Startup_Funding_Analysis

June 20, 2025

0.1 Startup Funding Analysis in India

By: Sagar Dhiman
Tools Used : Python, Pandas, NumPy, Seaborn, Matplotlib
Objective : Analyze startup funding trends across cities, industries, investors, and time using data analysis and visualizations.

[152]:
```
#importlibraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Ensure that graphs are displayed within the notebook
%matplotlib inline
```

[130]:
```
# read csv file using pandas

df = pd.read_csv(r'C:\Users\scpl\OneDrive\Desktop\IT\Data␣
 ↪Science\Datasets\startup_funding.csv')
```

[131]:
```
# Top 5 rows check

df.head()
```

[131]:

	Sr No	Date dd/mm/yyyy	Startup Name	Industry Vertical	SubVertical	City Location
0	1	09/01/2020	BYJU'S	E-Tech	E-learning	Bengaluru
1	2	13/01/2020	Shuttl	Transportation	App based shuttle service	Gurgaon
2	3	09/01/2020	Mamaearth	E-commerce	Retailer of baby and toddler products	Bengaluru
3	4	02/01/2020	https://www.wealthbucket.in/	FinTech	Online Investment	New Delhi
4	5	02/01/2020	Fashor	Fashion and Apparel	Embroiled Clothes For Women	Mumbai

```
                Investors Name       InvestmentnType Amount in USD Remarks
0    Tiger Global Management  Private Equity Round  20,00,00,000     NaN
1  Susquehanna Growth Equity              Series C     80,48,394     NaN
2      Sequoia Capital India              Series B   1,83,58,860     NaN
3             Vinod Khatumal          Pre-series A     30,00,000     NaN
4    Sprout Venture Partners            Seed Round     18,00,000     NaN
```

[132]:
```
# Data Basic Info Check

print("Shape of data:", df.shape)
```

```
Shape of data: (3044, 10)
```

[133]:
```
print("Columns list:\n", df.columns)
```

```
Columns list:
 Index(['Sr No', 'Date dd/mm/yyyy', 'Startup Name', 'Industry Vertical',
       'SubVertical', 'City  Location', 'Investors Name', 'InvestmentnType',
       'Amount in USD', 'Remarks'],
      dtype='object')
```

[134]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3044 entries, 0 to 3043
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Sr No              3044 non-null   int64 
 1   Date dd/mm/yyyy    3044 non-null   object
 2   Startup Name       3044 non-null   object
 3   Industry Vertical  2873 non-null   object
 4   SubVertical        2108 non-null   object
 5   City  Location     2864 non-null   object
 6   Investors Name     3020 non-null   object
 7   InvestmentnType    3040 non-null   object
 8   Amount in USD      2084 non-null   object
 9   Remarks            419 non-null    object
dtypes: int64(1), object(9)
memory usage: 237.9+ KB
```

[135]:
```
# Column Names Clean

print("Old Columns:\n", df.columns)
```

```
Old Columns:
 Index(['Sr No', 'Date dd/mm/yyyy', 'Startup Name', 'Industry Vertical',
       'SubVertical', 'City  Location', 'Investors Name', 'InvestmentnType',
```

```
       'Amount in USD', 'Remarks'],
      dtype='object')
```

[136]:
```
df.rename(columns={
    'ï»¿Sr No': 'Sr No',
    'Date dd/mm/yyyy': 'Date',
    'Startup Name': 'Startup',
    'Industry Vertical': 'Industry',
    'SubVertical': 'SubIndustry',
    'City  Location': 'City',
    'Investors Name': 'Investors',
    'InvestmentnType': 'Investment Type',
    'Amount in USD': 'Amount',
    'Remarks': 'Remarks'
}, inplace=True)

# New column names check
print("New Columns:\n", df.columns)
```

```
New Columns:
 Index(['Sr No', 'Date', 'Startup', 'Industry', 'SubIndustry', 'City',
       'Investors', 'Investment Type', 'Amount', 'Remarks'],
      dtype='object')
```

[137]:
```
# Duplicate Rows Check and Remove

print("Duplicate rows:", df.duplicated().sum())
df.drop_duplicates(inplace=True)
```

```
Duplicate rows: 0
```

[138]:
```
#  Check Missing Values in Each Column

df.isnull().sum()
```

[138]:
```
Sr No                0
Date                 0
Startup              0
Industry           171
SubIndustry        936
City               180
Investors           24
Investment Type      4
Amount             960
Remarks           2625
dtype: int64
```

[139]:
```
# Fill missing values
df['City'] = df['City'].fillna("Unknown")
df['Investors'] = df['Investors'].fillna("Undisclosed")

# Drop rows with null Investment Type
df = df[df['Investment Type'].notnull()]

# 3. Drop 'Remarks' column only if it exists
if 'Remarks' in df.columns:
    df.drop('Remarks', axis=1, inplace=True)
```

[140]:
```
df['Industry'] = df['Industry'].fillna("Unknown")
df['SubIndustry'] = df['SubIndustry'].fillna("Not Specified")
# Keep Amount as is - NaN
```

[141]:
```
# 4. Final check
df.isnull().sum()
```

[141]:
```
Sr No              0
Date               0
Startup            0
Industry           0
SubIndustry        0
City               0
Investors          0
Investment Type    0
Amount           959
dtype: int64
```

0.1.1 Group by Startup & Sum Amount

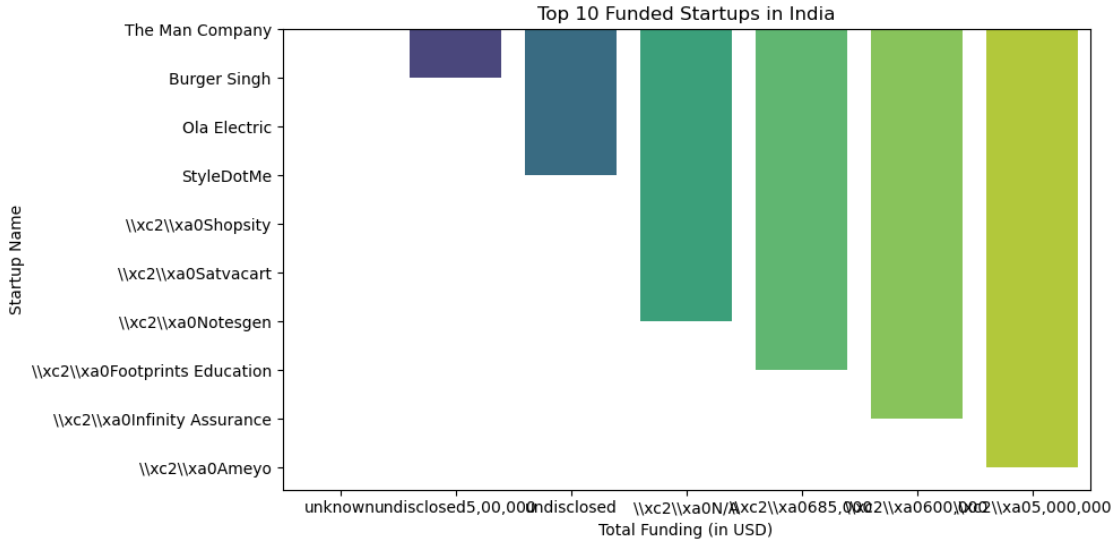
[151]:
```
# Filter the dataset to include only rows with valid 'Amount' values
df_amount = df[df['Amount'].notnull()]

# Startup-wise total funding calculate
top_startups = df_amount.groupby('Startup')['Amount'].sum().
 ↪sort_values(ascending=False).head(10)

# top 10
print(top_startups)
```

```
Startup
The Man Company                          unknown
Burger Singh                 undisclosed5,00,000
Ola Electric                         undisclosed
StyleDotMe                           undisclosed
\\xc2\\xa0Shopsity                  \\xc2\\xa0N/A
\\xc2\\xa0Satvacart                 \\xc2\\xa0N/A
```

```
\\xc2\\xa0Notesgen                        \\xc2\\xa0N/A
\\xc2\\xa0Footprints Education      \\xc2\\xa0685,000
\\xc2\\xa0Infinity Assurance        \\xc2\\xa0600,000
\\xc2\\xa0Ameyo                    \\xc2\\xa05,000,000
Name: Amount, dtype: object
```

[143]:
```
# Bar Plot - Top 10 Funded Startups
```

[144]:
```
# Visualization

plt.figure(figsize=(10,5))
sns.barplot(x=top_startups.values, y=top_startups.index, hue=top_startups.
 ↪index, dodge=False, legend=False, palette="viridis")
plt.title("Top 10 Funded Startups in India")
plt.xlabel("Total Funding (in USD)")
plt.ylabel("Startup Name")
plt.tight_layout()
plt.show()
```



0.1.2 Clean Investors Column

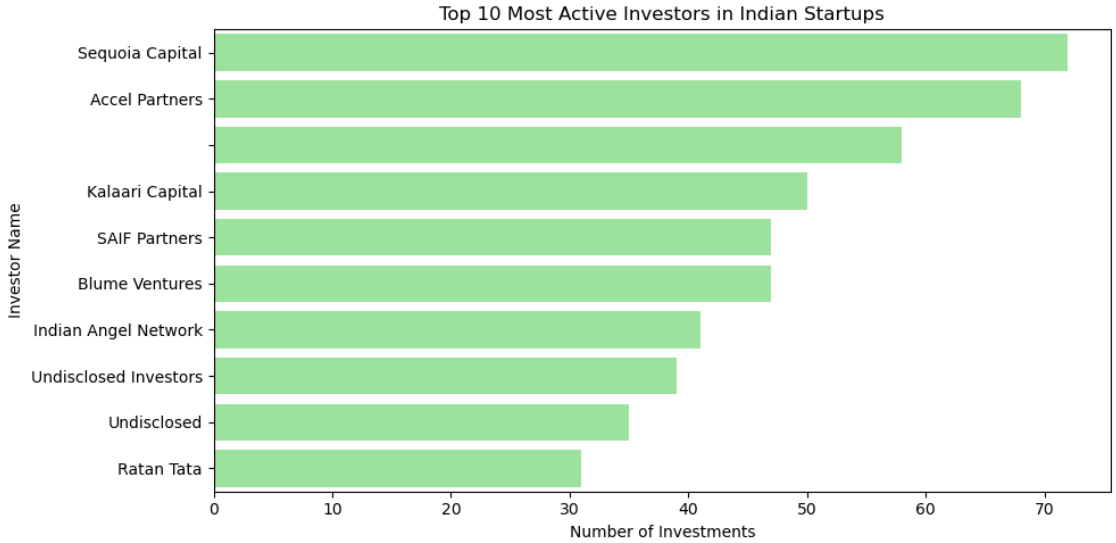
[150]:
```
# Split the 'Investors' column into lists where multiple investors are mentioned
investors_list = df['Investors'].dropna().str.split(',')

# Flatten the list (all investors in ek hi list)
all_investors = []
for sublist in investors_list:
    for investor in sublist:
        investor = investor.strip()
```

```
        all_investors.append(investor)
```

[148]:
```
# Import Counter to count how many times each investor appears
from collections import Counter

investor_count = Counter(all_investors)

# Top 10 most active investors
top_investors = investor_count.most_common(10)
```

[149]:
```
# Bar Plot - Top 10 Investors
```

[147]:
```
# Separate the data into two lists for plotting: investor names and their␣
 ↪corresponding investment counts
investor_names = [x[0] for x in top_investors]
investment_counts = [x[1] for x in top_investors]

# Plot
plt.figure(figsize=(10,5))
sns.barplot(x=investment_counts, y=investor_names, color='lightgreen')
plt.title("Top 10 Most Active Investors in Indian Startups")
plt.xlabel("Number of Investments")
plt.ylabel("Investor Name")
plt.tight_layout()
plt.show()
```



City-wise Number of Funded Startups

[84]:
```
df_amount = df[df['Amount'].notnull()]
```

[85]:
```
# Count funded startups per city
city_startup_count = df_amount['City'].value_counts().head(10)
```

[86]:
```
# Plot chart
plt.figure(figsize=(10,5))
sns.barplot(x=city_startup_count.values, y=city_startup_count.index,␣
 ↪color='skyblue')
```

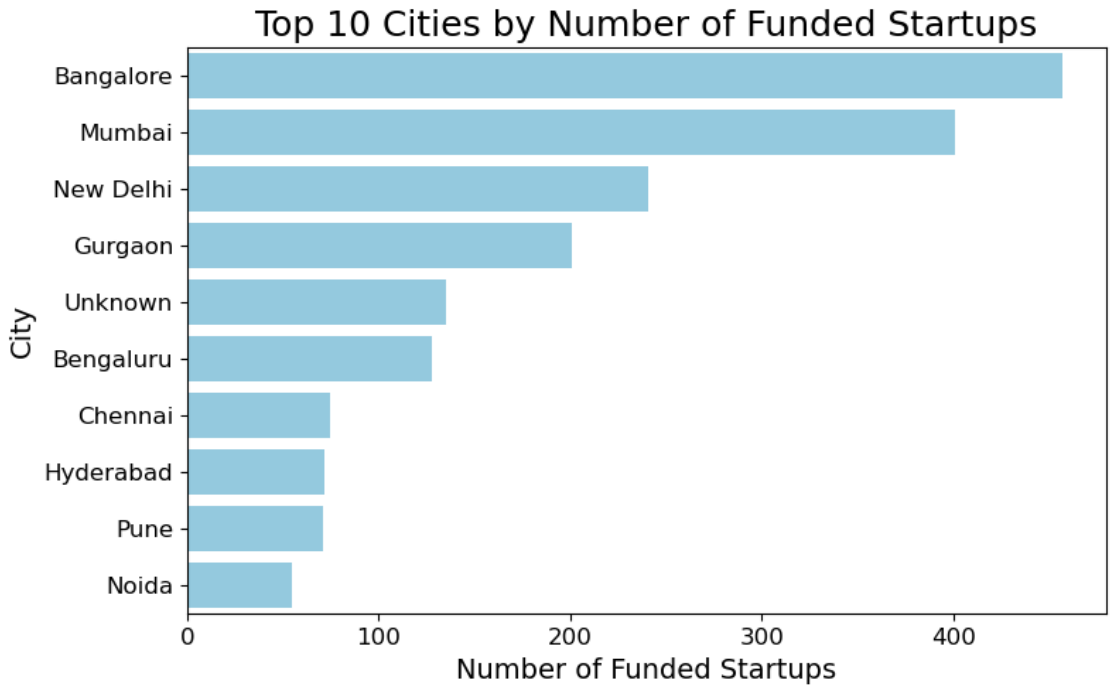
[86]:
```
<Axes: ylabel='City'>
```

[87]:
```
# Add titles & labels
plt.title("Top 10 Cities by Number of Funded Startups", fontsize=18)
plt.xlabel("Number of Funded Startups", fontsize=14)
plt.ylabel("City", fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
```

[87]:
```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
 [Text(0, 0, 'Bangalore'),
  Text(0, 1, 'Mumbai'),
  Text(0, 2, 'New Delhi'),
  Text(0, 3, 'Gurgaon'),
  Text(0, 4, 'Unknown'),
  Text(0, 5, 'Bengaluru'),
  Text(0, 6, 'Chennai'),
  Text(0, 7, 'Hyderabad'),
  Text(0, 8, 'Pune'),
  Text(0, 9, 'Noida')])
```

[89]:
```
plt.subplots_adjust(left=0.3, right=0.95, top=0.9, bottom=0.1)
plt.show()
```

```
<Figure size 640x480 with 0 Axes>
```



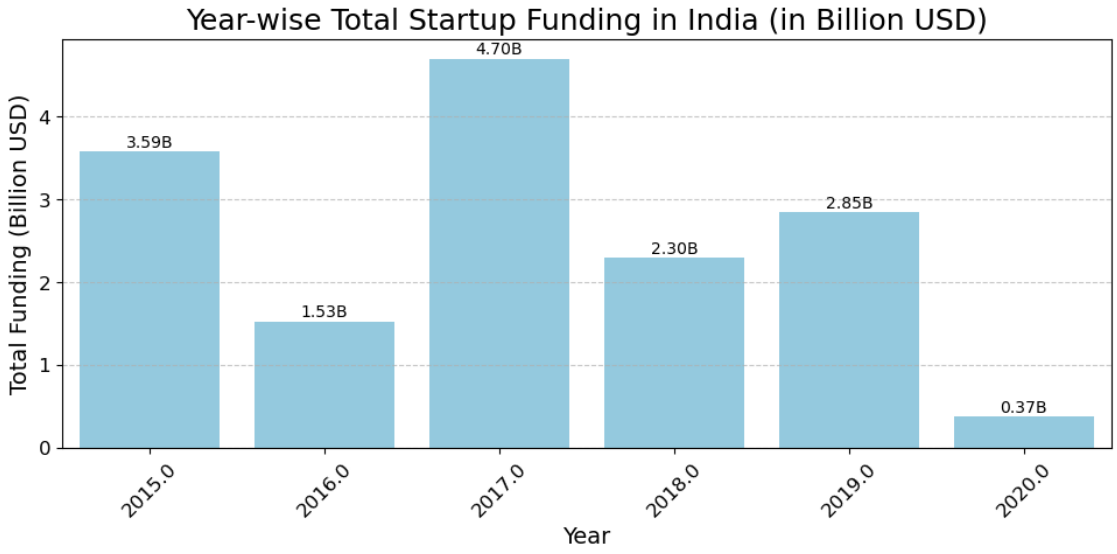
0.1.3 Year-wise Funding Analysis

[114]:
```
# Remove commas (like '2,00,000' → '200000')
df['Amount'] = df['Amount'].str.replace(',', '', regex=False)

# Convert to numeric
df['Amount'] = pd.to_numeric(df['Amount'], errors='coerce')
```

[119]:
```
# Filter valid rows
df_year = df[(df['Amount'].notnull()) & (df['Year'].notnull())]

# Group and sum
yearly_funding = df_year.groupby('Year')['Amount'].sum().sort_index()

# Convert to billion USD
yearly_funding_billion = yearly_funding / 1e9
```

[118]:
```
# Plot
plt.figure(figsize=(10,5))
sns.barplot(x=yearly_funding_billion.index, y=yearly_funding_billion.values,␣
 ↪color='skyblue')

