Project: Case Study (Part - II)

Ques 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. That means, find the location where startups has received funding maximum number of times. Plot the bar graph between location and number of funding. Take city name "Delhi" as "New Delhi". Check the case-sensitiveness of cities also. That means, at some place instead of "Bangalore", "bangalore" is given. Take city name as "Bangalore". For few startups multiple locations are given, one Indian and one Foreign. Consider the startup if any one of the city lies in given locations.

Solution- In order to find the location where the most funding is done, we can use numpy arrays to get the cities from the dataset. But before implementing the arrays, we need to solve the naming discrepancy in the CityLocation column of the dataset. We can solve this by using replace function in the pandas library as- ***pandas.dataframe.attribute.replace()***

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
dataset = pd.read_csv('startup_funding.csv', skipinitialspace = True, encoding = 'UTF-8')
df = dataset.copy()
df.CityLocation.replace('bangalore', 'Bangalore', inplace = True)
df.CityLocation.replace('Delhi', 'New Delhi', inplace = True)
```

After solving the discrepancy, we can fetch the data inside the column 'CityLocation' using ***pandas.dataframe.Column_name***:-

```
city_data = np.array(df.CityLocation)
```

Here df is the dataframe of the dataset. But since every row of city_data can contain more than one location so we need to separate these location out, hence for this we can use ***split()*** functionality in python. Since multiple locations are in form of 'A/B' hence we can 'A/B' to [A, B] by using split function.

```
cities = []
for c in city_data:
if c != np.nan:
x = str(c).split('/')
for k in x:
if k[0] == ' ':
k = k[1:]
if k[-1] == ' ':
k = k[:-1]
cities.append(k)
cities = np.array(cities)
```

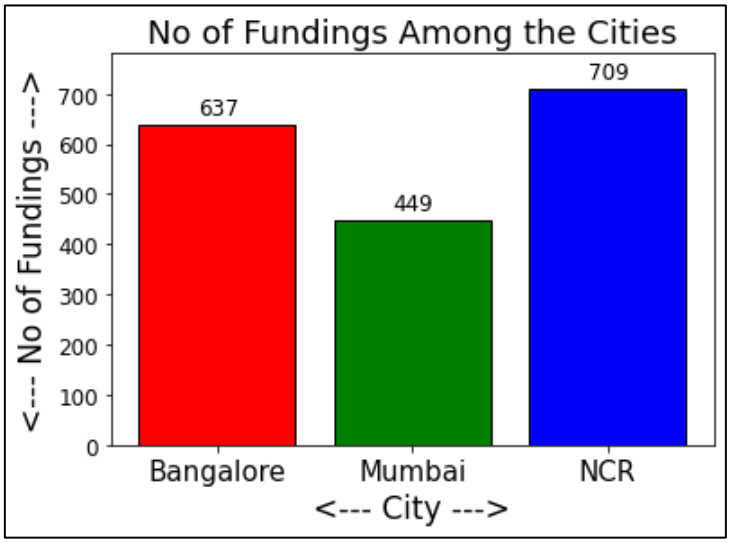
np.nan is the **NULL** value if the row does not contain any data in it and since location can contain extra space as initial character or last character and lead to another discrepancy, therefore we need to remove it. After the getting the all location from the city_data, we can calculate the frequency of city occurred in the cities.array using ***numpy.where()*** functionality:-

```
bn = len(np.where(cities == 'Bangalore')[0])
mum = len(np.where(cities == 'Mumbai')[0])
ncr = len(np.where((cities == 'New Delhi')|(cities == 'Gurgaon')|(cities == 'Noida'))[0])
```

numpy.where() returns the indices of the required the condition as a array in a tuple so for the frequency we can use ***len()*** function in python which returns the length of the array. Since NCR consist of 3 regions:- **New Delhi, Gurgaon, Noida**, we need to 3 conditions and taking overall bitwise OR. Since we have frequency the all the three cities, we can draw the bar graph using ***matplotlib*** library in python:-

```
cities = ['Bangalore', 'Mumbai', 'NCR']
values = [bn,mum,ncr]
plt.bar(cities, values, color = ['Red', 'Green', 'Blue'], edgecolor = 'black')
plt.xlabel('<--- City --->', fontsize = 17)
plt.ylabel('<--- No of Fundings --->', fontsize = 17)
plt.ylim(0, 780)
plt.yticks(fontsize = 12)
plt.xticks(rotation = 0, fontsize = 15)
for i in range(len(cities)):
    plt.text(i-0.1, values[i]+20, values[i], fontsize = 12)
plt.title('No of Fundings Among the Cities', fontsize = 18)
plt.show()
```

Here **cities** are x-values and **values** are y-values for the graph, x-label and y-label are added in order to make graph interactive. **plt.txt** is used to writing the exact frequency of the Cities and title is given in order to understand the graph.



On observing the Bar Graph of Number of fundings Among Bangalore, Mumbai and NCR, we found that NCR has maximum number of fundings, hence the person should consider **NCR** in order to establish the start-up.

Ques 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.

Solution- In order to find the investor who has invested maximum number of times, we can use the same approach as above i.e., we can use **numpy** arrays to count the frequency of the name that occurred in the array. In order to get the list of the investors name we can use ***pandas.Dataframe.ColumeName*** :-

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
dataset = pd.read_csv('startup_funding.csv', skipinitialspace = True, encoding = 'UTF-8')
df = dataset.copy()
```

After creating the dataframe of the Dataset, we can create the numpy array of Investors Name:-

```
investors_data = np.array(df.InvestorsName)
investors = []
for inv in investors_data:
    if inv != np.nan:
        x = str(inv).split(', ')
        for y in x:
            z = y.split(' and ')
            for k in z:
                a = k.split(',')
                for i in a:
                    if 'Undisclosed' not in i:
                        investors.append(i)
```

Since each in the column **df.InvestorName** contains more than one value, we need to separate each and every name with the help of the ***split()*** function in python on the basis of **',** ', ' **and** ' and **',**'. It is asked to ignore '**Undisclosed**' Investors and therefore we won't be including the investor. Now we can put these investors in a list and create a **numpy** array using ***numpy.array()***.

```
investors = np.array(investors)
inv = np.unique(investors)
```

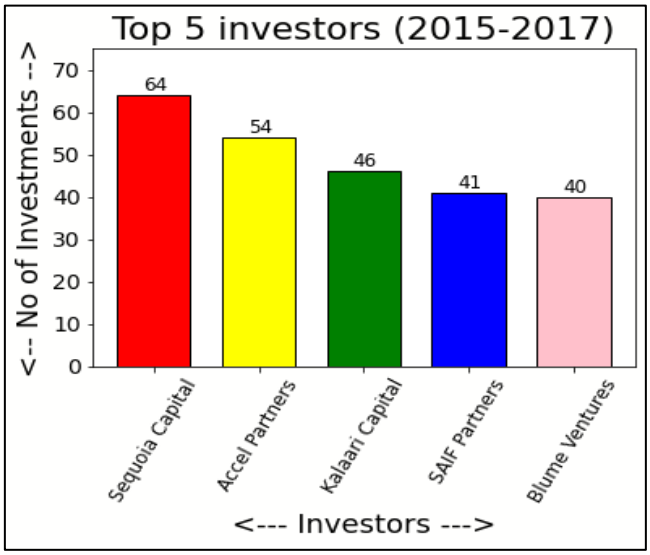
numpy.unique() creates the array of the names which are all unique. All though we can count the frequency of each name from previous array too, it could lead to errors and hence for the sake of the error free code we are using ***np.unique()*** array. Now since inv array has all unique values which calculate the frequency of each name in inv using ***np.where()*** where it returns the indices where the name is present in the array and since we need the just frequency we will use ***len()*** function in python to calculate the length of the array returned by ***np.where()*** function.

```
def f(x):
    return x[1]
invfreq = []
for i in inv:
    x = len(np.where(investors == i)[0])
    invfreq.append([i,x])
invfreq.sort(key = f, reverse = True)
invest = []
times = []
for i in range(5):
    invest.append(invfreq[i][0])
    times.append(invfreq[i][1])
```

invfreq is a list which contains the name of the investors and the frequency of the investor in each row. Since we need to sort the **invfreq** on the basis of frequency we use **key** parameter in ***sort()*** functionality and passed a function f which returns the frequency of each name in the list tells the sort function to sort on the basis of the frequency rather than name and since we need to sort the list in descending order, we passed the parameter **reverse** as **True**. Now since we need on the top 5 we can put the names and the frequency of the investors in the different list. Now we have investors with their number of investments. We can draw the bar graph of top 5 investors using **matplotlib.pyplot** library:-

```
plt.bar(invest, times, width = 0.7, color = ['red', 'yellow', 'green', 'blue', 'pink'], edgecolor
= 'black')
for i in range(5):
    plt.text(i-0.1, times[i]+1, times[i], fontsize = 12)
plt.xlabel('<--- Investors --->', fontsize = 18)
plt.ylabel('<-- No of Investments -->', fontsize = 18)
plt.ylim(0, 75)
plt.xticks(rotation = 60, fontsize = 12)
plt.yticks(fontsize = 14)
plt.title('Top 5 investors (2015-2017)', fontsize = 22)
plt.show()
```

Here **invest** and **times** are the x-values and y-values respectively, **color** is set in order to represent different investors, **edgecolor** is outline the edges of the bar. ***text()*** function is used to put the no of investments by each investor, labels help in the understanding the graph, since the names in the x-values are overlapping each other hence rotation in the ***xticks()*** functionality helps in the rotate the name to some angle so the names of the investors do not overlap.



On the observing the graph, we can say the top 5 investors are:- **'Sequoia Capital'** having, **'64'**, **'Accel Partners'** having **'54'**, **'Kalaari Capital'** having **'46'**, **'SAIF Partners'** having **'41'** and **'Blume Ventures'** having **'40'**, investments respectively.

Ques 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.

Solution- For finding the investors who have invested in different companies, we need to find the list of the companies for the each investor has invested, then we need to separate out the unique companies as told above some investors have investment in the companies more than one companies. But before implementing this, we need to handle in the discrepancy in the **'StartupName'** column in the dataset and for this we can ***pandas.Dataframe.replace()*** functionality.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
dataset = pd.read_csv('startup_funding.csv', skipinitialspace = True, encoding = 'UTF-8')
df = dataset.copy()
df.StartupName.replace('Flipkart.com', 'Flipkart', inplace = True)
df.StartupName.replace('Olacabs', 'Ola Cabs',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)
```

Approach for the solution as the previous question, we need to make a few changes in our code. Now rather than listing the name of the investors, we will list the name of the companies invested by them too. For this we will create a **numpy** array for the investors as well as for the companies. So for each row, we will get the name of the company and its investors.

```
investors = np.array(df.InvestorsName)
companies = np.array(df.StartupName)
```

Now we get the list of all investors and companies, we sort out the list of the companies as per of each investor. For this we can use **Dictionary** data structure in the python. Now before implementing this, since each row of the investors contains more than one investor. Hence, we need to separate out each investor and form the list of the company according to that.

```
inc = {}
for i in range(len(investors)):
    names1 = str(investors[i]).split(', ')
    for j in range(len(names1)):
        names2 = names1[j].split(' and ')
        for k in range(len(names2)):
            names3 = names2[k].split(',')
            for n in names3:
                if n == ' ' or n == '' or n == 'Undisclosed Investors' or n == 'Undisclosed invest
ors':
                    continue
                if n not in inc:
                    inc[n] = [companies[i]]
                else:
                    inc[n].append(companies[i])
```

In the above code, the names of investors has been separated using ***split()*** functionality in python and list of the companies is created and stored in the dictionary '**inc**' which contains **investors** as **key** and **list of the companies** as **value**. As told above, we need to ignore **empty values** and '**Undisclosed**' investors hence we are not including it in our dictionary. Since we get the investors and list of the companies they have invested, now we can move for further procedure. We can create the list of the investors and no of the different companies by using **set** data structure in python.

```
inf = []
for i in inc:
    inf.append([i, len(set(inc[i]))])
```

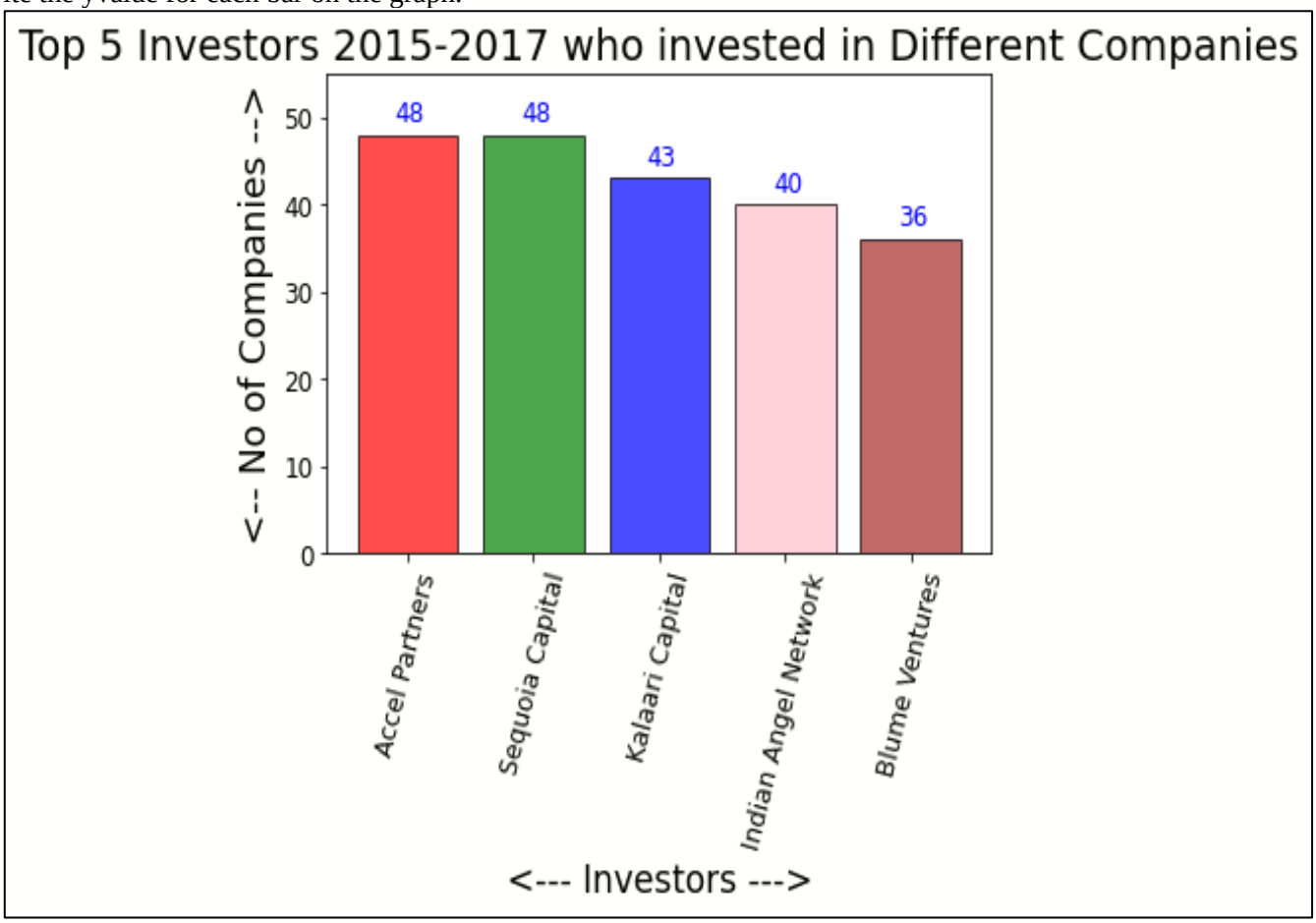
Since, **set** is the data structure which contains only **unique** items in it, hence if investors has invested more than one time in the company then still it will store for one time only. And thus we need total no of unique companies invested by an investor, we can ***len()*** functionality to find the length of the set. After getting the list of the investors with total number of the different companies invested, we can sort out the list as per no of companies using ***sort()*** functionality.

```
inf.sort(key = f, reverse = True)
comp = []
inv = []
for i in range(5):
    comp.append(inf[i][0])
    inv.append(inf[i][1])
```

To sort inf as per the no of companies, set **key** parameter as function 'f', as it returns the total no of companies invested by each investor, to sort in descending order set **reverse** parameter as **True**. As the list has been sorted, we can store the top 5 investors and no of companies they have invested in separate lists. Now we can plot graph the bar using **matplotlib.pyplot** library in python.

```
c = ['red','green','blue','pink','brown']
plt.bar(comp, inv, edgecolor = 'black', color = c, alpha = 0.7)
plt.xticks(rotation = 75, fontsize = 12)
plt.yticks(fontsize = 12)
plt.ylim(0,55)
for i in range(5):
    plt.text(i-0.1, inv[i]+1.5, inv[i], fontsize = 12, color = 'blue')
plt.xlabel('<--- Investors --->', fontsize = 17)
plt.ylabel('<-- No of Companies -->', fontsize = 17)
plt.title('Top 5 Investors 2015-2017 who invested in Different Companies', fontsize = 20)
plt.show()
```

c is the list of the **colors** which is used to represent different companies. **comp** and **inv** are the xvalues and yvalues of the graph. **alpha** parameter is used to define the opacity of the color inside the bar. ***text()*** function has been used to write the yvalue for each bar on the graph.



On observing the graph of the Top 5 investors who invested in Different Companies are - **Accel Partners**, **Sequioia Capital**, **Kalaari Capital**, **Indian Angel Network**, **Blume Ventures** have invested in **48**, **48**, **43**, **40**, **36** different companies respectively.

Ques 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.

Solution- In order to find top 5 investors who have funded through Seed Funding or Crowd Funding, we can use numpy arrays to get the list of investors who funded either through 'Seed Funding' or 'Crowd Funding'. But before implementing, we need to solve discrepancy in '**InvestmentType**' and '**StartupName**' column of the dataset. For this we can use ***pandas.dataframe.replace()*** function in **pandas** library.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
dataset = pd.read_csv('startup_funding.csv', skipinitialspace = True, encoding = 'UTF-8')
df = dataset.copy()
df.InvestmentType.replace('SeedFunding', 'Seed Funding', inplace = True)
df.InvestmentType.replace('Crowd funding', 'Crowd Funding', inplace = True)
df.InvestmentType.replace('PrivateEquity', 'Private Equity', inplace = True)
df.StartupName.replace('Flipkart.com', 'Flipkart', inplace = True)
df.StartupName.replace('Olacabs', 'Ola Cabs',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)
```

Once we solve the problem of discrepancy, we can find out the list of investors who invested through '**Seed Funding**' or '**Crowd Funding**' using boolean indexing using condition :- **df.InvestmentType == 'Seed Funding'**, **df.InvestmentType == 'Crowd Funding'**.

```
cond1 = df.InvestmentType == 'Seed Funding'
cond2 = df.InvestmentType == 'Crowd Funding'
investors = np.array(df.InvestorsName[(cond1)|(cond2)])
companies = np.array(df.StartupName[(cond1)|(cond2)])
```

Once we got the list of the investors and companies having investment type '**Seed Funding**' or '**Crowd Funding**'. We can find top 5 investors who have most invested in different companies as same above problem. We will use **Dictionary** data structure for to keep all companies invested by each investor.

```
cond1 = df.InvestmentType == 'Seed Funding'
cond2 = df.InvestmentType == 'Crowd Funding'
investors = np.array(df.InvestorsName[(cond1)|(cond2)])
companies = np.array(df.StartupName[(cond1)|(cond2)])
invc = {}
for i in range(len(investors)):
    name1 = str(investors[i]).split(', ')
    for j in range(len(name1)):
        name2 = name1[j].split(' and ')
        for k in range(len(name2)):
            name3 = name2[k].split(',')
            for n in name3:
                if n == '' or n == ' ' or n == 'Undisclosed Investors' or n == 'Undisclosed invest
ors':
                    continue
                if n not in invc:
                    invc[n] = [companies[i]]
                else:
                    invc[n].append(companies[i])
```

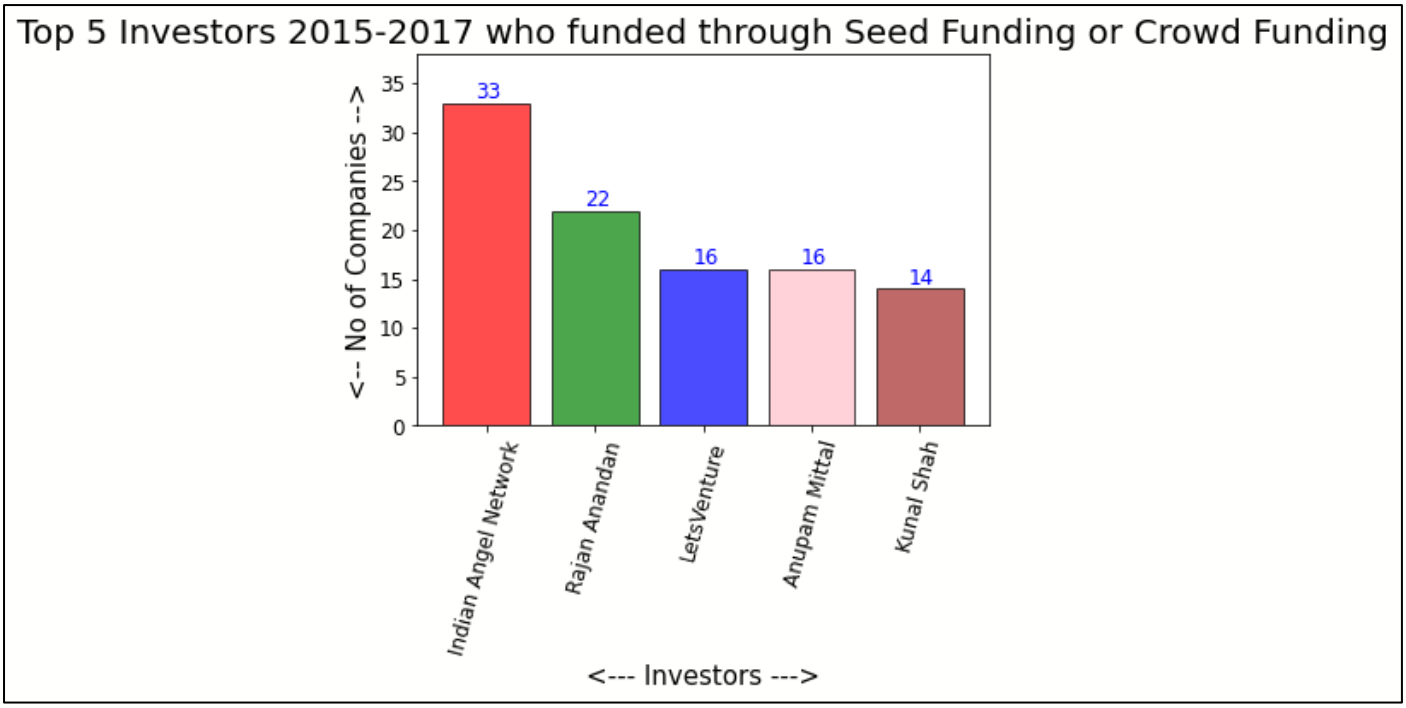
After the getting the list of the companies invested by each investor, we used **Set** data structure in python to find out unique companies and since we need only just frequency, we use ***len()*** functionality in python to find out the no of different companies invested, and add it another list.

```
invf = []
for i in invc:
    invf.append([i, len(set(invc[i]))])
invf.sort(key = f, reverse = True)
inv,comp = [],[]
for i in range(5):
    comp.append(invf[i][0])
    inv.append(invf[i][1])
```

After the getting the list of the investors with the no of companies they have invested, we can sort it using ***sort()*** functionality. We can save top 5 investors and no of companies into another list. Now we got top 5 investors and companies we can plot a graph using **matplotlib.pyplot** library in python.

```
c = ['red','green','blue','pink','brown']
plt.bar(comp, inv, edgecolor = 'black', color = c, alpha = 0.7)
plt.xticks(rotation = 75, fontsize = 12)
plt.yticks(fontsize = 12)
plt.ylim(0,38)
for i in range(5):
    plt.text(i-0.1, inv[i]+0.5, inv[i], fontsize = 12, color = 'blue')
plt.xlabel('<--- Investors --->', fontsize = 15)
plt.ylabel('<-- No of Companies -->', fontsize = 15)
plt.title('Top 5 Investors 2015-
2017 who funded through Seed Funding or Crowd Funding', fontsize = 20)
plt.show()
```

c is the list of the **colors** which is used to represent different companies. **comp** and **inv** are the xvalues and yvalues of the graph. **alpha** parameter is used to define the opacity of the color inside the bar. ***text()*** function has been used to write the yvalue for each bar on the graph.



On observing the graph of the Top 5 investors who invested in Different Companies with investment type as Seed Funding or Crowd Funding are – **Indian Angel Network**, **Rajan Anandan**, **LetsVenture**, **Anupam Mittal**, **Kunal Shah** have invested in **33**, **22**, **16**, **16**, **14** different companies respectively.

Ques 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.

Solution- Before we get started finding the solution of the above problem, we need to remove the discrepancy in the **InvestmentType** and **StartupName** column of the dataset. For this, we can use ***pandas.Dataframe.replace()*** function in pandas library.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
dataset = pd.read_csv('startup_funding.csv', skipinitialspace = True, encoding = 'UTF-8')
df = dataset.copy()
df.InvestmentType.replace('SeedFunding', 'Seed Funding', inplace = True)
df.InvestmentType.replace('Crowd funding', 'Crowd Funding', inplace = True)
df.InvestmentType.replace('PrivateEquity', 'Private Equity', inplace = True)
df.StartupName.replace('Flipkart.com', 'Flipkart', inplace = True)
df.StartupName.replace('Olacabs', 'Ola Cabs',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)
```

After resolving the discrepancy in the columns of the dataset, we need to find out the list of investors and companies who have invested through '**Private Equity**' and to achieve this we need a separate list of investors and companies with the help boolean indexing in the **numpy** and **pandas** library using condition- **df.InvestmentType** == '**Private Equity**'.

```
cond = df.InvestmentType == 'Private Equity'
investors = np.array(df.InvestorsName[cond])
companies = np.array(df.StartupName[cond])
```

Now after getting the list of the investors and companies having investment type as '**Private Equity**', we can use **Dictionary** data structure to find out the list companies invested by each investor.

```
invc = {}
for i in range(len(investors)):
    name1 = str(investors[i]).split(', ')
    for j in range(len(name1)):
        name2 = name1[j].split(' and ')
        for k in range(len(name2)):
            name3 = name2[k].split(',')
            for n in name3:
                if n == '' or n == ' ' or n == 'Undisclosed Investors' or n == 'Undisclosed invest
ors':
                    continue
                if n not in invc:
                    invc[n] = [companies[i]]
                else:
                    invc[n].append(companies[i])
```

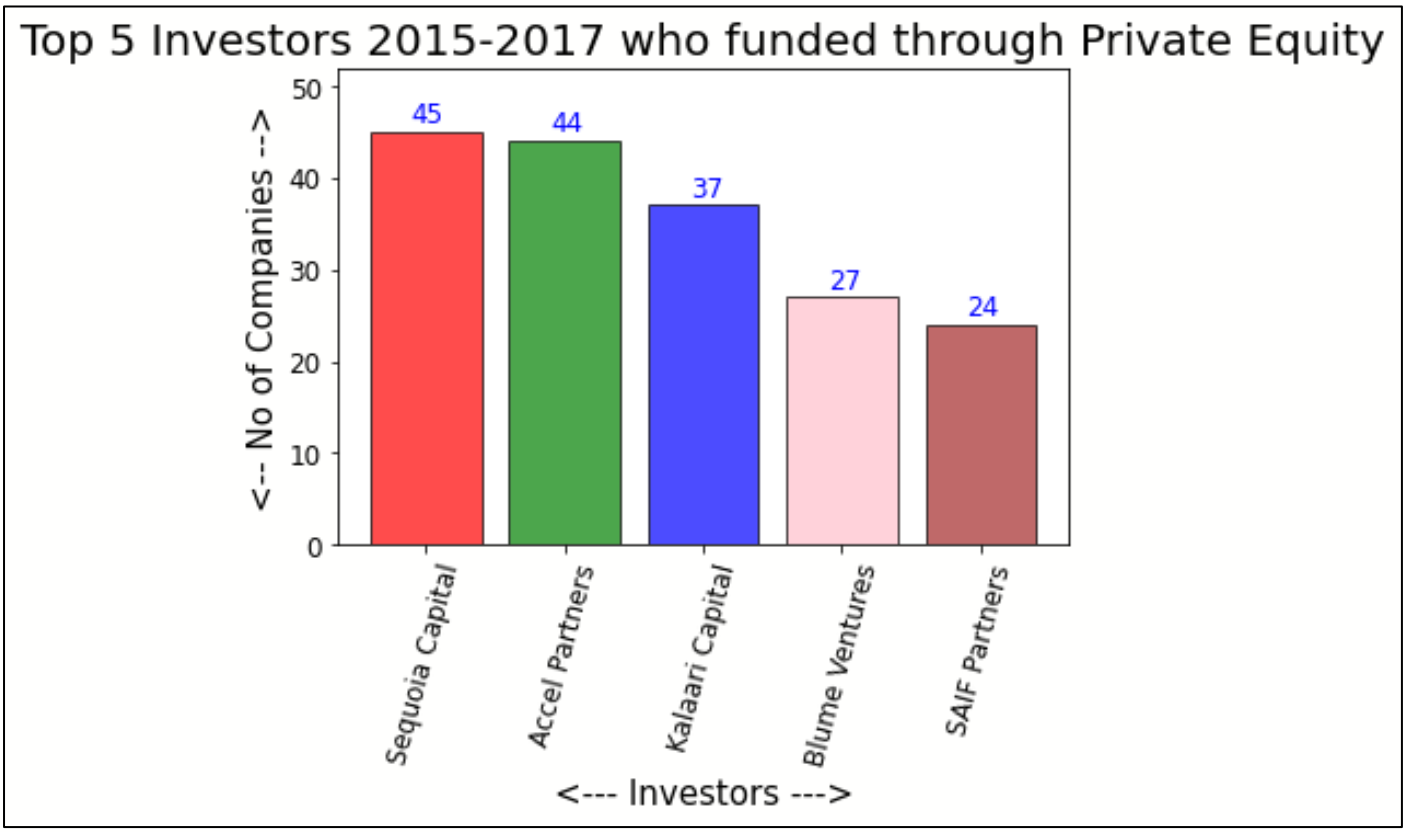
After getting the list of investors with the list of companies that they have invested in a dictionary, we can use **Set** data structure where we get only a list having all unique name of the companies and hence we need the no of different companies we can use ***len()*** functionality in python for the frequency.

```
invf = []
for i in invc:
    invf.append([i, len(set(invc[i]))])
invf.sort(key = f, reverse = True)
inv,comp = [],[]
for i in range(5):
    comp.append(invf[i][0])
    inv.append(invf[i][1])
```

We can sort the list of investors using ***sort()*** functionality, separate out top 5 investors in a different list. Now we have list of investors, and list of no of companies we can use to plot the graph using ***matplotlib.pyplot*** library in python.

```
c = ['red','green','blue','pink','brown']
plt.bar(comp, inv, edgecolor = 'black', color = c, alpha = 0.7)
plt.xticks(rotation = 75, fontsize = 12)
plt.yticks(fontsize = 12)
plt.ylim(0,52)
for i in range(5):
    plt.text(i-0.1, inv[i]+1, inv[i], fontsize = 12, color = 'blue')
plt.xlabel('<--- Investors --->', fontsize = 15)
plt.ylabel('<-- No of Companies -->', fontsize = 15)
plt.title('Top 5 Investors 2015-2017 who funded through Private Equity', fontsize = 20)
plt.show()
```

c is the list of the **colors** which is used to represent different companies. **comp** and **inv** are the xvalues and yvalues of the graph. **alpha** parameter is used to define the opacity of the color inside the bar. ***text()*** function has been used to write the yvalue for each bar on the graph.



On observing the graph of the Top 5 investors who invested in Different Companies with investment type as Private Equity are – **Sequoia Capital**, **Accel Partners**, **Kalaari Capital**, **Blume Ventures**, **SAIF Partners** have invested in **45, 44**, **37**, **27**, **24** different companies respectively.