

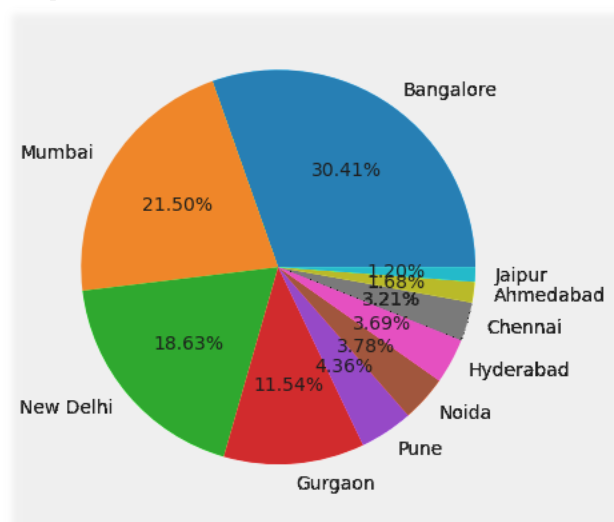
1. Top 5 Cities for receiving funds for a startup:

My friend Rahul had developed a product and is now willing gather funds for his startup. So I decided to help him in this matter with my data analytics expertise. I have mined and gathered a CSV data file called "Startup Funding" which contains a list of all the startups who have got funding from different investors during the tenure 31/01/2015 to 01/08/2017.

Upon analysis using Python with tools like NumPy, Pandas and Matplotlib I was able to come to the following conclusion:

The top 10 cities in which start-ups got funded are as follows:

Bangalore 635
Mumbai 449
New Delhi 389
Gurgaon 241
Pune 91
Noida 79
Hyderabad 77
Chennai 67
Ahmedabad 35
Jaipur 25



CODE FOR ANALYSIS:

The code that I used goes as below:

This is first half of the code where in the first para I have opened the file using a CSV DictReader which gave me a dictionary of all the contents in the data file with the column headers as key and the column elements as values. I personally prefer this over the pandas data frame as its much easier and versatile to handle and requires memorizing less syntaxes for usual operations. I shall be using it more often then pandas throughout this report.

Firstly I had found the unique elements in the column "CityLocation" using "np.unique" function of NumPy array (not part of the main code as it was just for checking) and it helped me to check for different spelling mistakes present in the data file. The first task was to deal with the mistakes refine them for uniformity. Eg: Delhi had to be converted to New Delhi etc. Next task was to create a dictionary of cities so as to keep a count of the frequency of fundings that each city received. The next few lines of the code in second para solves this purpose exactly.

i

```
In [128]: import numpy as np
import csv
import matplotlib.pyplot as plt
with open("/Users/rishismac/Desktop/startup_funding.csv") as file_obj:
    file_data=csv.DictReader(file_obj,skipinitialspace=True)
    city={}

    for row in file_data:
        if "Delhi" in row["CityLocation"]:
            row["CityLocation"]="New Delhi"
        if row["CityLocation"]=="':
            continue

        if city.get(row["CityLocation"].lower()) is None:
            if "/" in row["CityLocation"].lower():
                if city.get(row["CityLocation"].lower().split("/")[0].strip()) is None:
                    city[row["CityLocation"].lower().split("/")[0].strip()]=1
                else:
                    city[row["CityLocation"].lower().split("/")[0].strip()]+=1
            else:
                city[row["CityLocation"].lower()]=1
        else:
            city[row["CityLocation"].lower()]+=1
```

In the second half of code, I have sorted the dictionary in descending order on the basis of values rather than keys. In the final list I have considered only the top 10 cities on the basis of most number of startups they have and also naturally the most number of funding rounds.

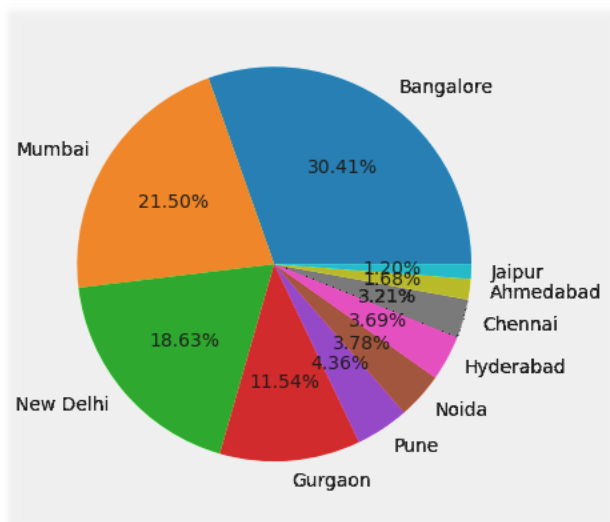
The result along with pie chart is shared above so please have a look.

```

sorted_dict = []
for w in sorted(city, key=city.get, reverse=True):
    sorted_dict.append((w, city[w]))
sorted_dict = sorted_dict[:10]
x = []
y = []
for ele in sorted_dict:
    if len(ele[0].split()) > 1:
        str1 = ele[0].split()[0][0].upper() + ele[0].split()[0][1:]
        str2 = ele[0].split()[1][0].upper() + ele[0].split()[1][1:]
        final = str1 + ' ' + str2
    else:
        str1 = ele[0][0].upper()
        final = str1 + ele[0][1:]
    x.append(final)
    y.append(int(ele[1]))
    print(final, ele[1])
plt.pie(y, labels=x, autopct="%.2f%%")
plt.show()

```

The result along with pie chart is shared below so please have a look. it has been created using the matplotlib module in Python and shows exactly the percentage of funding rounds which happened in the top 10 cities.



Conclusion:

Due to its financial restriction, Rahul can choose only between three locations - Bangalore, Mumbai, and NCR and as per the current trend and the pie chart it is evident that Bangalore is the city where the money is and so I suggested him to pursue Bangalore for his start-up.

2. Top 5 Investors as per funding rounds:

Rahul had gone to Bangalore to kick start his product start-up as per my suggestion but it turns out he had been struggling to garner the attention of the investors and not knowing where to start from it has been mostly labour work pitching to a wide variety of investors. So he again approached me to have a narrower path of search. On taking a closer look at the data I decided to help him find the top 5 investors on the basis of number of funding rounds That they have made. Using a bit more data analytics I had narrowed down on the following list of investors:

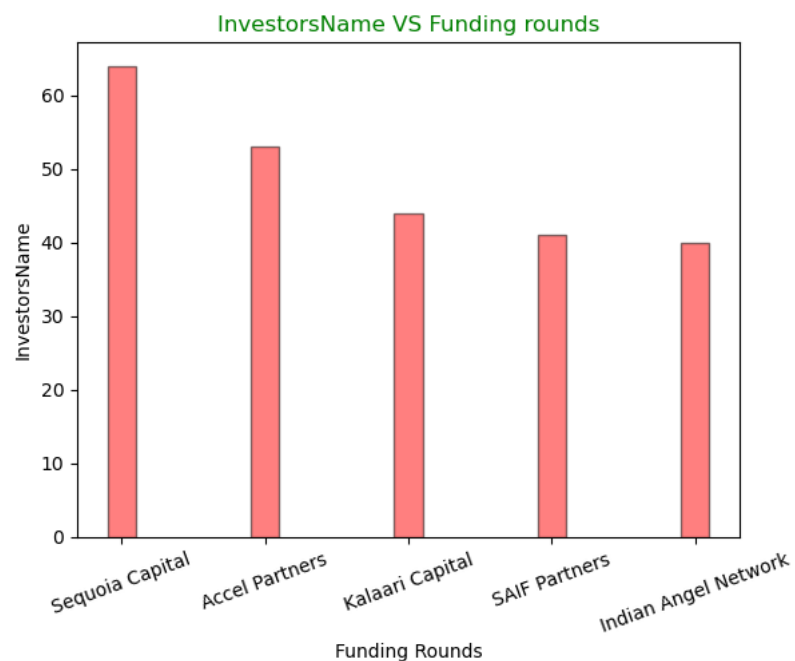
Sequoia Capital 64

Accel Partners 53

Kalaari Capital 44

SAIF Partners 41

Indian Angel Network 40



CODE FOR ANALYSIS:

The code for the above analysis is given below:

The first half deals with the fact that in case of many start-ups there are multiple investors and in order consider each of them we need to split them rather than considering the entire list as one element. This is performed precisely by the split(",") function by using "," as the delimiter.

In the next part considering each investor as a separate entity I have created a dictionary on the basis of their funding rounds.

```
In [289]: import numpy as np
import csv
import matplotlib.pyplot as plt

with open("/Users/rishismac/Desktop/startup_funding.csv") as file_obj:
    file_data=csv.DictReader(file_obj,skipinitialspace=True)
    Investors={}
    for row in file_data:
        if "," in row["InvestorsName"]:
            a=row["InvestorsName"].split(",")

            for i in a:

                if Investors.get(i.strip()) is None:
                    Investors[i.strip()]=1
                else:
                    Investors[i.strip()]+=1

            if row["InvestorsName"]==" " or row["InvestorsName"]=="Undisclosed Investors" or row["InvestorsName"]=="Undisclo

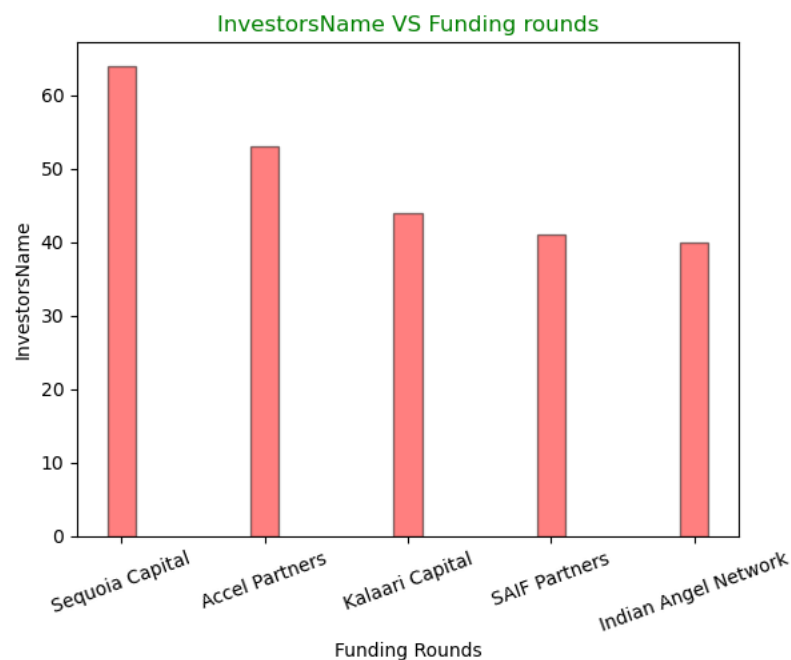
                continue
            if Investors.get(row["InvestorsName"]) is None:
                Investors[row["InvestorsName"]]=1
            else:
                Investors[row["InvestorsName"]]+=1
```

In the next half of code I have sorted the dictionary in descending order on the basis of values rather than keys and plotted a bar graph on Investor name VS Funding rounds using matplotlib module.

```
sorted_startup=[]
for w in sorted(Investors,key=Investors.get,reverse=True):
    if w==" ":
        continue
    sorted_startup.append([w,Investors[w]])
sorted_startup=sorted_startup[:5]
x=[]
y=[]
sorted_startup
for ele in sorted_startup:
    x.append(ele[0])
    y.append(int(ele[1]))
for i in range(5):
    print(x[i],y[i])
plt.bar(x,y,edgecolor="black",width=0.2,color="red",alpha=0.5)
plt.xlabel("Funding Rounds")
plt.ylabel("InvestorsName")
plt.xticks(rotation=20)
plt.title("InvestorsName VS Funding rounds", color="green")
plt.show()
```

As per the graph created ,the following list comes up:

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



3. Top 5 Investors as per funding rounds in different companies:

After re-analysing the dataset I found out that some investors have invested in the same start-up at different number of funding rounds. So before finalising the previous list, yoi wanted to improvise it by finding the top 5 investors who have invested in different number of start-ups. This list will be more helpful than the previous list in finding the investment for Rahul's start-up. The first task was to have uniform name for same startups as there are few spelling errors, like-Ola, Flipkart, Oyo and Paytm (handling only the important ones). And then I segregated the start-ups again as per there frequency ensuring that the repetitions were handled.

CODE FOR ANALYSIS:

The main challenge was to ensure that only the unique investor and start-up pair was only counted rather than counting all the repetitions. I had taken care of that by picking the “investor name/start-up name “ into a dictionary and then taking only the keys of the dictionary to get the unique pairs.

I had to again unpack the investor and start-up names separated by “/” to get a final list of counts per investors. the detailed code for the same is given below:

```
In [17]: import numpy as np
import csv
import matplotlib.pyplot as plt

with open("/Users/rishimac/Desktop/startup_funding.csv") as file_obj:
    file_data=csv.DictReader(file_obj,skipinitialspace=True)
    investor_startup_freq={}
    for row in file_data:
        if row["StartupName"]=="Ola Cabs":
            row["StartupName"]="Ola"
        if row["StartupName"]=="OlaCabs":
            row["StartupName"]="Ola"
        if row["StartupName"]=="Oyo Rooms":
            row["StartupName"]="Oyo"
        if row["StartupName"]=="OyoRooms":
            row["StartupName"]="Oyo"
        if row["StartupName"]=="Oyorooms":
            row["StartupName"]="Oyo"
        if row["StartupName"]=="Flipkart.com":
            row["StartupName"]="Flipkart"
        if row["StartupName"]=="Paytm Marketplace":
            row["StartupName"]="Paytm"
        if row["InvestorsName"]=="":
            continue
        if row["InvestorsName"]==" or row["InvestorsName"]=="Undisclosed Investors" or row["InvestorsName"]=="Undisclo
            continue
        if "," in row["InvestorsName"]:
            a=row["InvestorsName"].split(",")
            for i in a:

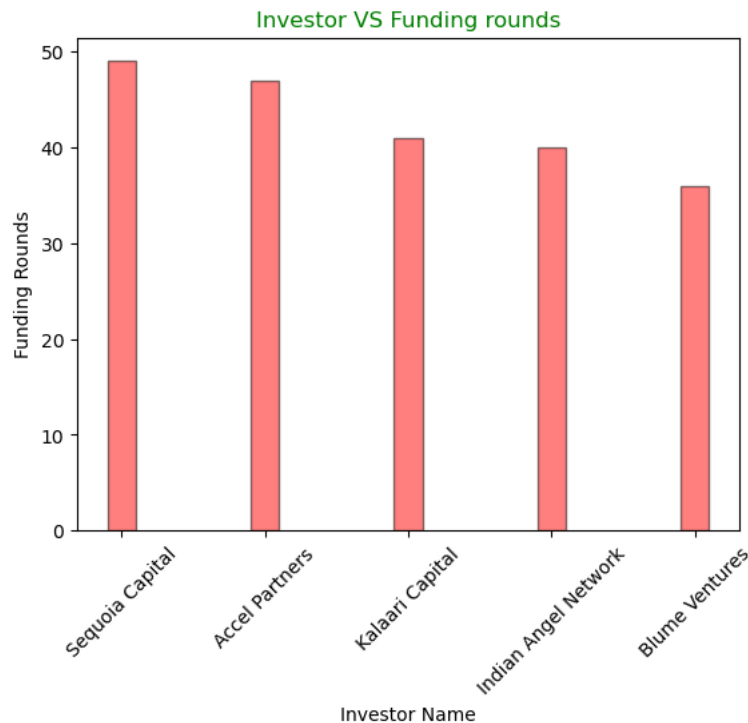
                if investor_startup_freq.get(i.strip()+"/"+row["StartupName"].strip()) is None:
                    investor_startup_freq[i.strip()+"/"+row["StartupName"].strip()]=1
```

```
                else:
                    investor_startup_freq[i.strip()+"/"+row["StartupName"].strip()]+=1
            else:
                if investor_startup_freq.get(row["InvestorsName"].strip()+"/"+row["StartupName"].strip()) is None:
                    investor_startup_freq[row["InvestorsName"].strip()+"/"+row["StartupName"].strip()]=1
                else:
                    investor_startup_freq[row["InvestorsName"].strip()+"/"+row["StartupName"].strip()]+=1

investor_dict={}
lst=list(investor_startup_freq.keys())
for i in lst:
    if investor_dict.get(i.split("/")[0]) is None:
        investor_dict[i.split("/")[0]]=1
    else:
        investor_dict[i.split("/")[0]]+=1

sorted_dict=[]
for w in sorted(investor_dict,key=investor_dict.get,reverse=True):
    if w=="":
        continue
    sorted_dict.append([w,investor_dict[w]])
final_list=sorted_dict[:5]
x=[]
y=[]
for ele in final_list:
    x.append(ele[0])
    y.append(int(ele[1]))
    print(ele[0],int(ele[1]))
plt.bar(x,y,edgecolor="black",width=0.2,color="red",alpha=0.5)
plt.xticks(rotation=45)
plt.ylabel("Funding Rounds")
plt.xlabel("Investor Name")
plt.title("Investor VS Funding rounds", color="green")
plt.show()
```

Sequoia Capital 49
Accel Partners 47
Kalaari Capital 41
Indian Angel Network 40
Blume Ventures 36



Conclusion:

As per my suggestion he decided to approach the above list of investors for improving the chances of getting funding.

3. Top 5 Investors as per funding rounds in different companies with Seed and Crowd Funding types only:

Even after putting so much effort in finding the probable investors, it didn't turn out to be helpful for Rahul. So I went to Rahul to understand the situation better and he explained to me about the different Investment Types and their features. This new information will be helpful in finding the right investor. Since his start-up is at an early stage, the best-suited investment type would be - Seed Funding and Crowd Funding. So I decided to find the top 5 investors who have invested in a different number of start-ups and their investment type is Crowd Funding or Seed Funding. Correct spelling of investment types are - "Private Equity", "Seed Funding", "Debt Funding", and "Crowd Funding". So I had to deal with them in the process as well take care of the spellings in start-up names like: Ola, Flipkart, Oyo and Paytm.

Code for analysis:

the first task of correcting the names has been done by checking for unique investment type using “np.unique()” and looking for all wrong spellings and then correcting them.

Then using similar method of dealing with repetitions as in the above problem I made a new dictionary of investors and funding rounds keeping “Seed Funding” and “Crowd Funding” as Filters.

The code for the same is given below:

```
In [22]: import numpy as np
import csv
import matplotlib.pyplot as plt

with open("/Users/rishismac/Desktop/startup_funding.csv") as file_obj:
    file_data=csv.DictReader(file_obj,skipinitialspace=True)
    investor_startup_freq={}
    for row in file_data:
        if row["StartupName"]=="Ola Cabs":
            row["StartupName"]="Ola"
        if row["StartupName"]=="Olacabs":
            row["StartupName"]="Ola"
        if row["StartupName"]=="Oyo Rooms":
            row["StartupName"]="Oyo"
        if row["StartupName"]=="OyoRooms":
            row["StartupName"]="Oyo"
        if row["StartupName"]=="Oyorooms":
            row["StartupName"]="Oyo"
        if row["StartupName"]=="Flipkart.com":
            row["StartupName"]="Flipkart"
        if row["StartupName"]=="Paytm Marketplace":
            row["StartupName"]="Paytm"
        if row["InvestmentType"]=="PrivateEquity":
            row["InvestmentType"]="Private Equity"
        if row["InvestmentType"]=="SeedFunding":
            row["InvestmentType"]="Seed Funding"
        if row["InvestmentType"]=="Crowd funding":
            row["InvestmentType"]="Crowd Funding"
        if row["InvestorsName"]==" " or row["InvestorsName"]=="Undisclosed Investors" or row["InvestorsName"]=="Undisclo":
            continue
        if row["InvestmentType"]=="Crowd Funding" or row["InvestmentType"]=="Seed Funding":
            if " " in row["InvestorsName"]:
                a=row["InvestorsName"].split(",")
                for i in a:
                    if investor_startup_freq.get(i.strip()+"/"+row["StartupName"].strip()) is None:
                        investor_startup_freq[i.strip()+"/"+row["StartupName"].strip()]=1
                    else:
                        investor_startup_freq[i.strip()+"/"+row["StartupName"].strip()]+=1
            else:
                investor_startup_freq[row["InvestorsName"].strip()+"/"+row["StartupName"].strip()]+=1
        else:
            if investor_startup_freq.get(row["InvestorsName"].strip()+"/"+row["StartupName"].strip()) is None:
                investor_startup_freq[row["InvestorsName"].strip()+"/"+row["StartupName"].strip()]=1
            else:
                investor_startup_freq[row["InvestorsName"].strip()+"/"+row["StartupName"].strip()]+=1

investor_dict={}
lst=list(investor_startup_freq.keys())
for i in lst:
    if investor_dict.get(i.split("/")[0]) is None:
        investor_dict[i.split("/")[0]]=1
    else:
        investor_dict[i.split("/")[0]]+=1

sorted_dict={}
for w in sorted(investor_dict,key=investor_dict.get,reverse=True):
    if w==" ":
        continue
    sorted_dict.append([w,investor_dict[w]])
final_list=sorted_dict[:5]
x=[]
y=[]
for ele in final_list:
    x.append(ele[0])
    y.append(int(ele[1]))
    print(ele[0],int(ele[1]))
plt.bar(x,y,edgecolor="black",width=0.2,color="red",alpha=0.5)
plt.xticks(rotation=45)
plt.ylabel("Funding Rounds")
plt.xlabel("Investor Name")
plt.title("Investor VS Funding rounds in Seed Funding and Crowd Funding", color="green")
plt.show()
```

Indian Angel Network 33

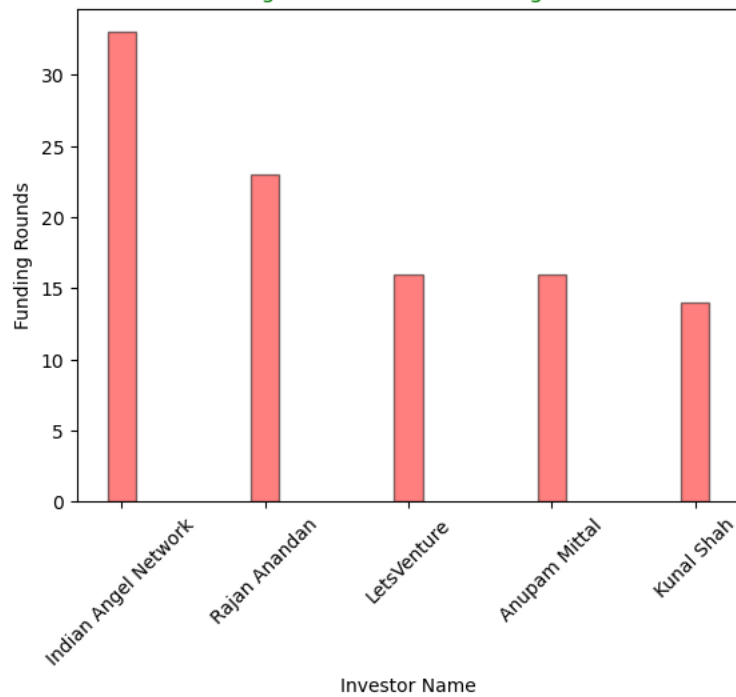
Rajan Anandan 23

LetsVenture 16

Anupam Mittal 16

Kunal Shah 14

Investor VS Funding rounds in Seed Funding and Crowd Funding



Conclusion:

As per the graphs I suggested the following companies: Indian Angel Network, Rajan Anandan, Anupam Mittal, LetsVenture and Kunal Shah.