PROJECT : CASESTUDY OF STARTUP FUNDING DATASET

INTRODUCTION

- This dataset has funding information of the Indian startups from January 2015 to August 2017.
- In this DATASET,it conatins the details of all the Investors and their Investments in different STARTUPS.
- All the columns are mentioned below

1. Feature Details :
2. SNo - Serial number
3. Date - Date of funding in format DD/MM/YYYY.
4. StartupName - Name of the startup which got funded.
5. IndustryVertical - Industry to which the startup belongs.
6. SubVertical - Sub-category of the industry type.
7. CityLocation - City which the startup is based out of.
8. InvestorsName - Name of the investors involved in the funding round.
9. InvestmentType - Either Private Equity or Seed Funding.
10. AmountInUSD - Funding Amount in USD.
11. Remarks - Other information, if any.

OBJECTIVES

- In this CASE-STUDY we are going to solve some Problems where if one person has a product to Launch in the market, and to establish a Product startup in INDIA.
- By using the Given DataSET startup_funding.csv(https://drive.google.com/file/d/1UaWCHLBZpxCmTSrmQQ5RXM5OMay17z8V/view) we are going to Solve problems according to the Product Owner requirements.
- Insights -
  - Find out what type of startups are getting funded in the last few years?
  - Who are the important investors?
  - What are the hot fields that get a lot of funding these days?

- Libraries used in this CASE STUDY
- PANDAS - Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool,built on top of the Python programming language.
- Matplotlib is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc.

PROBLEM 1:

Your Friend has developed the Product and he wants to establish the product startup and he is searching for a perfect location where getting the investment has a high chance. But due to its financial restriction, he can choose only between three locations - Bangalore, Mumbai, and NCR. As a friend, you want to help your friend deciding the location. NCR include Gurgaon, Noida and New Delhi. Find the location where the most number of funding is done.

OBJECTIVES

- Find the location where the most number of funding is done.
- Use appropriate DATA VISUALIZATION TECHNIQUE to represent the data

DATA CLEANING

- According to the given Problem we have to find out the Top Cities which has maximum number of fundings.
- To solve this Problem i have used libraries like csv,Pandas for extracting the Data and Cleaning the Data,matplotlib for Data Visualization

- At First remove all the NaN Values from required columns
  - df.dropna(subset=["CityLocation","AmountInUSD"],inplace=True)
- Check all the unique values of "CityLocation" column so that no errors in the CASE-SENTIVITY.If any CASE-SENTIVITY errors then replace them with accurate or Required name.
- for example
  - df.CityLocation.replace("bangalore","Bangalore",inplace=True)
  - df.CityLocation.replace("Delhi","New Delhi",inplace=True)

In [85]:

```
import pandas as pd
import matplotlib.pyplot as plt
import collections
import csv
df=pd.read_csv("Datasets/startup_funding.csv")

#Data Cleaning

df.dropna(subset=["CityLocation","AmountInUSD"],inplace=True) #Dropping all NaN values from the "CityLocation" and "AmountInUSD" Columns
df.CityLocation.replace("bangalore","Bangalore",inplace=True) #Checking CASE-SENSTIVITY and correcting, to have Unique values in the data set.
df.CityLocation.replace("Delhi","New Delhi",inplace=True) #Checking CASE-SENSTIVITY and correcting, to have Unique values in the data set.
df.CityLocation.value_counts()
```

Out[85]:

```
Bangalore                405
Mumbai                   300
New Delhi                213
Gurgaon                  165
Pune                      54
Hyderabad                 53
Chennai                   47
Noida                     45
Ahmedabad                 25
Jaipur                    10
Kolkata                    8
Vadodara                   5
Chandigarh                 5
Pune / US                  4
Goa                        4
Indore                     3
Singapore                  3
Kanpur                     2
New Delhi / US             2
Bhopal                     2
Bangalore / SFO            2
Bangalore/ Bangkok         2
Coimbatore                 2
Gwalior                    2
Bangalore / USA            1
Gurgaon / SFO              1
US/India                   1
New York/ India            1
Mumbai / Global            1
Bangalore / San Mateo      1
Mumbai / NY                1
India / US                 1
Dallas / Hyderabad         1
Belgaum                    1
Noida / Singapore          1
USA/India                  1
Pune/Seattle               1
Bangalore / Palo Alto      1
Boston                     1
Hyderabad/USA              1
Jodhpur                    1
Mumbai / UK                1
Varanasi                   1
Pune / Dubai               1
SFO / Bangalore            1
Trivandrum                 1
Lucknow                    1
Kerala                     1
Panaji                     1
Missourie                  1
Udupi                      1
Name: CityLocation, dtype: int64
```