# Annotate bars
for index, value in enumerate(yearly_funding_billion.values):
```

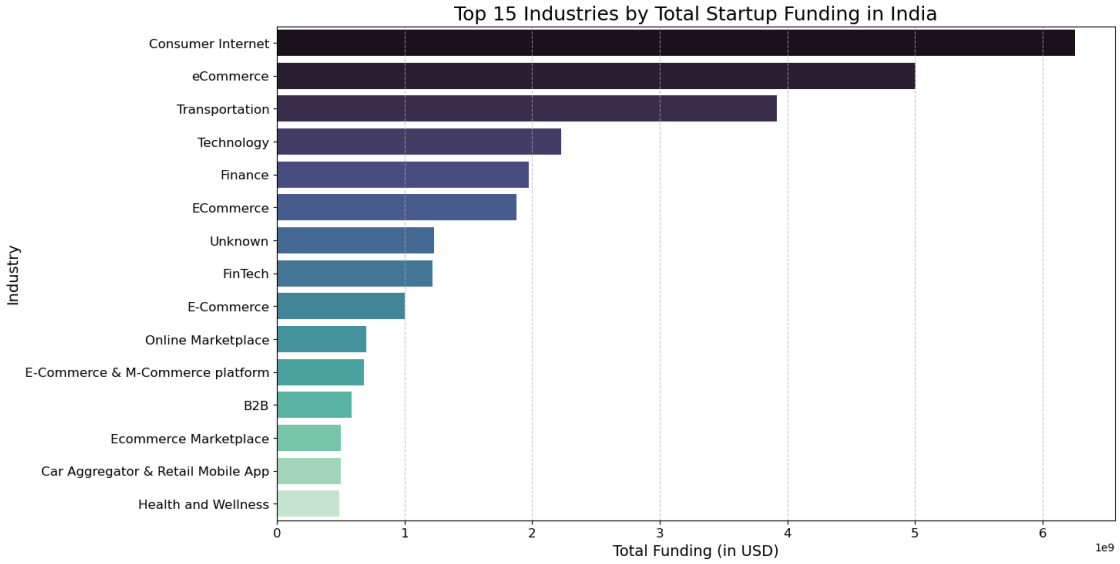
```
    plt.text(index, value + 0.05, f'{value:.2f}B', ha='center', fontsize=10)

# Title and styling
plt.title("Year-wise Total Startup Funding in India (in Billion USD)",␣
 ↪fontsize=18)
plt.xlabel("Year", fontsize=14)
plt.ylabel("Total Funding (Billion USD)", fontsize=14)
plt.xticks(rotation=45, fontsize=12)
plt.yticks(fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.subplots_adjust(left=0.08, right=0.98, top=0.9, bottom=0.2)

plt.show()
```



0.1.4 Industry-wise Total Funding Analysis

[122]:
```
# Remove missing industries
df_industry = df[df['Industry'].notnull() & df['Amount'].notnull()]

# Group by Industry and sum Amount
industry_funding = df_industry.groupby('Industry')['Amount'].sum().
 ↪sort_values(ascending=False).head(15)

# Create DataFrame for hue compatibility
industry_funding_df = industry_funding.reset_index()
industry_funding_df.columns = ['Industry', 'TotalFunding']
```

[124]:
```
# Plot
plt.figure(figsize=(16,8))
sns.barplot(data=industry_funding_df, y='Industry', x='TotalFunding',␣
 ↪hue='Industry', palette='mako', legend=False)

# Add labels
plt.title("Top 15 Industries by Total Startup Funding in India", fontsize=18)
plt.xlabel("Total Funding (in USD)", fontsize=14)
plt.ylabel("Industry", fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.subplots_adjust(left=0.3, right=0.98, top=0.9, bottom=0.1)

plt.show()
```



[125]:
```
# Startup Funding Category (Custom Column)
```

[126]:
```
# Create category column based on Amount
def funding_category(amount):
    if amount >= 100000000:
        return 'High'
    elif amount >= 10000000:
        return 'Medium'
    else:
        return 'Low'

df['Funding Category'] = df['Amount'].apply(funding_category)
```

```
# Count how many startups in each category
df['Funding Category'].value_counts()
```

[126]:
```
Funding Category
Low       2553
Medium     415
High        72
Name: count, dtype: int64
```

[127]:
```
# NumPy Stats - Mean, Median, STD of Funding

import numpy as np

amount_clean = df['Amount'].dropna()
mean_funding = np.mean(amount_clean)
median_funding = np.median(amount_clean)
std_funding = np.std(amount_clean)

print("Mean:", mean_funding)
print("Median:", median_funding)
print("Standard Deviation:", std_funding)
```

```
Mean: 18356017.872938894
Median: 1700000.0
Standard Deviation: 121365561.52151532
```

[ ]:

Conclusion :

This project provides valuable insights into India's startup ecosystem, highlighting key funding trends, top industries, cities, and investors using Python, Pandas, and data visualization.

[ ]:

[ ]: