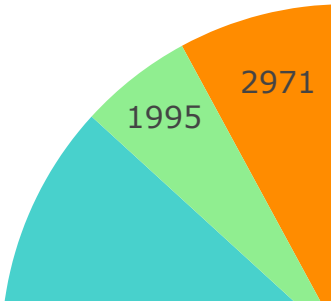
Out[32]:

	index	status
0	operating	32597
1	acquired	2971
2	closed	1995

Observation: A very high number of startups are operating well

In [33]:
```
colors = [ 'mediumturquoise', 'darkorange', 'lightgreen']

fig_3 = go.Figure(data=[go.Pie(labels=status['index'],
                               values=status['status'],pull=[0, 0, 0])])
```

```
fig_3.update_traces(hoverinfo='label+percent', textinfo='value', textfont_size=15,
                  marker=dict(colors=colors))
fig_3.show()
```



Observations:

- The majority of startups, constituting 32,597 entities, are currently in the "operating" phase, indicating active business operations and ongoing market presence.
- A significant number of startups, totaling 2,971, have transitioned to the "acquired" status, suggesting successful exits through acquisition by larger companies or investors.
- Additionally, 1,995 startups are categorized as "closed," reflecting instances where businesses have ceased operations, highlighting the inherent risks and challenges in the startup landscape.

Q3. What is the total fundings considering all the funding types/channels and what is the relation between the status and total fundings?

Total Funding considering all the funding types (channels)

In [34]:
```
# 'funding_total_usd' column : this column is the sum of all fundings collected from one or m
# However, most of the values are string type and needs to be numeric (float)
df['funding_total_usd'].apply(type).value_counts()
```

Out[34]:
```
<class 'float'>    37563
Name: funding_total_usd, dtype: int64
```

In [35]:
```
#let's create a new column named 'total_funding' and fill it with the sum all the fundings co

funding_channels=['seed','venture', 'equity_crowdfunding', 'undisclosed', 'convertible_note',
                  'grant', 'private_equity', 'post_ipo_equity', 'post_ipo_debt', 'secondary_m
```

```
df['total_funding']=0
for c in funding_channels:
    df['total_funding']=df['total_funding']+df[c]

df[['funding_total_usd','total_funding']].head()
```

Out[35]:

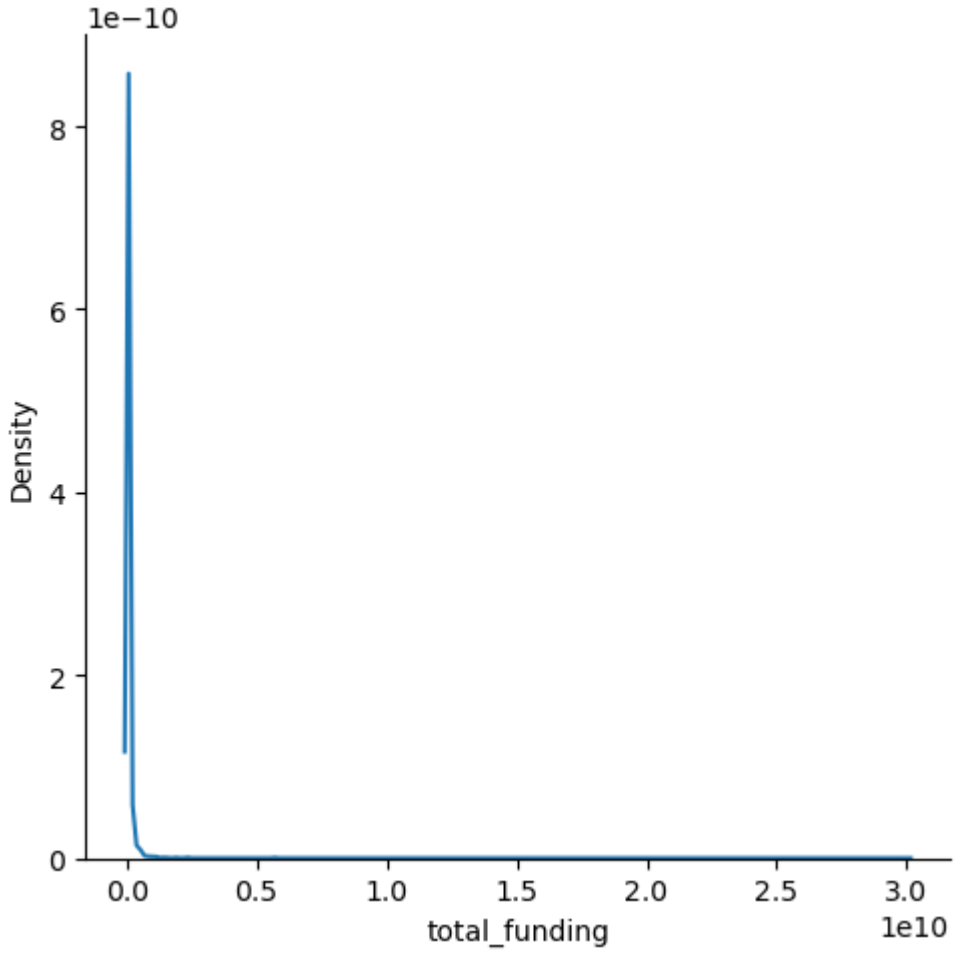
	funding_total_usd	total_funding
0	1750000.0	1750000.0
2	40000.0	40000.0
3	1500000.0	1500000.0
4	60000.0	60000.0
7	2000000.0	2000000.0

In [36]:
```
len(df[df['total_funding']==0])/len(df)
```

Out[36]:
```
0.16476319782764956
```

In [37]:
```
# Plotting the distribution
sns.displot(df[df['total_funding'] > 0]['total_funding'], kind='kde')
```

Out[37]:
```
<seaborn.axisgrid.FacetGrid at 0x219c186b5e0>
```



In [38]:
```
df['total_funding'].describe()
```

Out[38]:
```
count    3.756300e+04
mean     1.396056e+07
std      1.711716e+08
min      0.000000e+00
25%      6.503500e+04
50%      1.000000e+06
75%      7.088885e+06
max      3.007950e+10
Name: total_funding, dtype: float64
```

Observation:

- The mean funding amount across these startups is approximately 13.96 million dollar , suggesting a substantial average investment level in the startup ecosystem.
- However, the standard deviation of 17.12 million indicates a wide variability in funding amounts, reflecting diverse funding levels across different startups.
- Notably, the minimum funding amount is 0 dollar, indicating instances where startups have received no funding or where data may be missing or unavailable.
- The 25th percentile (Q1) funding amount is 65,035 dollar, indicating that a quarter of startups have received funding amounts below this value.
- Similarly, the median funding amount (50th percentile or Q2) is 1 million dollar, suggesting that half of the startups have received funding amounts at or below this value.
- The 75th percentile (Q3) funding amount is 7.09 million dollar , indicating that three-quarters of startups have received funding amounts below this value.
- The maximum funding amount recorded is 30.08 billion dollar , highlighting instances of exceptionally high funding levels received by certain startups, which significantly skew the average and demonstrate the presence of outliers in dataset.