- Split all the names which are having multiple names in single column("CityLocation") and append in a empty list city_name
- Use Dictionary for City name as KEY and their count as VALUES.
- Importing the library collections sort the Dictionary in descending order in order to get Top cities with respect to their count.
- Take to empty LISTS to append the city names and their count of number of fundings respectivelty

In [86]:

```
city_name=[] #An Empty LIST to append the required CITIES
df=df.CityLocation
for names in df.values:
    split_names=names.split("/") #splitting the cities which have more than one.For Example:Mumbai / Global.Which will converted to list[Mumbai , Global]
    for name in split_names:
        stripped_name=name.strip() # To remove the LEADING SPACES and TRAILING SPACES from the List
        if ((stripped_name=="New Delhi") | (stripped_name=="Bangalore") | (stripped_name=="Mumbai") | (stripped_name=="Gurgaon") | (stripped_name=="Noida")):
            city_name.append(stripped_name) #Append the required CITIES into the city_name list
```

In [87]:

```
dic={} #Dictionary to count the Number of fundings for the CITIES
for ele in city_name:
    dic[ele] = dic.get(ele,0) + 1
bar_city=[] #Empty list to append the CITIES
bar_city_values=[]#Empty list to append the NUMBER OF FUNDINGS
ord_dic=sorted(dic,key=dic.get,reverse=True) #Sorting the dictionary with respect to the Values in descending order
print("CITY  ","Number Of Fundings")
for i in range(len(ord_dic)):
    print(ord_dic[i]," ",dic[ord_dic[i]]) #Printing the Cities corresponding to their values
    bar_city.append(ord_dic[i]) # Appending the CITIES
    bar_city_values.append(dic[ord_dic[i]]) # Appending the values of NUMBER OF FUNDINGS
```

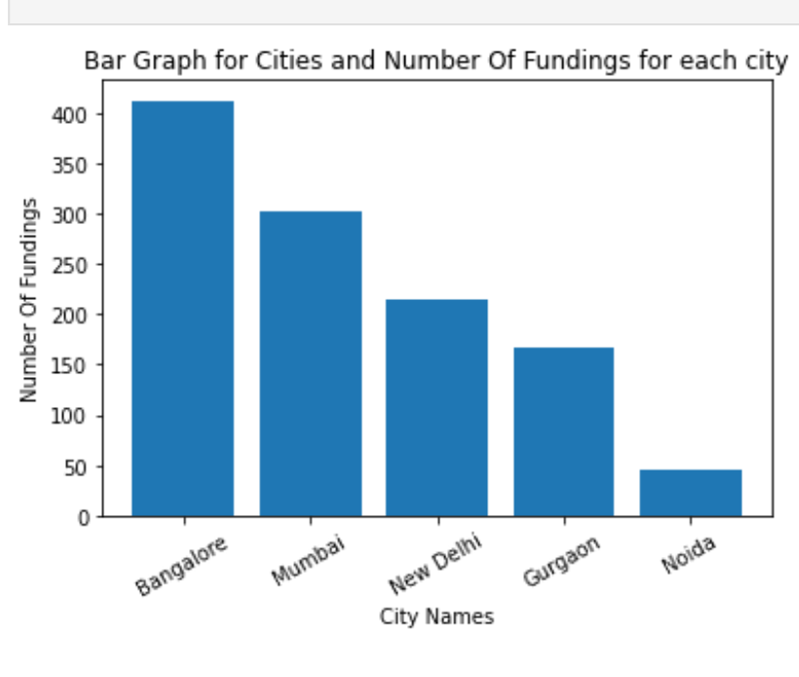
```
CITY   Number Of Fundings
Bangalore   413
Mumbai   303
New Delhi   215
Gurgaon   166
Noida   46
```

DATA VISUALIZATION:

- For DATA VISUALIZATION we import matplotlib library
- The BAR GRAPH is the appropriate DATA VISUALIZATION to represent the TOP cities with their Count of Number of Fundings

In [88]:

```
plt.bar(bar_city,bar_city_values) #plotting the BAR GRAPH with respect to the "bar_city,bar_city_values".
plt.xlabel("City Names")
plt.ylabel("Number Of Fundings")
plt.title("Bar Graph for Cities and Number Of Fundings for each city")
plt.xticks(rotation=30)
plt.show()
```



PROBLEM 2:

Even after trying for so many times, your friend’s startup could not find the investment. So you decided to take this matter in your hand and try to find the list of investors who probably can invest in your friend’s startup. Your list will increase the chance of your friend startup getting some initial investment by contacting these investors. Find the top 5 investors who have invested maximum number of times (consider repeat investments in one company also). In a startup, multiple investors might have invested. So consider each investor for that startup. Ignore undisclosed investors.

OBJECTIVES:

- Find the list of investors.
- Find the top 5 investors who have invested maximum number of times (consider repeat investments in one company also)

DATA CLEANING

- At First remove all the NaN Values from required columns
  - df.dropna(subset=["InvestorsName",],inplace=True)
- crate a DATAFRAME by ignoring all the "Undisclosed Investors" from the data
  - df=df[df.InvestorsName!="Undisclosed Investors"]

In [89]:

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

df=pd.read_csv("Datasets/startup_funding.csv")
df.dropna(subset=["InvestorsName"],inplace=True)
investor_names=[]
df=df[df.InvestorsName!="Undisclosed Investors"]
df=df[df.InvestorsName!="Undisclosed investors"]
df=df[df.InvestorsName!="undisclosed investors"]
df=df.InvestorsName
```

- Split all the names which are having multiple names in single column("InvestorsName") and append in a empty list city_name and use strip() for removing leading and trailing spaces

In [90]:

```
for names in df.values:
    split_names=names.split(",")
    for each_name in split_names:
        if each_name!="":
            stripped_names=each_name.strip()
            investor_names.append(stripped_names)
```

- Use Dictionary for City name as KEY and their count as VALUES.
- Sort the dictonary with respect to their Values
- Take to empty LISTS to append the investor names and their count of number of fundings respectivelty

In [91]:

```
dic={}
for ele in investor_names:
    dic[ele]=dic.get(ele,0)+1
ord_dic=sorted(dic,key=dic.get,reverse=True)
investor_name=[]
investor_count=[]
for i in range(0,5):
    print(ord_dic[i],dic[ord_dic[i]])
    investor_name.append(ord_dic[i])
    investor_count.append(dic[ord_dic[i]])
```

```
Sequoia Capital 64
Accel Partners 53
Kalaari Capital 44
SAIF Partners 41
Indian Angel Network 40
```

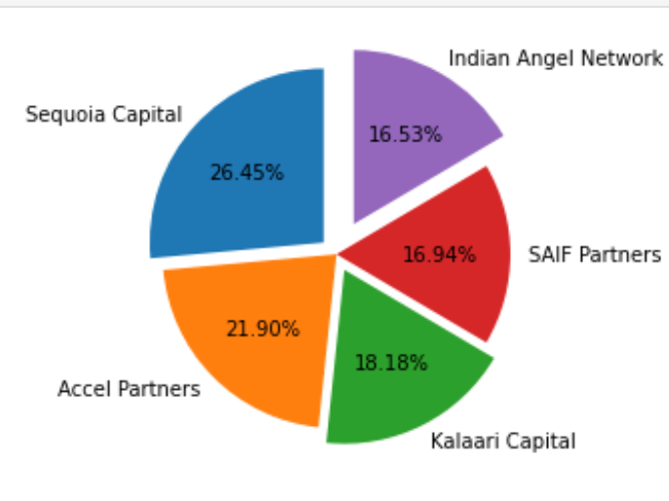
DATA VISULAIZATION

- For DATA VISUALIZATION we import matplotlib library
- The PIE CHART is the appropriate DATA VISUALIZATION to represent the TOP cities with their Count of Number of Fundings
- Using of PIE CHART can be easily represent the percentages of each individual investores

In [92]:

```
explode=[0.1,0,0.1,0,0.2]

plt.pie(investor_count,labels=investor_name,autopct="%.2f%%",startangle=90,explode=explode)
plt.axis("equal")
plt.show()
```



PROBLEM 3

After re-analysing the dataset you found out that some investors have invested in the same startup at different number of funding rounds. So before finalising the previous list, you want to improvise it by finding the top 5 investors who have invested in different number of startups. This list will be more helpful than your previous list in finding the investment for your friend startup. Find the top 5 investors who have invested maximum number of times in different companies. That means, if one investor has invested multiple times in one startup, count one for that company. There are many errors in startup names. Ignore correcting all, just handle the important ones - Ola, Flipkart, Oyo and Paytm.

OBJECTIVES

- Find the top 5 investors who have invested maximum number of times in different companies.
- If one investor has invested multiple times in one startup, count one for that company

DATA CLEANING

- There are many errors in startup names correct the names of Ola, Flipkart, Oyo and Paytm
  - df.StartupName.replace("Ola Cabs","Ola",inplace=True)
- At First remove all the NaN Values from required columns
  - df.dropna(subset=["InvestorsName","StartupName",],inplace=True)
- crate a DATAFRAME by ignoring all the "Undisclosed Investors" from the data
  - df=df[df.InvestorsName!="Undisclosed Investors"]

In [93]:

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

df=pd.read_csv("Datasets/startup_funding.csv")
df.dropna(subset=["StartupName","InvestorsName"],inplace=True)
df.StartupName.replace("Ola Cabs","Ola",inplace=True)
df.StartupName.replace("Olacabs","Ola",inplace=True)
df.StartupName.replace("Filpkart.com","Flipkart",inplace=True)
df.StartupName.replace("Flipkart.com","Flipkart",inplace=True)
df.StartupName.replace("Oyo Rooms","Oyo",inplace=True)
df.StartupName.replace("OYO Rooms","Oyo",inplace=True)
df.StartupName.replace("Oyorooms","Oyo",inplace=True)
df.StartupName.replace("OyoRooms","Oyo",inplace=True)
df.StartupName.replace("Paytm Marketplace","Paytm",inplace=True)
df=df[df.InvestorsName!="Undisclosed investors"]
df=df[df.InvestorsName!="undisclosed investors"]
df=df[df.InvestorsName!="Undisclosed Investors"]
df=df[df.InvestorsName!="undisclosed investor"]
```

- Take two empty (startup_name and investor_name) lists to append the Startupnames and InvestorsName respectively.

In [94]:

```
startup_name=[]
investor_name=[]

for index,row in df.iterrows():
    s_name=row["StartupName"]
    i_name=row["InvestorsName"]
    i_name=i_name.split(",")
    for names in i_name:
        if names!="":
            investor_name.append(names.strip())
            startup_name.append(s_name)
```

- Now create a panadas DATAFRAME for startup_name,investor_name with the columns names StartupName and InvestorsName respectively.
- Group the columns InvestorsName and StartupName
- use nunique() to return all the unique Investors name with respect to columns
- Sort the DATAFRAME in descending order with respect to values

In [95]:

```
df=pd.DataFrame({"StartupName":startup_name,"InvestorsName":investor_name})
df=df.groupby("InvestorsName")["StartupName"].nunique()
df=df.sort_values(ascending=False)[0:5]
df
```

Out[95]:

```
InvestorsName
Sequoia Capital         48
Accel Partners          47
Kalaari Capital         41
Indian Angel Network    40
Blume Ventures          36
Name: StartupName, dtype: int64
```

- Take to empty LISTS to append the investor names and their count of number of fundings respectivelty

In [96]:

```
investor=[]
investor_count=[]
for i in range(0,5):
    print(df.index[i],df.values[i])
    investor.append(df.index[i])
    investor_count.append(df.values[i])
```

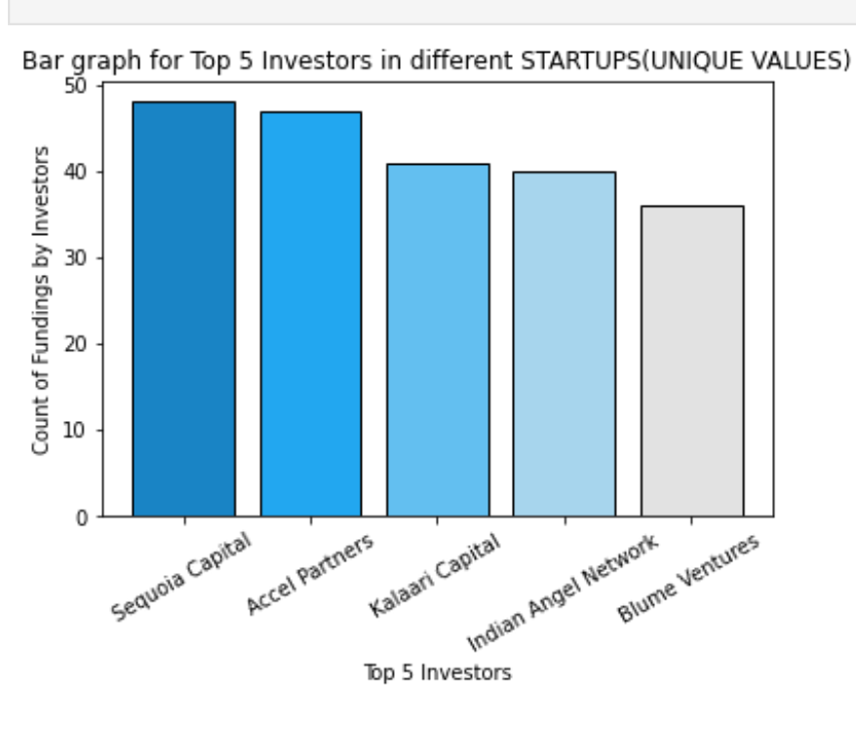
```
Sequoia Capital 48
Accel Partners 47
Kalaari Capital 41
Indian Angel Network 40
Blume Ventures 36
```

DATA VISULAIZATION

- For DATA VISUALIZATION we import matplotlib library
- The BAR GRAPH is the appropriate DATA VISUALIZATION to represent the TOP Investors with their Count of Number of Fundings

In [97]:

```
plt.title("Bar graph for Top Investors")
plt.xlabel("")
plt.xticks(rotation=30)
plt.bar(investor,investor_count,color=["#1984c5", "#22a7f0", "#63bff0", "#a7d5ed", "#e2e2e2"],edgecolor='black')
plt.title("Bar graph for Top 5 Investors in different STARTUPS(UNIQUE VALUES)")
plt.xlabel("Top 5 Investors")
plt.ylabel("Count of Fundings by Investors")
plt.xticks(rotation=30)
plt.show()
```



PROBLEM 4

Even after putting so much effort in finding the probable investors, it didn't turn out to be helpful for your friend. So you went to your investor friend to understand the situation better and your investor friend explained to you about the different Investment Types and their features. This new information will be helpful in finding the right investor. Since your friend startup is at an early stage startup, the best-suited investment type would be - Seed Funding and Crowdfunding. Find the top 5 investors who have invested in a different number of startups and their investment type is Crowdfunding or Seed Funding. Correct spelling of investment types are - "Private Equity", "Seed Funding", "Debt Funding", and "Crowd Funding". Keep an eye for any spelling mistake. You can find this by printing unique values from this column. There are many errors in startup names. Ignore correcting all, just handle the important ones - Ola, Flipkart, Oyo and Paytm.

OBJECTIVES

- Find the top 5 investors who have invested in a different number of startups and their investment type is Crowdfunding or Seed Funding

DATA CLEANING

- Correct spelling of investment types are - "Private Equity", "Seed Funding", "Debt Funding", and "Crowd Funding"
  - df.InvestmentType.replace("Crowd funding","Crowd Funding",inplace=True)
  - df.InvestmentType.replace("PrivateEquity","Private Equity",inplace=True)
  - df.InvestmentType.replace("SeedFunding","Seed Funding",inplace=True)>
- There are many errors in startup names correct the names of Ola, Flipkart, Oyo and Paytm
  - df.StartupName.replace("Ola Cabs","Ola",inplace=True)
- At First remove all the NaN Values from required columns
  - df.dropna(subset=["InvestorsName","StartupName",],inplace=True)
- crate a DATAFRAME by ignoring all the "Undisclosed Investors" from the data
  - df=df[df.InvestorsName!="Undisclosed Investors"]

In [98]:

```
import pandas as pd
import matplotlib.pyplot as plt
# import numpy as np
# import collections
# import csv

df=pd.read_csv("Datasets/startup_funding.csv")
df.dropna(subset=["StartupName","InvestorsName"],inplace=True)
df.StartupName.replace("Ola Cabs","Ola",inplace=True)
df.StartupName.replace("Olacabs","Ola",inplace=True)
df.StartupName.replace("Filpkart.com","Flipkart",inplace=True)
df.StartupName.replace("Flipkart.com","Flipkart",inplace=True)
df.StartupName.replace("Oyo Rooms","Oyo",inplace=True)
df.StartupName.replace("OYO Rooms","Oyo",inplace=True)
df.StartupName.replace("Oyorooms","Oyo",inplace=True)
df.StartupName.replace("OyoRooms","Oyo",inplace=True)
df.StartupName.replace("Paytm Marketplace","Paytm",inplace=True)
df.InvestmentType.replace("Crowd funding","Crowd Funding",inplace=True)
df.InvestmentType.replace("PrivateEquity","Private Equity",inplace=True)
df.InvestmentType.replace("SeedFunding","Seed Funding",inplace=True)

df=df[df.InvestorsName!="Undisclosed investors"]
df=df[df.InvestorsName!="undisclosed investors"]
df=df[df.InvestorsName!="Undisclosed Investors"]
df=df[df.InvestorsName!="undisclosed investor"]
```

- Retreive the InvestmentType Column containing the Seed Funding and Crowd Funding

In [99]:

```
df=df[(df.InvestmentType=="Seed Funding") | (df.InvestmentType=="Crowd Funding")]
```

- Now create a panadas DATAFRAME for startup_name,investor_name with the columns names StartupName and InvestorsName respectively.
- Group the columns InvestorsName and StartupName
- use nunique() to return all the unique Investors name with respect to columns
- Sort the DATAFRAME in descending order with respect to values

In [100…

```
startup_name=[]
investor_name=[]
for index,row in df.iterrows():
    s_name=row["StartupName"]
    i_name=row["InvestorsName"]

    i_name=i_name.split(",")
    for names in i_name:
        if names!="":
            investor_name.append(names.strip())
            startup_name.append(s_name)
df=pd.DataFrame({"StartupName":startup_name,"InvestorsName":investor_name})
df=df.groupby("InvestorsName")["StartupName"].nunique()
df=df.sort_values(ascending=False)[0:5]
df
```

Out[100…

```
InvestorsName
Indian Angel Network        33
Rajan Anandan               23
LetsVenture                 16
Anupam Mittal               16
Group of Angel Investors    14
Name: StartupName, dtype: int64
```

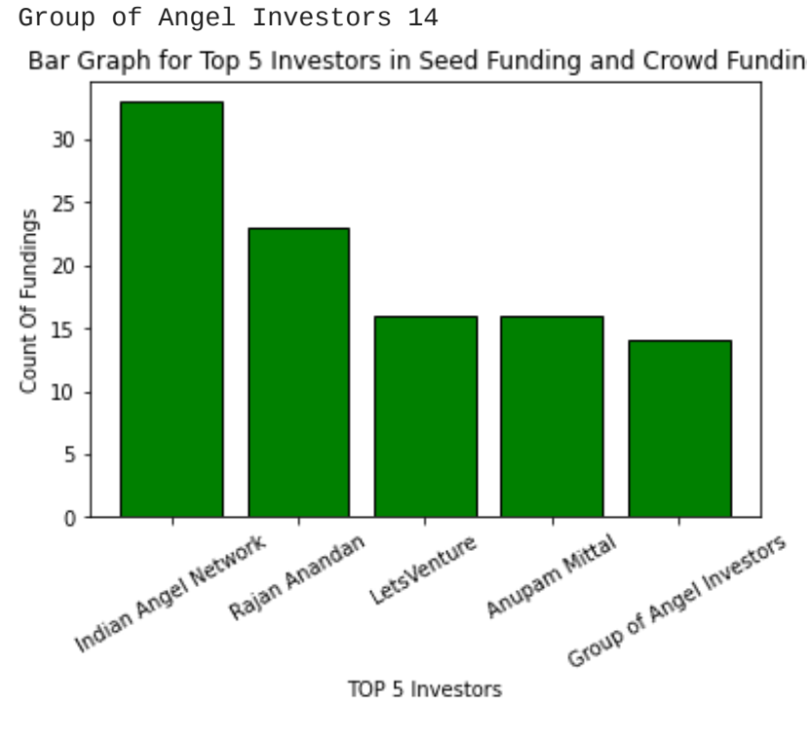
DATA VISUALIZATION

- For DATA VISUALIZATION we import matplotlib library
- The BAR GRAPH is the appropriate DATA VISUALIZATION to represent the TOP Investors with their Count of Number of Fundings

In [101…

```
investor=[]
investor_count=[]
for i in range(0,5):
    print(df.index[i],df.values[i])
    investor.append(df.index[i])
    investor_count.append(df.values[i])
labels=investor
plt.bar(investor,investor_count,color="green",edgecolor='black')
plt.xticks(rotation=30)
plt.title("Bar Graph for Top 5 Investors in Seed Funding and Crowd Funding")
plt.xlabel("TOP 5 Investors")
plt.ylabel("Count Of Fundings")
plt.show()
```

```
Indian Angel Network 33
Rajan Anandan 23
LetsVenture 16
Anupam Mittal 16
Group of Angel Investors 14
```



PROBLEM 5

Due to your immense help, your friend startup successfully got seed funding and it is on the operational mode. Now your friend wants to expand his startup and he is looking for new investors for his startup. Now you again come as a saviour to help your friend and want to create a list of probable new new investors. Before moving forward you remember your investor friend advice that finding the investors by analysing the investment type. Since your friend startup is not in early phase it is in growth stage so the best-suited investment type is Private Equity. Find the top 5 investors who have invested in a different number of startups and their investment type is Private Equity. Correct spelling of investment types are - "Private Equity", "Seed Funding", "Debt Funding", and "Crowd Funding". Keep an eye for any spelling mistake. You can find this by printing unique values from this column. There are many errors in startup names. Ignore correcting all, just handle the important ones - Ola, Flipkart, Oyo and Paytm.

OBJECTIVES

- Find the top 5 investors who have invested in a different number of startups and their investment type is Private Equity.

DATA CLEANING

- Correct spelling of investment types are - "Private Equity", "Seed Funding", "Debt Funding", and "Crowd Funding"
  - df.InvestmentType.replace("Crowd funding","Crowd Funding",inplace=True)
  - df.InvestmentType.replace("PrivateEquity","Private Equity",inplace=True)
  - df.InvestmentType.replace("SeedFunding","Seed Funding",inplace=True)>
- There are many errors in startup names correct the names of Ola, Flipkart, Oyo and Paytm
  - df.StartupName.replace("Ola Cabs","Ola",inplace=True)
- At First remove all the NaN Values from required columns
  - df.dropna(subset=["InvestorsName","StartupName",],inplace=True)
- crate a DATAFRAME by ignoring all the "Undisclosed Investors" from the data
  - df=df[df.InvestorsName!="Undisclosed Investors"]

In [102…

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

df=pd.read_csv("Datasets/startup_funding.csv")
df.dropna(subset=["StartupName","InvestorsName"],inplace=True)
df.StartupName.replace("Ola Cabs","Ola",inplace=True)
df.StartupName.replace("Olacabs","Ola",inplace=True)
df.StartupName.replace("Filpkart.com","Flipkart",inplace=True)
df.StartupName.replace("Flipkart.com","Flipkart",inplace=True)
df.StartupName.replace("Oyo Rooms","Oyo",inplace=True)
df.StartupName.replace("OYO Rooms","Oyo",inplace=True)
df.StartupName.replace("Oyorooms","Oyo",inplace=True)
df.StartupName.replace("OyoRooms","Oyo",inplace=True)
df.StartupName.replace("Paytm Marketplace","Paytm",inplace=True)
df.InvestmentType.replace("Crowd funding","Crowd Funding",inplace=True)
df.InvestmentType.replace("PrivateEquity","Private Equity",inplace=True)
df.InvestmentType.replace("SeedFunding","Seed Funding",inplace=True)

df=df[df.InvestorsName!="Undisclosed investors"]
df=df[df.InvestorsName!="undisclosed investors"]
df=df[df.InvestorsName!="Undisclosed Investors"]
df=df[df.InvestorsName!="undisclosed investor"]
```

- Retreive the InvestmentType Column containing the Private Equity

In [103…

```
df=df[(df.InvestmentType=="Private Equity")]
```

- Now create a panadas DATAFRAME for startup_name,investor_name with the columns names StartupName and InvestorsName respectively.
- Group the columns InvestorsName and StartupName
- use nunique() to return all the unique Investors name with respect to columns
- Sort the DATAFRAME in descending order with respect to values

In [104…

```
startup_name=[]
investor_name=[]
for index,row in df.iterrows():
    s_name=row["StartupName"]
    i_name=row["InvestorsName"]
    t_name=row["InvestmentType"]
    i_name=i_name.split(",")
    for names in i_name:
        if names!="":
            investor_name.append(names.strip())
            startup_name.append(s_name)
df=pd.DataFrame({"StartupName":startup_name,"InvestorsName":investor_name})
df=df.groupby("InvestorsName")["StartupName"].nunique()
df=df.sort_values(ascending=False)[0:5]
df
```

Out[104…

```
InvestorsName
Sequoia Capital    45
Accel Partners     43
Kalaari Capital    35
Blume Ventures     27
SAIF Partners      24
Name: StartupName, dtype: int64
```

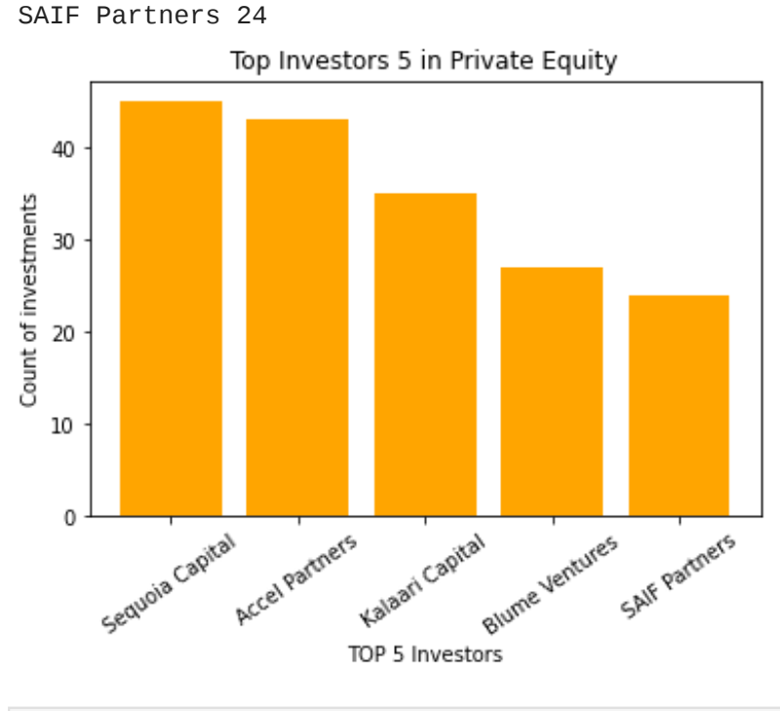
DATA VISUALIZAION

- For DATA VISUALIZATION we import matplotlib library
- The BAR GRAPH is the appropriate DATA VISUALIZATION to represent the TOP Investors with their Count of Number of Fundings

In [105…

```
investor=[]
investor_count=[]
for i in range(0,5):
    print(df.index[i],df.values[i])
    investor.append(df.index[i])
    investor_count.append(df.values[i])
plt.bar(investor,investor_count,color="orange")
plt.xticks(rotation=35)
plt.title("Top Investors 5 in Private Equity")
plt.xlabel("TOP 5 Investors")
plt.ylabel("Count of investments")
plt.show()
```

```
Sequoia Capital 45
Accel Partners 43
Kalaari Capital 35
Blume Ventures 27
SAIF Partners 24
```



In [ ]: