

PROJECT : CASE STUDY (Part - II)

Name : Shobhit Bansal

Email Id : bshobhit67@gmail.com

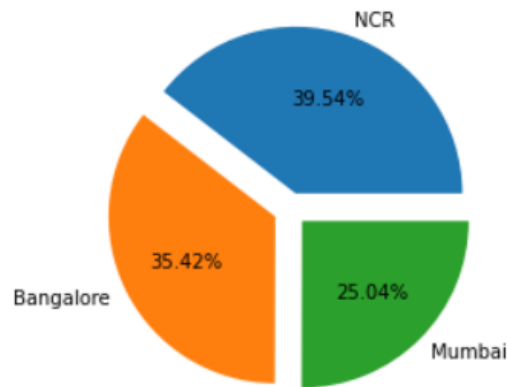
Problem 1 : To find the best suited place for a start up.

Code Logic and Justification : Firstly, the .CSV file is converted into a pandas data frame which only includes the data of City Location. NaN values are handled by simply dropping them as they do not result in any loss of data whatsoever. Many data entries have two locations, one of them is the Indian Base city while the other is a foreign town separated by a “ \ ”. So we use the pre-defined pandas function `df.apply(fn)` where `fn` is a user defined function to remove the foreign city. The spelling errors are corrected using the `.replace` functionality. The `.replace` functionality is also used to define a new data entry i.e. NCR where we replace New Delhi, Gurgaon, Noida by only one NCR. Using `value_counts` it was found that NCR was on the top with 709 fundings followed by Bangalore with 635 and Mumbai with 449 fundings during the given time period. Plotting a pie chart interpreted that out of the number of fundings received by the three locations, NCR topped all with acquiring nearly 40 % of the fundings.

Conclusion : According to the given dataset and list of preferred locations, NCR is the best suited for acquiring Investment

NCR 709
Bangalore 635
Mumbai 449

Pie Plot :

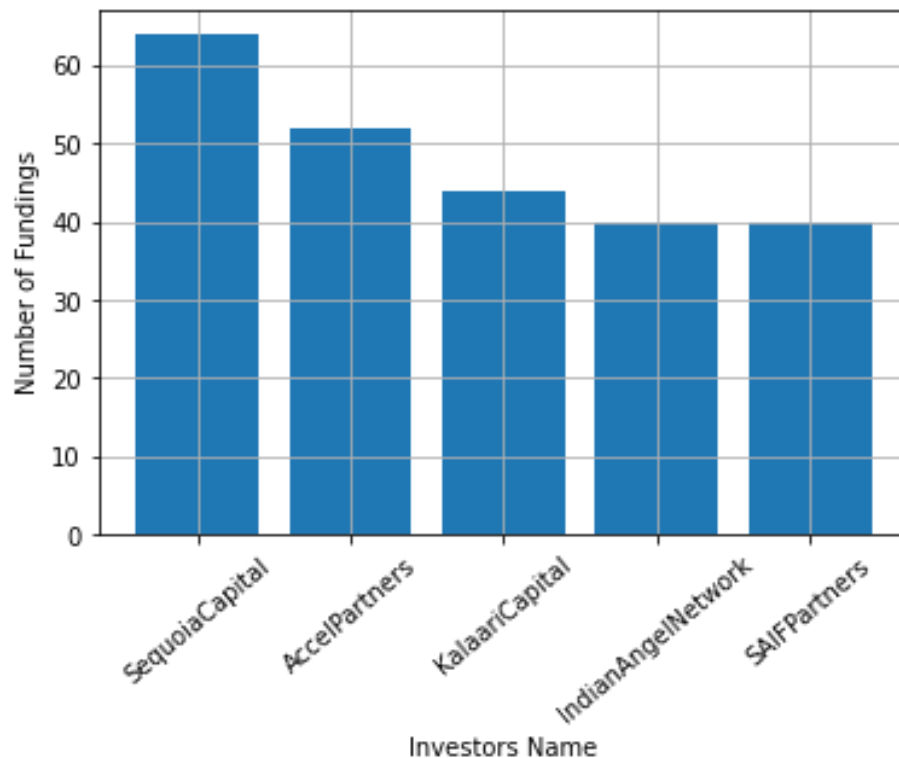


Problem 2 : Top 5 investors who have invested maximum number of times.

Code Logic and Justification : Firstly, the .CSV file is converted into a pandas data frame which only includes the data of Investors Name. NaN values are handled by simply dropping them as they do not result in any loss of data whatsoever. Using the .replace functionality we removed all spaces in the data as many entries had a space after comma, before comma or not at all. So removing spaces makes the task a lot easier. To account for multiple investors investing in the same round of funding, we split the string using the .split with “,” as a delimiter. The frequency of a certain investor is maintained using a dictionary / hashmap. Each entry of the data frame is operated upon using the .apply(function) where function is a user defined function that splits the string data and accordingly update the frequency in the dictionary. Then finally to sort the dictionary, we convert the dictionary into a pandas data frame to use the .sort_values to sort the dictionary. We also plot a bar graph which makes it easier to interpret the result.

Conclusion and Bar Plot: The list of the top 5 investors as well as the bar plot is given below :

```
Sequoia Capital 64
Accel Partners 52
Kalaari Capital 44
Indian Angel Network 40
SAIF Partners 40
```

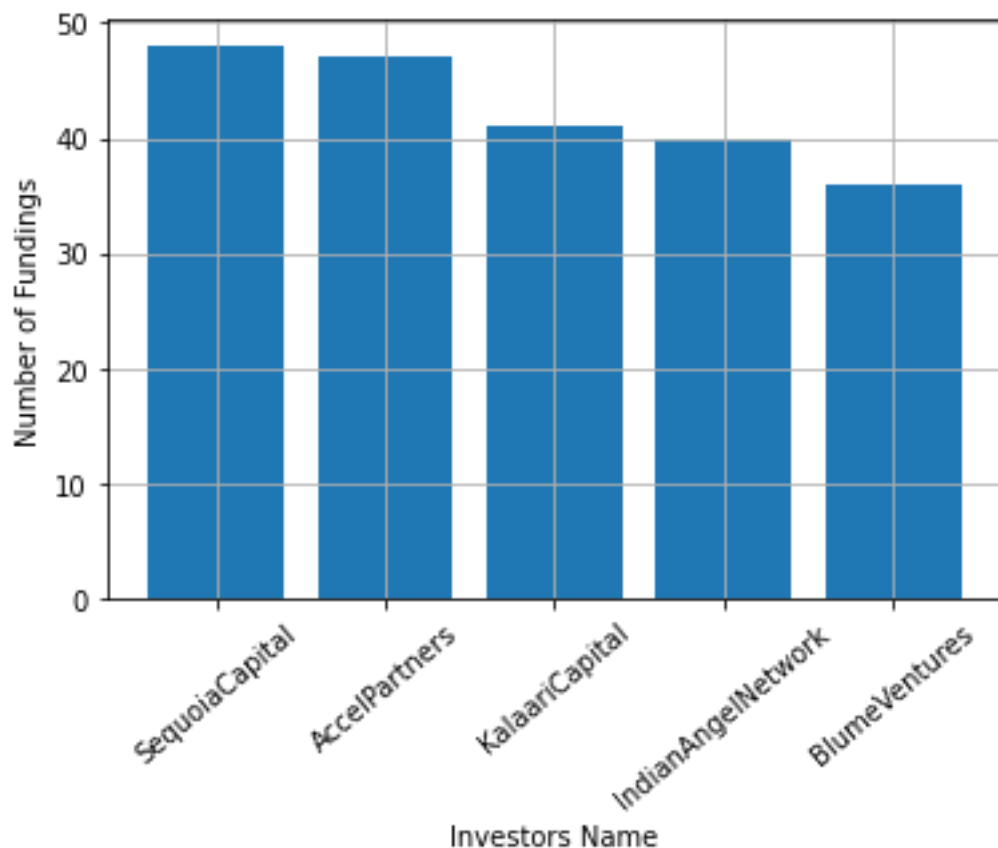


Problem 3 : Top 5 investors who invested maximum number of times in different start up.

Code Logic and Justification : Firstly, the .CSV file is converted into a pandas data frame which only includes the data of Investors Name and Startup Names. NaN values are handled by simply dropping them as they do not result in any loss of data whatsoever. Using the .replace functionality we removed all spaces in the data as many entries had a space after comma, before comma or not at all. So removing spaces makes the task a lot easier. Also using the .replace functionality spelling errors in names of some important start-ups is handled. For investors who have invested multiple number of times in the same start-up, we use two dictionaries. One is to check if the investor had already invested in the same start-up or not. So a dictionary named check is maintained whose keys are the concatenated string of start-up name and investor name. If the string start-up name + Investor name is in the keys of the check dictionary, we move forward. If not, we add 1 against the Investor name which is a key in the second dictionary that maintains the frequency. So the second dictionary contains the unsorted information about investors who invested maximum number of times in different start-ups. Finally to sort the dictionary, we convert the dictionary into a pandas data frame to use the .sort_values to sort the dictionary. We also plot a bar graph which makes it easier to interpret the result.

Conclusion and Bar Plot : The list of the top 5 investors as well as the bar plot is given below :

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



Problem 4 and 5 : Top 5 Investor who invested maximum number of times in different start-ups where type of investment was :

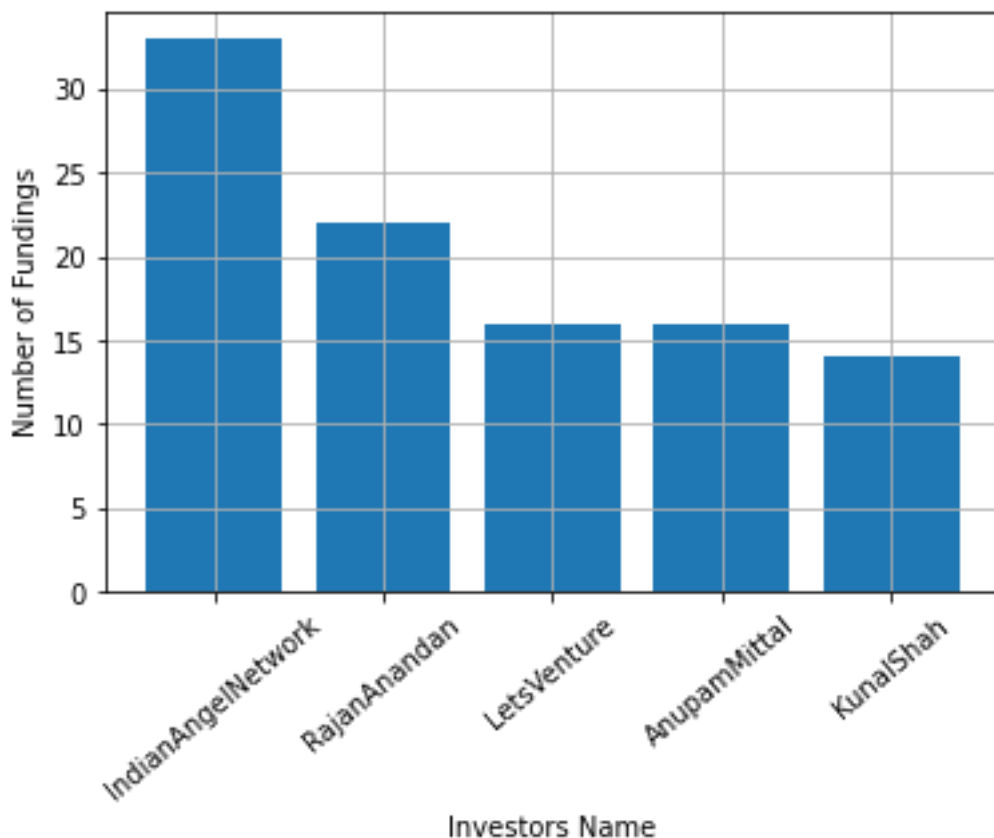
- i. Seed Funding or Crowd Funding
- ii. Private Equity

Code Logic and Justification : These two problems are almost equivalent to problem 3. We just have to do a few extra steps before we can apply the algorithm used in problem 3. Firstly we need to include Investors Name, Start-up name and Investment type in our data frame. Then, using the .replace

functionality, spelling errors are handled in the Investment Types. Using Boolean Indexing we edit the data frame to only those entries that have the Investment Type as Seed Funding or Crowd Funding as in problem 4 or Private Equity as in problem 5. After these manipulation we can use the exact same algorithm in problem 3 that gives us the Top 5 investors and the Bar Plot.

Conclusion and Bar Plot (Problem 4) : The list of the top 5 investors as well as the bar plot is given below :

```
Indian Angel Network 33
Rajan Anandan 22
Lets Venture 16
Anupam Mittal 16
Kunal Shah 14
```



Conclusion and Bar Plot (Problem 5) : The list of the top 5 investors as well as the bar plot is given below :

Sequoia Capital 45
Accel Partners 42
Kalaari Capital 35
Blume Ventures 27
SAIF Partners 23