In [39]:
```
df2=df[['status','founded_year','total_funding']]
df2.head()
```

Out[39]:

	status	founded_year	total_funding
0	acquired	2012.0	1750000.0
2	operating	2012.0	40000.0
3	operating	2011.0	1500000.0
4	operating	2014.0	60000.0
7	operating	2007.0	2000000.0

Observation:

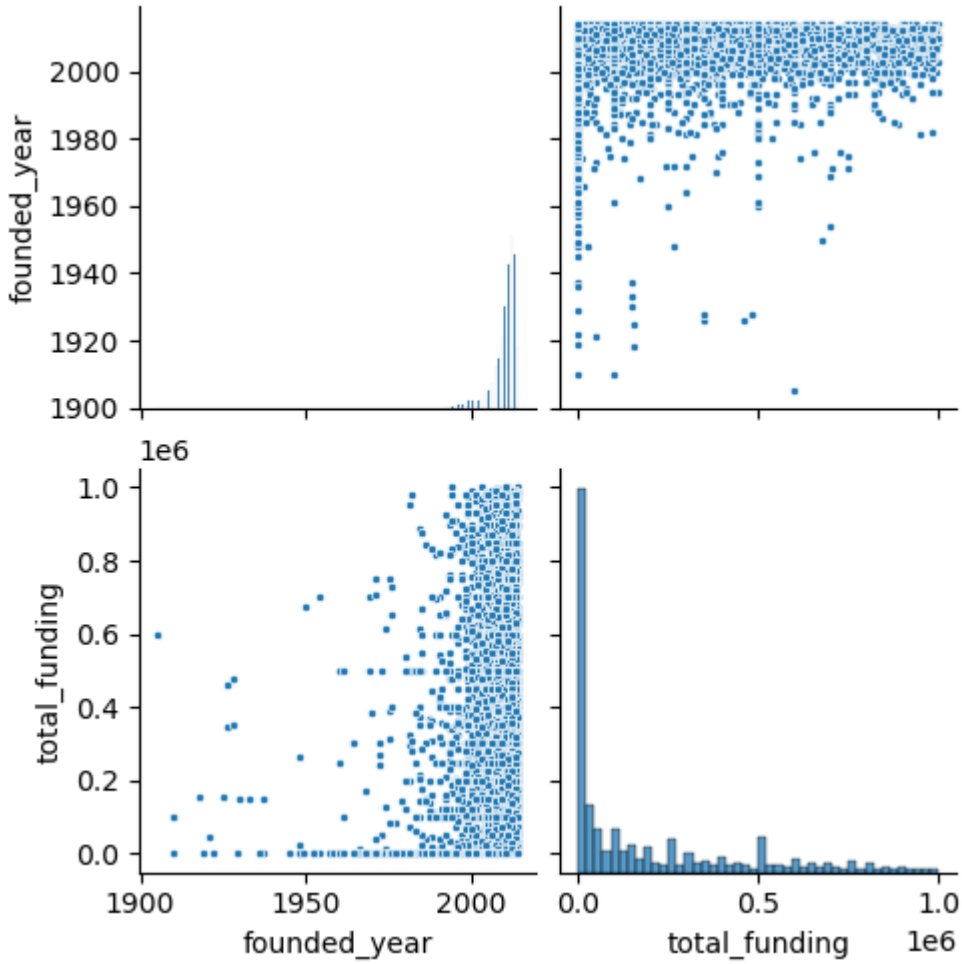
- Acquired startups have generally received higher funding amounts compared to operating startups.
- Startups acquired in various years, such as 2012, have secured substantial funding, with funding amounts typically exceeding those received by operating startups in the same year.
- Conversely, operating startups across different founding years have received comparatively lower funding amounts, indicating potential challenges in accessing significant investment capital compared to their acquired counterparts.
- The disparity in funding levels between acquired and operating startups underscores the impact of acquisition events on fundraising success and highlights the importance of understanding the funding dynamics within different startup cohorts.

In [40]:
```
df.groupby(['status'])['total_funding'].describe()
```

Out[40]:

status	count	mean	std	min	25%	50%	75%	max
acquired	2971.0	2.152392e+07	1.166976e+08	0.0	1250000.0	6450000.0	20000000.0	5.700000e+09
closed	1995.0	6.167956e+06	3.923606e+07	0.0	25000.0	400000.0	3000000.0	1.567504e+09
operating	32597.0	1.374813e+07	1.800542e+08	0.0	54068.0	1000000.0	6215250.0	3.007950e+10

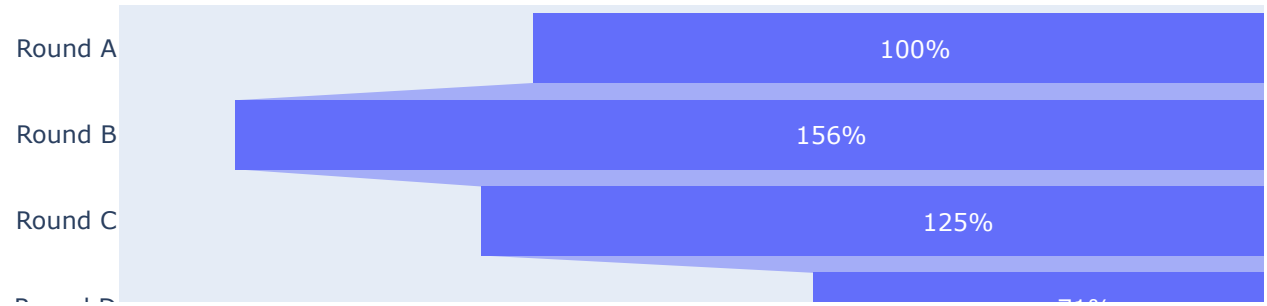
In [41]:
```
g = sns.pairplot(df2[df2['total_funding']<1000000],markers='.')
```



Q4. How many rounds of funding each of these categories received?

In [42]:
```
#segregating the dataframe on the basis of status
acquired_df=df.query('status == "acquired"')
operating_df=df.query('status == "operating"')
closed_df=df.query('status == "closed"')

#Plotting a funnel chart to understand the proportion
fig_5 = go.Figure()

fig_5.add_trace(go.Funnel(
    name = 'acquired',
    y = ["Round A", "Round B","Round C","Round D",
         "Round E","Round F","Round G","Round H"],
    x = [acquired_df['round_A'].mean(),acquired_df['round_B'].mean(),acquired_df['round_C'].m
         acquired_df['round_D'].mean(),acquired_df['round_E'].mean(),
         acquired_df['round_F'].mean(),acquired_df['round_G'].mean(),acquired_df['round_H'].m
    textinfo = "percent initial"))

fig_5.add_trace(go.Funnel(
    name = 'operating',
    orientation = "h",
    y = ["Round A", "Round B","Round C","Round D",
         "Round E","Round F","Round G","Round H"],
```

```
    x = [operating_df['round_A'].mean(),operating_df['round_B'].mean(),operating_df['round_C'
         operating_df['round_D'].mean(),operating_df['round_E'].mean(),
         operating_df['round_F'].mean(),operating_df['round_G'].mean(),operating_df['round_H'
    textposition = "inside",
    textinfo = "percent previous"))


fig_5.add_trace(go.Funnel(
    name = 'closed',
    orientation = "h",
    y = ["Round A", "Round B","Round C","Round D",
         "Round E","Round F","Round G","Round H"],
    x = [closed_df['round_A'].mean(),closed_df['round_B'].mean(),closed_df['round_C'].mean(),
         closed_df['round_D'].mean(),closed_df['round_E'].mean(),
         closed_df['round_F'].mean(),closed_df['round_G'].mean(),closed_df['round_H'].mean()]
    textposition = "outside",
    textinfo = "percent total"))

fig_5.show()
```



In [60]:

```
# Create DataFrame
data = {
    'Status': ['Acquired', 'Operating', 'Closed'],
    'Round A': [acquired_df['round_A'].mean(), operating_df['round_A'].mean(), closed_df['rou
    'Round B': [acquired_df['round_B'].mean(), operating_df['round_B'].mean(), closed_df['rou
    'Round C': [acquired_df['round_C'].mean(), operating_df['round_C'].mean(), closed_df['rou
    'Round D': [acquired_df['round_D'].mean(), operating_df['round_D'].mean(), closed_df['rou
    'Round E': [acquired_df['round_E'].mean(), operating_df['round_E'].mean(), closed_df['rou
    'Round F': [acquired_df['round_F'].mean(), operating_df['round_F'].mean(), closed_df['rou
    'Round G': [acquired_df['round_G'].mean(), operating_df['round_G'].mean(), closed_df['rou
    'Round H': [acquired_df['round_H'].mean(), operating_df['round_H'].mean(), closed_df['rou
}

df_table = pd.DataFrame(data)
df_table
```

Out[60]:

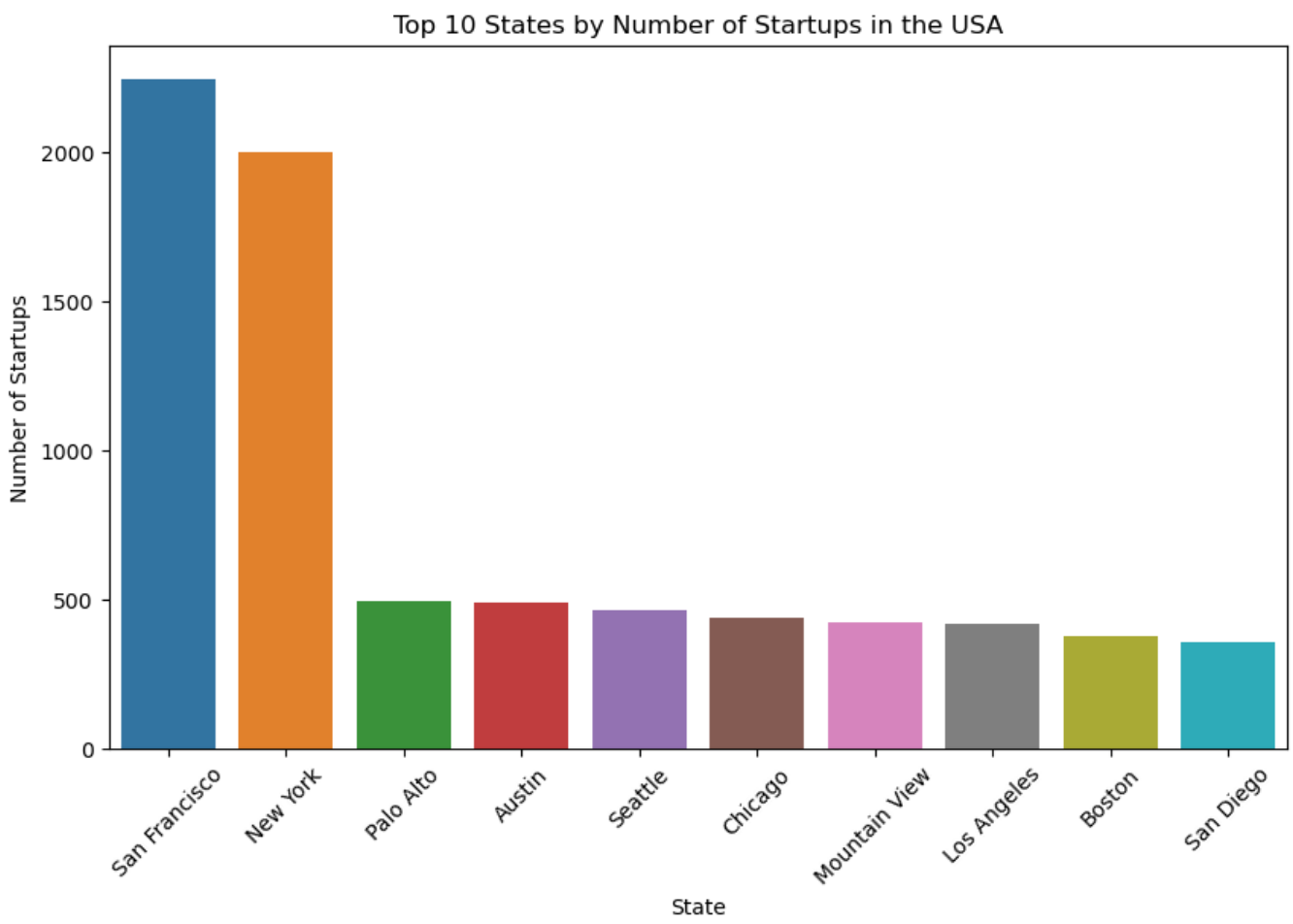
	Status	Round A	Round B	Round C	Round D	Round E	Round F	Rou
0	Acquired	2.308948e+06	3.602019e+06	2.880537e+06	1.646160e+06	582453.171323	172191.187816	28778.18
1	Operating	1.250589e+06	1.551752e+06	1.326464e+06	8.719082e+05	423779.766911	218897.238151	83197.35
2	Closed	9.493801e+05	1.155056e+06	6.134484e+05	3.006161e+05	168611.528822	215789.473684	25062.65

Observation:

- Acquired startups have the highest mean funding amounts across all funding rounds compared to operating and closed startups, indicating a greater level of investment and financial support in the acquisition process.
- Round B appears to be the most heavily funded round for all three statuses, with acquired startups receiving the highest mean funding amount, followed by operating and closed startups.
- Operating startups exhibit moderate mean funding amounts across all rounds, suggesting ongoing investment and growth opportunities despite lower funding levels compared to acquired startups.
- Closed startups have the lowest mean funding amounts across all rounds, indicating potential challenges or limitations in accessing sufficient funding to sustain operations and growth, contributing to their closure status.

Q5. What are the regions in the USA with the most startups?

In [44]:
```
#dataframe with country code USA
USA = pd.DataFrame(df.query('country_code == "USA"')['city'].value_counts(),columns=['city'])
USA.reset_index(inplace=True)


# Selecting top 10 states
top_10_states = USA.head(10)

# Plotting using Seaborn
plt.figure(figsize=(10, 6))
sns.barplot(x='index', y='city', data=top_10_states)
plt.title('Top 10 States by Number of Startups in the USA')
plt.xlabel('State')
plt.ylabel('Number of Startups')
plt.xticks(rotation=45)  # Rotating x-axis labels for better readability
plt.show()
```

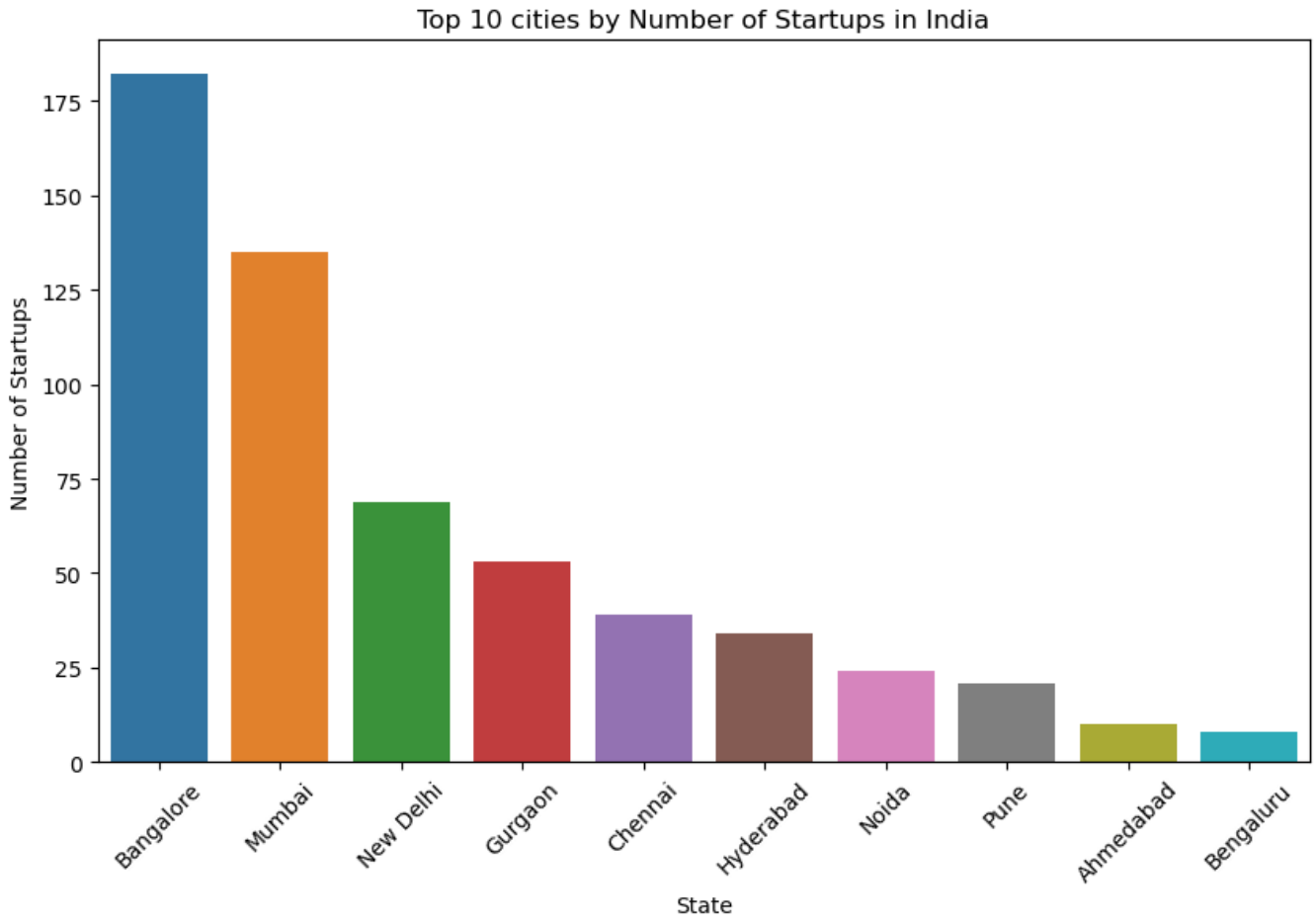


Observations: San Francisco , 'New York and 'Palo Alto' are the top states in the USA with the highest startup count.

Q6. What are the regions in India with the most startups?

Regions in India with the most startups

In [45]:
```
#dataframe with country code IND
India = pd.DataFrame(df.query('country_code == "IND"')['city'].value_counts(),columns=['city'
India.reset_index(inplace=True)


# Selecting top 10 states
top_10_states = India.head(10)

# Plotting using Seaborn
plt.figure(figsize=(10, 6))
sns.barplot(x='index', y='city', data=top_10_states)
plt.title('Top 10 cities by Number of Startups in India')
plt.xlabel('State')
plt.ylabel('Number of Startups')
plt.xticks(rotation=45)  # Rotating x-axis labels for better readability
plt.show()
```

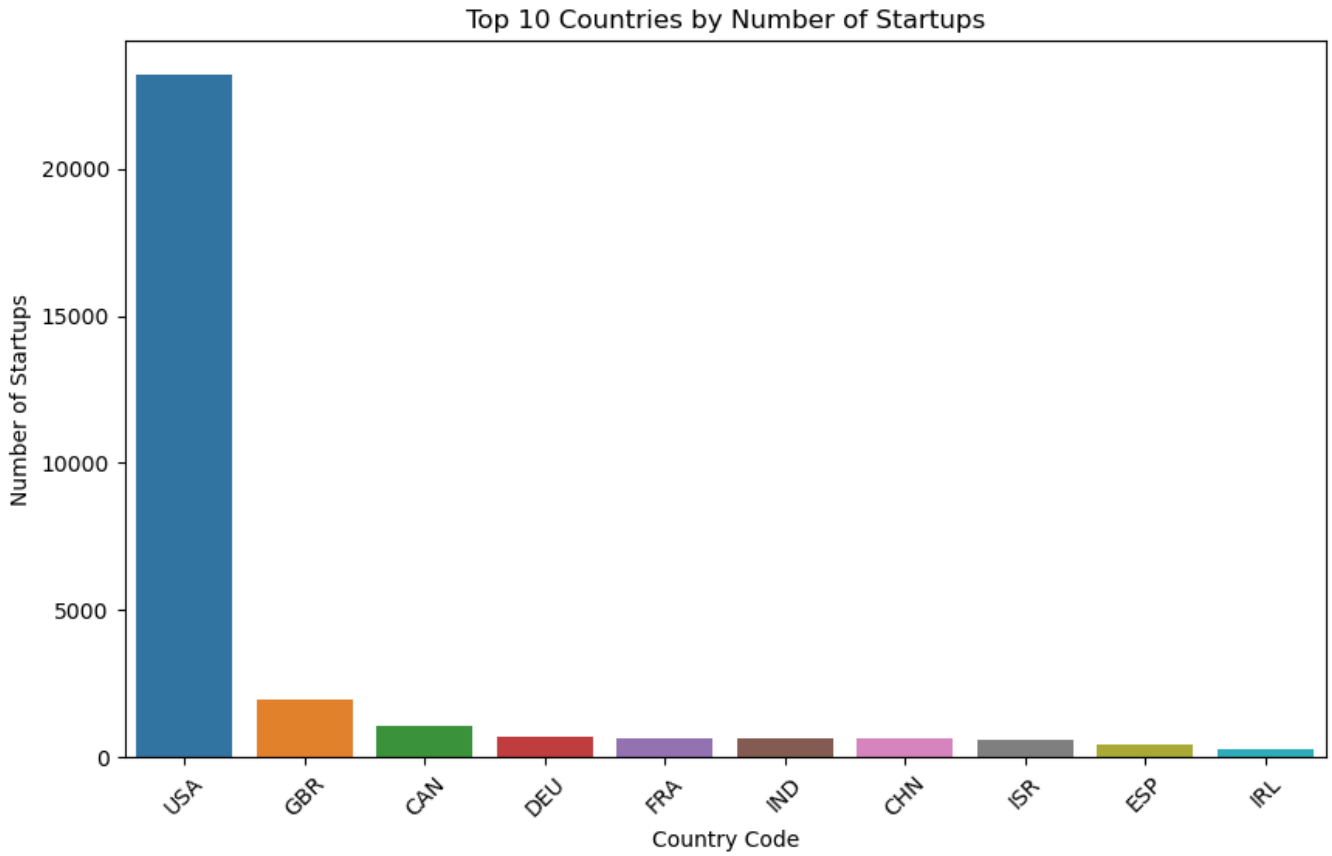


Observations: Banglore, Mumbai and New Delhi are the top cities in India with the highest startup count.

Q7. What are the top countries with the most startups?

Top countries in the world with the highest startup count

In [46]:
```
# Counting the number of startups by country code
top_countries = df['country_code'].value_counts().head(10)

# Creating a DataFrame from the series
top_countries_df = pd.DataFrame(top_countries).reset_index()
top_countries_df.columns = ['country_code', 'num_startups']

# Creating a DataFrame from the series
top_countries_df = top_countries.reset_index()
top_countries_df.columns = ['country_code', 'num_startups']

# Plotting using Seaborn
plt.figure(figsize=(10, 6))
sns.barplot(x='country_code', y='num_startups', data=top_countries_df)
plt.title('Top 10 Countries by Number of Startups')
plt.xlabel('Country Code')
plt.ylabel('Number of Startups')
plt.xticks(rotation=45)  # Rotating x-axis labels for better readability
plt.show()
```



In [47]:
```
# Calculate total number of startups
total_startups = top_countries_df['num_startups'].sum()
# Calculate percentage of startups for each country
top_countries_df['percentage'] = (top_countries_df['num_startups'] / total_startups) * 100
top_countries_df
```

Out[47]:

	country_code	num_startups	percentage
0	USA	23182	76.746342
1	GBR	1968	6.515262
2	CAN	1074	3.555585
3	DEU	722	2.390254
4	FRA	669	2.214792
5	IND	667	2.208171
6	CHN	653	2.161822
7	ISR	591	1.956565
8	ESP	422	1.397073
9	IRL	258	0.854135

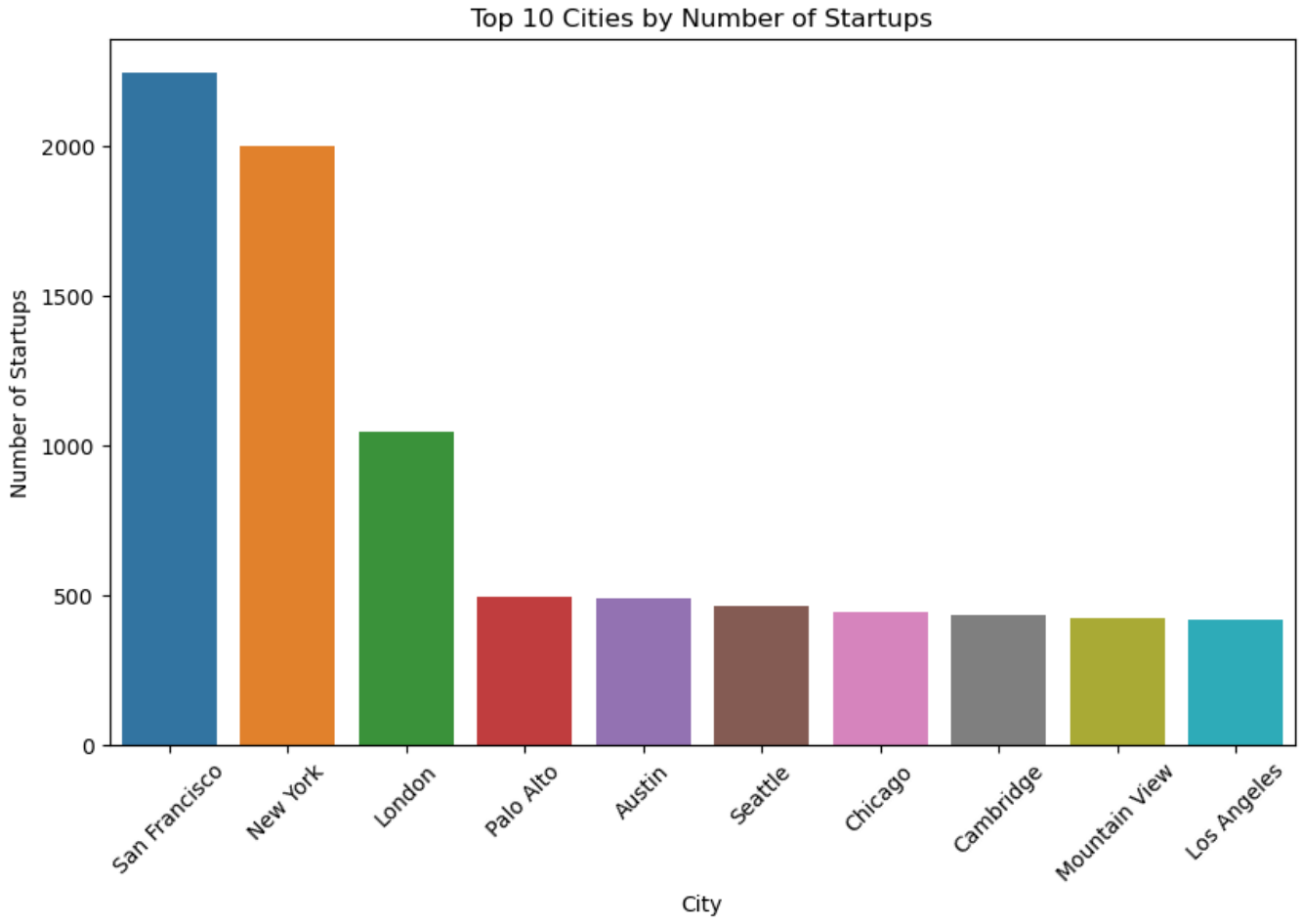
Observation:

- The USA dominates the startup landscape with the highest number of startups, accounting for approximately 76.75% of the total startups analyzed in the dataset, underscoring its position as a global leader in entrepreneurship and innovation.
- Following the USA, the United Kingdom (GBR) and Canada (CAN) represent significant startup hubs, with 6.52% and 3.56% of the total startups, respectively, reflecting their robust entrepreneurial ecosystems and supportive business environments.
- Other countries such as Germany (DEU), France (FRA), India (IND), and China (CHN) also exhibit notable startup activity, each contributing around 2-3% of the total startups, highlighting their emergence as key players in the global startup landscape.

- Israel (ISR), Spain (ESP), and Ireland (IRL) round out the top ten countries with smaller but still significant contributions to the total startup count, showcasing diverse geographical hotspots for entrepreneurial growth and innovation.

Q7. What are the top cities with the most startups?

In [48]:
```
# Counting the number of startups by region
top_countries = df['city'].value_counts().head(10)

# Creating a DataFrame from the series
top_countries_df = pd.DataFrame(top_countries).reset_index()
top_countries_df.columns = ['city', 'num_startups']

# Creating a DataFrame from the series
top_countries_df = top_countries.reset_index()
top_countries_df.columns = ['city', 'num_startups']

# Plotting using Seaborn
plt.figure(figsize=(10, 6))
sns.barplot(x='city', y='num_startups', data=top_countries_df)
plt.title('Top 10 Cities by Number of Startups')
plt.xlabel('City')
plt.ylabel('Number of Startups')
plt.xticks(rotation=45)  # Rotating x-axis labels for better readability
plt.show()
```



Observations: Top cities are San Francisco, New York and London.

Q7. What is the year of founding for startups?

Founding years of startups

As the values in 'founded_at' column starting from 1902 which of no relevance to the data analysis. We will consider years after 1950

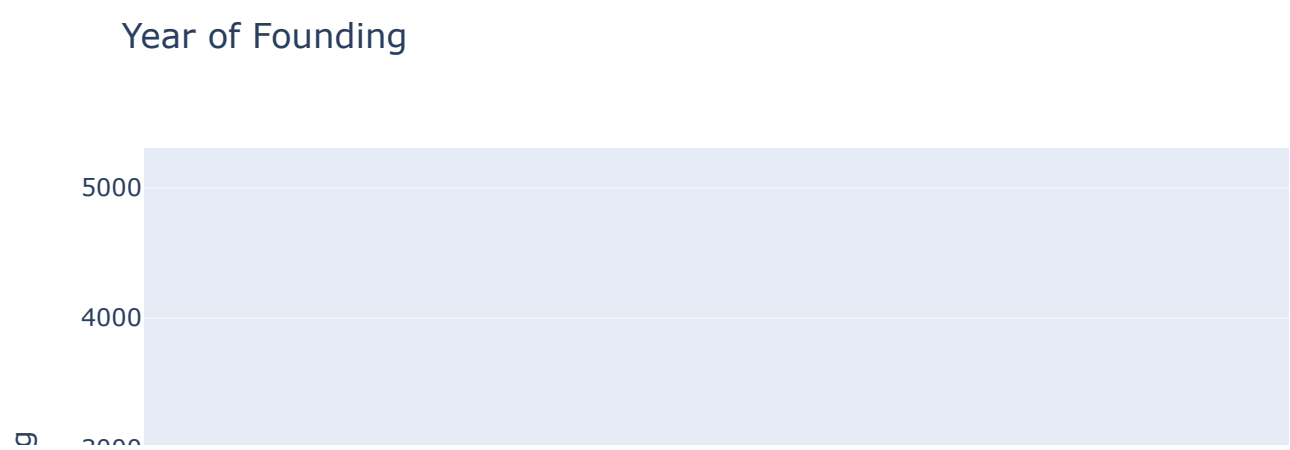
In [49]:
```
df['founded_at'].min()
```

Out[49]:
```
'1902-01-01'
```

In [50]:
```
df['founded_at'].max()
```

Out[50]:
```
'2014-12-13'
```

In [51]:
```
# Change the datatype of 'founded_at' column to datetime
df['founded_at'] = pd.to_datetime(df['founded_at'], errors='coerce')

# Drop rows with Null values in 'founded_at' column
df.dropna(subset=['founded_at'], inplace=True)

# Extract the year from 'founded_at' column and filter for years starting from 1950
foundation = pd.DataFrame({'year': df['founded_at'][df['founded_at'].dt.year >= 1950].dt.year

# Count the number of startups founded each year
startup_counts = foundation['year'].value_counts().sort_index()

# Plotting using Plotly
fig = go.Figure([go.Bar(x=startup_counts.index, y=startup_counts.values, marker_color='red')]
fig.update_layout(title='Year of Founding', xaxis=dict(title='Year'), yaxis=dict(title='Count
fig.show()
```



Observation:

- The year 2012 stands out with the highest number of startups, indicating a significant influx of entrepreneurial activity during that period.

- Moreover, there is a noticeable surge in the number of new startups from 2000 to 2014, reflecting a period of pronounced growth and expansion in the startup ecosystem.

5. Feature Engineering (Creating New Features)

Q8. Identify the startups that:

a. got funding in less than 1 year

b. got funded after more than 20 years

c. got funded before they were founded

8.a Startups that received funding in less than 1 year

In [52]:
```
# Change the datatype of 'first_funding_at' column to datetime
df['first_funding_at'] = pd.to_datetime(df['first_funding_at'], errors='coerce')

# Drop rows with Null values in 'first_funding_at' column
df.dropna(subset=['first_funding_at'], inplace=True)

# Defining and creating a new feature "difference" between the date of foundation and first f
df['difference'] = (df['first_funding_at'] - df['founded_at']).dt.days

# Extracting positive values only and saving in one_year DataFrame
one_year = df[(df['difference'] > 0) & (df['difference'] < 365)].copy()  # Make a copy to avo

# Sorting DataFrame by 'funding_total_usd' in descending order
one_year.sort_values(by='funding_total_usd', ascending=False, inplace=True)

# Creating 'difference_level' column and setting its values
one_year['difference_level'] = 0  # Initialize 'difference_level' column with 0

# Reordering columns and resetting index
one_year = one_year[['name', 'market', 'funding_total_usd',  'founded_at', 'first_funding_at'

# Displaying the resulting DataFrame
one_year
```

Out[52]:

	name	market	funding_total_usd	founded_at	first_funding_at	status	country_code
0	Facebook	Communities	2.425700e+09	2004-02-04	2004-09-01	operating	USA
1	Uber	Transportation	1.507450e+09	2009-03-01	2009-08-01	operating	USA
2	Fisker Automotive	Automotive	1.451000e+09	2008-01-01	2008-01-14	acquired	USA
3	Cloudera	Analytics	1.201000e+09	2008-10-13	2009-03-16	operating	USA
4	Alibaba	E-Commerce	1.112000e+09	1999-06-01	1999-10-01	operating	CHN
...	...	...	...	...	...	...	...
9777	The Butler	NaN	0.000000e+00	2014-02-01	2014-03-23	operating	ESP
9778	The Bucket BBQ	Hospitality	0.000000e+00	2013-02-19	2013-11-04	operating	USA
9779	emere	NaN	0.000000e+00	2012-05-30	2012-09-26	operating	USA
9780	Cupid-Labs	Local	0.000000e+00	2013-01-01	2013-05-15	operating	UKR
9781	Buzzstarter Inc	Software	0.000000e+00	2013-09-01	2014-03-01	operating	USA

9782 rows × 7 columns

In [53]:
```
count_companies = len(one_year)

print("Number of companies that received funding within 1 year of founding date:", count_comp
```

```
Number of companies that received funding within 1 year of founding date: 9782
```

8.b Received funding in more than 20 years

In [54]:
```
# Change the datatype of 'first_funding_at' column to datetime
df['first_funding_at'] = pd.to_datetime(df['first_funding_at'], errors='coerce')

# Drop rows with Null values in 'first_funding_at' column
df.dropna(subset=['first_funding_at'], inplace=True)

# Define the difference between the date of foundation and first funding
df['difference'] = (df['first_funding_at'] - df['founded_at']).dt.days

# Extracting positive values only and saving in more_than_20_years DataFrame
more_than_20_years = df[df['difference'] > 20*365].copy()  # Make a copy to avoid SettingWith

# Sorting DataFrame by 'funding_total_usd' in descending order
more_than_20_years.sort_values(by='funding_total_usd', ascending=False, inplace=True)

# Creating 'difference_level' column and setting its values
more_than_20_years['difference_level'] = 20  # Initialize 'difference_level' column with 20 (

# Reordering columns and resetting index
more_than_20_years = more_than_20_years[['name', 'market', 'funding_total_usd',  'founded_at'

# Count of companies that received funding more than 20 years after founding date
count_companies_more_than_20_years = len(more_than_20_years)

# Displaying the resulting DataFrame
more_than_20_years
```

Out[54]:

	name	market	funding_total_usd	founded_at	first_funding_at	status	country_code
0	Verizon Communications	Mobile	3.007950e+10	1983-10-07	2010-01-26	operating	USA
1	First Data Corporation	Trading	3.500000e+09	1971-01-01	2014-07-03	operating	USA
2	Zebra Technologies	Enterprise Software	2.000000e+09	1991-01-01	2014-09-16	operating	USA
3	Quad/Graphics	Local Businesses	1.900000e+09	1971-01-01	2014-04-28	operating	USA
4	Xerox	Hardware + Software	1.100000e+09	1906-01-01	2012-03-15	operating	USA
...	...	...	...	...	...	...	...
971	inGenius Engineering	Governments	0.000000e+00	1989-02-19	2014-06-18	operating	CAN
972	Informaat	Consulting	0.000000e+00	1986-01-01	2013-09-03	operating	NLD
973	InComm	Games	0.000000e+00	1992-01-01	2012-08-31	operating	USA
974	Sterling Heights Dentist	Local Businesses	0.000000e+00	1975-01-01	2013-01-01	operating	USA
975	1-800-DENTIST	Health and Wellness	0.000000e+00	1986-01-01	2010-08-19	operating	USA

976 rows × 7 columns

In [55]:
```
# Print count of companies that received funding more than 20 years after founding date
print("Number of companies that received funding more than 20 years after founding date:", co
```

```
Number of companies that received funding more than 20 years after founding date: 976
```

Observations:

- A significant number of companies were able to secure funding within a relatively short period after their founding date, potentially reflecting investor confidence in their business models, products, or services. It suggests that these startups were able to attract investors quickly, which could be attributed to various factors such as innovative ideas, strong founding teams, market potential, or early traction.
- A smaller proportion of startups were able to secure funding after a considerably longer period since their founding date, indicating a different investment scenario or business trajectory. Startups receiving funding after more than 20 years may represent companies that have undergone significant growth, innovation, or pivots over time, leading to renewed investor interest or strategic partnerships.

8.c Received funded before they were founded

In [56]:
```
# Change the datatype of 'first_funding_at' column to datetime
df['first_funding_at'] = pd.to_datetime(df['first_funding_at'], errors='coerce')
df['founded_at'] = pd.to_datetime(df['founded_at'], errors='coerce')

# Define the difference between the date of foundation and first funding
df['difference'] = (df['first_funding_at'] - df['founded_at']).dt.days

# Extracting negative values only and saving in negative DataFrame
negative = df[df['difference'] < 0].copy()  # Make a copy to avoid SettingWithCopyWarning
```

```
# Count of companies that received funding before they were founded
count_companies_negative_difference = len(negative)

# Displaying the resulting DataFrame
negative
```

Out[56]:

	permalink	name	market	funding_total_usd	status	country_code	
2	/organization/rock-your-paper	'Rock' Your Paper	Publishing	40000.0	operating	EST	
20	/organization/1000memories	1000memories	Curated Web	2535000.0	acquired	USA	SF
54	/organization/140-proof	140 Proof	Advertising	5500000.0	operating	USA	SF
68	/organization/1calendar	1calendar	Education	40000.0	operating	DNK	Co
73	/organization/1daylater	1DayLater	Curated Web	43811.0	operating	NaN	
...	...	...	...	...	...	...	
49336	/organization/zova	Zova	Sports	0.0	operating	AUS	
49343	/organization/zqgame	ZQGame	Games	4220018.0	operating	USA	Lo
49365	/organization/zulily	zulily	E-Commerce	138600000.0	operating	USA	
49426	/organization/zynga	Zynga	Technology	866550786.0	operating	USA	SF
49435	/organization/zzzzapp-com	Zzzzapp Wireless ltd.	Web Development	97398.0	operating	HRV	

2691 rows × 771 columns

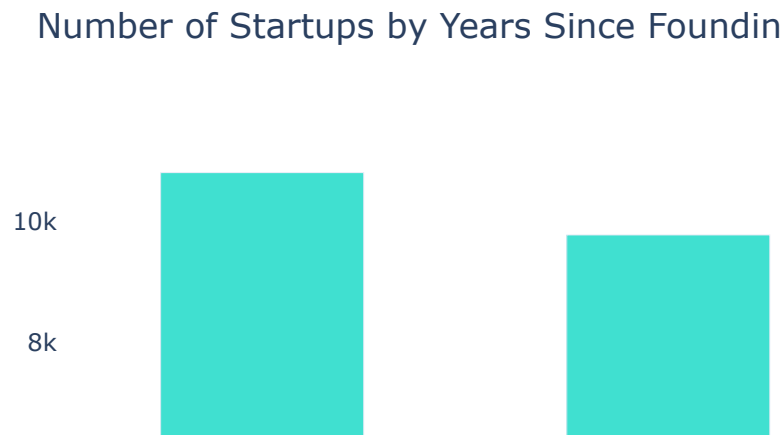
In [57]:
```
# Print count of companies that received funding before they were founded
print("Number of companies that received funding before they were founded:", count_companies_
```

```
Number of companies that received funding before they were founded: 2691
```

Q9. Number of Startups by Years Since Founding

In [58]:
```
from plotly import graph_objects as go
import pandas as pd

# Range setting for funding level
positive = df[df['difference'] > 0].copy()  # Make a copy to avoid SettingWithCopyWarning
positive['difference_level'] = 0
positive.loc[positive['difference'] < 365, 'difference_level'] = 'under 1 year'
positive.loc[(positive['difference'] >= 365) & (positive['difference'] < 1095), 'difference_l
positive.loc[(positive['difference'] >= 1095) & (positive['difference'] < 1825), 'difference_
positive.loc[(positive['difference'] >= 1825) & (positive['difference'] < 3650), 'difference_
positive.loc[(positive['difference'] >= 3650) & (positive['difference'] < 7300), 'difference_
positive.loc[positive['difference'] >= 7300, 'difference_level'] = 'over 20 years'

# Counting the number of startups for each difference level
level_counts = positive['difference_level'].value_counts()

# Plotting the values using Plotly's graph objects
fig = go.Figure(data=[go.Bar(x=level_counts.index, y=level_counts.values, marker_color='turqu
fig.update_layout(title='Number of Startups by Years Since Founding',
                  xaxis=dict(title='Years Since Founding'),
                  yaxis=dict(title='Counts'),
                  bargap=0.5,
```

```
                  plot_bgcolor='white')
fig.show()
```



In [59]:

```
# Create a DataFrame for table
table_data = pd.DataFrame({'Years Since Founding': level_counts.index,
                           'Counts': level_counts.values})

# Plotting the table using Plotly's graph objects
fig = go.Figure(data=[go.Table(header=dict(values=['Years Since Founding', 'Counts']),
                               cells=dict(values=[table_data['Years Since Founding'], table_d
                     ])
fig.update_layout(title='Number of Startups by Years Since Founding')
fig.show()
```

Number of Startups by Years Since Founding

Years Since Founding
1-3 years
under 1 year
5-10 years
3-5 years
10-20 years
over 20 years

Insights:

- A substantial number of startups received funding within their first year of founding, indicating strong early-stage investor interest.
- Additionally, there's a notable increase in funding recipients within the initial 3 years, indicating continued support for emerging ventures.
- However, there's a decline in funding recipients beyond 5 years, with a significant drop after the first decade, implying challenges in attracting investment for more mature companies. Nonetheless, a small but consistent number of startups continue to secure funding even after two decades, showcasing enduring investor confidence in select long-standing ventures.

Overall Insights:

- The software sector emerges as the frontrunner in startup funding, demonstrating a strong investment trend, with biotechnology and mobile markets closely trailing behind, indicating significant investor confidence and support in these sectors.
- Post-2020, the top five sectors driving funding and startup activity comprise biotechnology, curated web, e-commerce, mobile, and software, showcasing a notable shift in investment focus towards these industries.
- The highest funded sectors, namely Biotechnology, Mobile, and Software, have secured substantial investment amounts, with Biotechnology leading the pack, followed closely by Mobile and Software.
- The majority of startups, constituting 32,597 entities, are currently in the "operating" phase, indicating active business operations and ongoing market presence.

- A significant number of startups, totaling 2,971, have transitioned to the "acquired" status, suggesting successful exits through acquisition by larger companies or investors.
- Acquired startups have generally received higher funding amounts compared to operating startups.
- Startups acquired in various years, such as 2012, have secured substantial funding, with funding amounts typically exceeding those received by operating startups in the same year.
- Conversely, operating startups across different founding years have received comparatively lower funding amounts, indicating potential challenges in accessing significant investment capital compared to their acquired counterparts.
- The disparity in funding levels between acquired and operating startups underscores the impact of acquisition events on fundraising success and highlights the importance of understanding the funding dynamics within different startup cohorts.
- Acquired startups have the highest mean funding amounts across all funding rounds compared to operating and closed startups, indicating a greater level of investment and financial support in the acquisition process.
- Round B appears to be the most heavily funded round for all three statuses, with acquired startups receiving the highest mean funding amount, followed by operating and closed startups.
- Operating startups exhibit moderate mean funding amounts across all rounds, suggesting ongoing investment and growth opportunities despite lower funding levels compared to acquired startups.
- San Francisco, New York, and Palo Alto emerge as the top cities in the USA with the highest startup counts, indicating vibrant startup ecosystems and significant entrepreneurial activity in these regions.
- Bangalore, Mumbai, and New Delhi stand out as the top cities in India with the highest startup counts, showcasing thriving startup ecosystems and substantial entrepreneurial presence in these metropolitan areas.
- San Francisco, New York, and London emerge as the leading cities globally, showcasing vibrant entrepreneurial ecosystems and significant innovation hubs.
- The year 2012 stands out with the highest number of startups, indicating a significant influx of entrepreneurial activity during that period.
- Moreover, there is a noticeable surge in the number of new startups from 2000 to 2014, reflecting a period of pronounced growth and expansion in the startup ecosystem.
- A significant number of companies were able to secure funding within a relatively short period after their founding date, potentially reflecting investor confidence in their business models, products, or services. It suggests that these startups were able to attract investors quickly, which could be attributed to various factors such as innovative ideas, strong founding teams, market potential, or early traction.
- A smaller proportion of startups were able to secure funding after a considerably longer period since their founding date, indicating a different investment scenario or business trajectory. Startups receiving funding after more than 20 years may represent companies that have undergone significant growth, innovation, or pivots over time, leading to renewed investor interest or strategic partnerships

- A substantial number of startups received funding within their first year of founding, indicating strong early-stage investor interest
- Additionally, there's a notable increase in funding recipients within the initial 3 years, indicating continued support for emerging ventures
- However, there's a decline in funding recipients beyond 5 years, with a significant drop after the first decade, implying challenges in attracting investment for more mature companies. Nonetheless, a small but consistent number of startups continue to secure funding even after two decades, showcasing enduring investor confidence in select long-standing ventures.

Recommendation:

- Sector Diversification: While software, biotechnology, and mobile markets are receiving significant funding, it's essential for investors to diversify their portfolios across different sectors to mitigate risks and explore emerging opportunities in sectors like curated web, e-commerce, and health tech.
- Post-2020 Investment Strategy: Investors should consider adjusting their investment strategies to align with the evolving startup landscape, focusing on sectors such as biotechnology, curated web, e-commerce, mobile, and software that have demonstrated resilience and growth potential post-2020.
- Strategic Acquisitions: Companies aiming for successful exits should focus on building scalable and innovative solutions, as evidenced by the higher funding amounts received by acquired startups. Strategic partnerships and acquisitions can provide significant value to both startups and acquiring companies.
- Location-Based Investments: Investors looking to capitalize on thriving startup ecosystems should prioritize investments in cities like San Francisco, New York, Palo Alto, Bangalore, Mumbai, and New Delhi, which serve as prominent hubs for entrepreneurial activity and innovation.
- Early-stage Funding: Recognizing the importance of early-stage funding, investors should actively seek opportunities to support promising startups within their first year of founding, as these companies have demonstrated potential for rapid growth and scalability.
- Long-term Investment Opportunities: While there's a decline in funding recipients beyond 5 years, investors should remain vigilant for long-term investment opportunities, particularly in startups with enduring value propositions, strong market traction, and innovative business models.
- Support for Operating Startups: Operating startups, despite receiving comparatively lower funding amounts, represent ongoing growth opportunities. Investors should consider providing strategic support, mentorship, and funding to help these startups scale and succeed in competitive markets.

By incorporating these recommendations into their investment strategies, investors can effectively navigate the dynamic startup landscape, capitalize on emerging trends, and maximize returns on investment while fostering innovation and entrepreneurship.

** End of Project **

In [ ]: