**Project: Case Study-2**

This is a Project Submission by Gurpreet Singh, email – [singh.gurpreet513@gmail.com](mailto:singh.gurpreet513@gmail.com). This pdf file contains the answers and the justification. The code in the .ipynb file needs to run to get the desired answer output.

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.

Answer – Bangalore has the highest number of fundings equal to 635

Justification – The code has the following output:

Bangalore 635

Mumbai 449

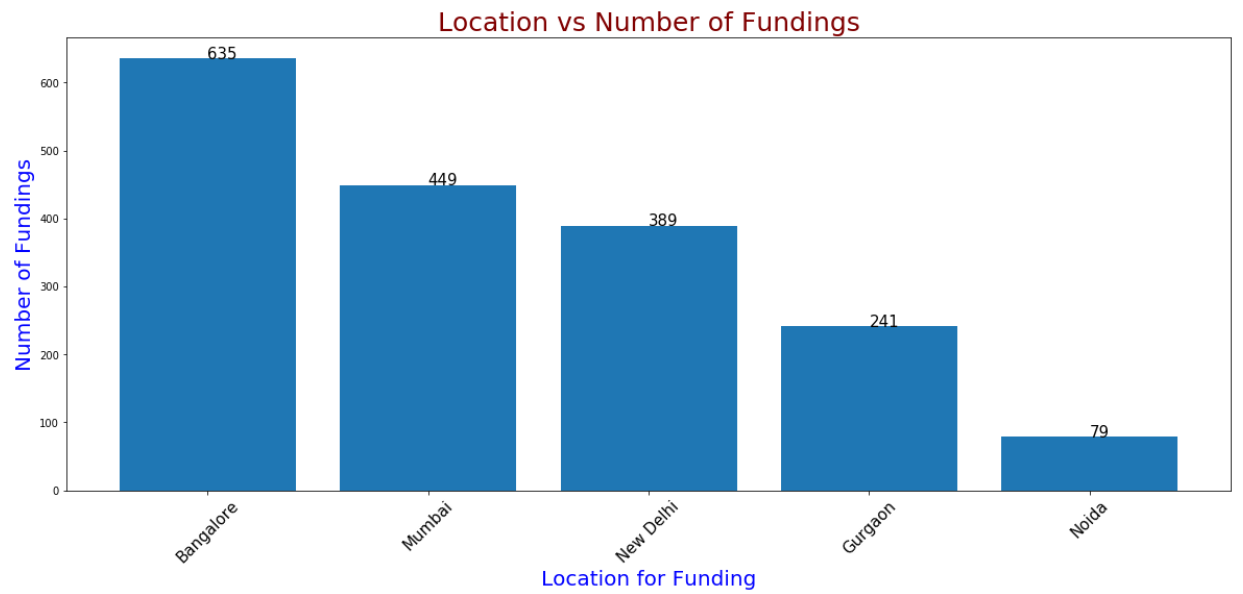
New Delhi 389

Gurgaon 241

Noida 79

We can clearly see that Bangalore has 635 number of investments(highest), then we have Mumbai with 449 investments, New Delhi with 389 investments, Gurgaon with 241 investments, and Noida had 79 investments(least among these city preferences)

On a Bar Graph-



Now for the outcome that we have reached to, first we use our Pandas library and use the read\_csv() function to read our startup.csv file into a variable name say “df”. After Dropping the City Location with NaN, we change all the values of CityLocation column by ‘applying’ the separateCity function that we defined. After getting all the CityLocations corrected(first problem was to handle the city locations which were a combination of 2 places), Now we can replace the CityLocation “Delhi” by “New Delhi” and also handle the case correctness for places like “bangalore” and correcting them to “Bangalore”. Then we will define a variable say “location” and give it the values of the CityLocation column i.e. equal to df.CityLocation. Now we create an empty dictionary say “d” and we iterate over the location to count the number of occurrences for all the cities that we want to consider. d.get() function will return us 0 if that location according to the condition is not present, otherwise if present it will give us the count and we keep adding 1 with each occurrence. Now all the main steps have been covered. We have all the cities(our 5 cities) and their count of investments. Now we are only left with plotting the graph. For that first we will put x= list(d.keys()) and y = list(d.values()), and we are using the list function so that we can also iterate over x and y to print the cities and the count in the end. We use matplotlib.pyplot library as plt to plot the bar graph using the plt.bar() function. plt.rcParams[‘figure.figsize’] helps us to adjust the size for the graph. Title, xlabel, ylabel, xticks and the necessary texts using a loop have been showed on the graph too. In the end using a loop we will simply print all the cities and number of fundings respectively.

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.

Answer – The top 5 investors are Sequoia Capital(64 investments), Accel Partners(53 investments),

Kalaari Capital(44 investments), SAIF Partners(41 investments), Indian Angel Network(40 investments)

Justification – The code has the following output:

Sequoia Capital 64

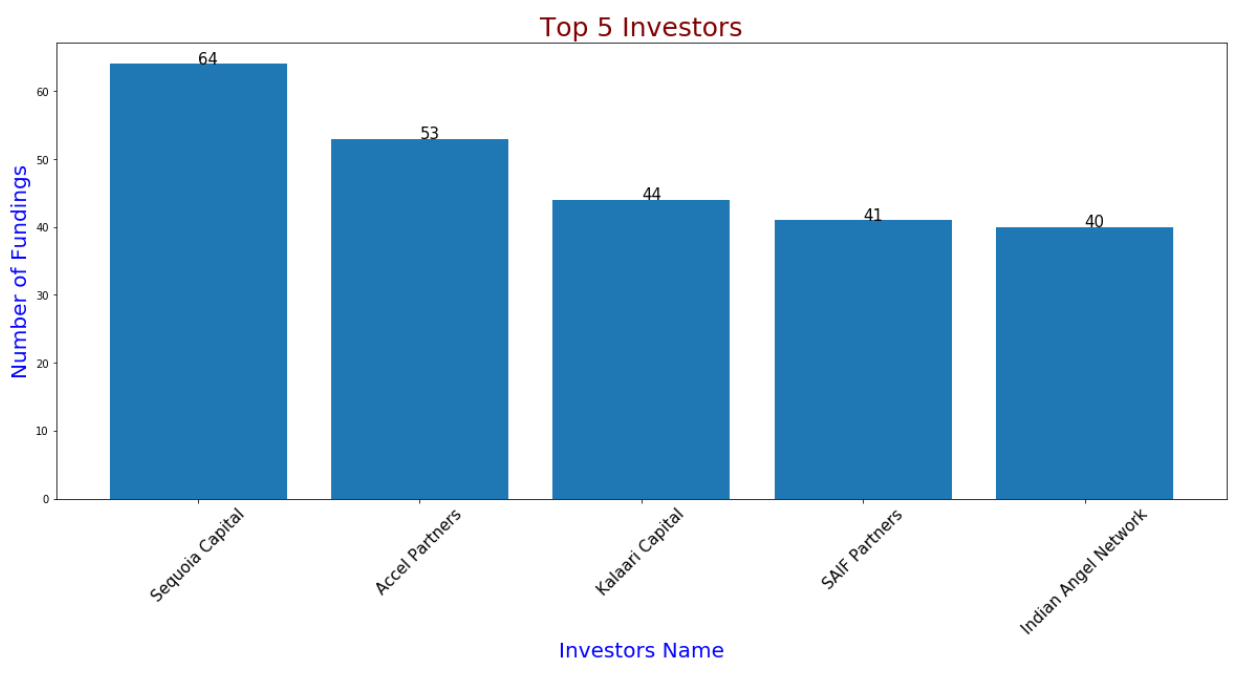
Accel Partners 53

Kalaari Capital 44

SAIF Partners 41

Indian Angel Network 40

On a Bar Graph-



The code output simply tells us the top 5 investors that we have(according to the number of investments they make).

Now for the outcome that we have reached to, first we use our Pandas library and use the read\_csv() function to read our startup.csv file into a variable name say “df”. Then we will drop the NaN values for InvestorNames column because those will not be required. Now as the InvestorNames column has a

few difficulties like that there can be a number of investors in a single row. Now our need is to count

the number of investments made by each investor, so we will simply call a function lets say the name of the function is createDictionary and the argument passed to this function will by df.InvestorsName.

Now comes the part of defining our function. for our function, we will first create an empty dictionary,

and then we will iterate over each row for the parameter passed to the function, incase there is only

one investor name(so it will not have commas in it) we will check if that investor name is present(if it is or maybe is not that can be handles by out get function and we can keep incrementing the count),

else if we multiple investor names we will simply split the string, turn it into a list and increment over

the list values to add those names or to increment the count of those investor names into our

dictionary. After we have all the investor names as the keys and the number of investments as the

values in our dictionary, we will simply return the dictionary to the function call. Now the return

dictionary is stored in a variable say “dictionary”. Now using our DataFrame function of pandas library, we will use the dictionary keys and values to form a DataFrame. We will use the sort function and sort it by [0] because 0 is our column name and ascending will be equal to False so that we have the

highest number of investors starting from the top. Now simply using a loop we will print the names of

the top 5 investors according to the number of their investments. And then plot bar graph.

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.

Answer – The top 5 investors with investments in different companies(each company count taken as one) are Indian Angel Network(with investment in 33 different Startups), followed by Ratan Tata(with investment in 27 different Startups), then Kalaari Capital(with investment in 25 different Startups), then

Sequoia Capital(with investment in 18 different Startups) and on position 5 we have Accel Partners(with investment in 18 different Startups like Sequoia Capital)

Justification – The code has the following output:

Indian Angel Network 33

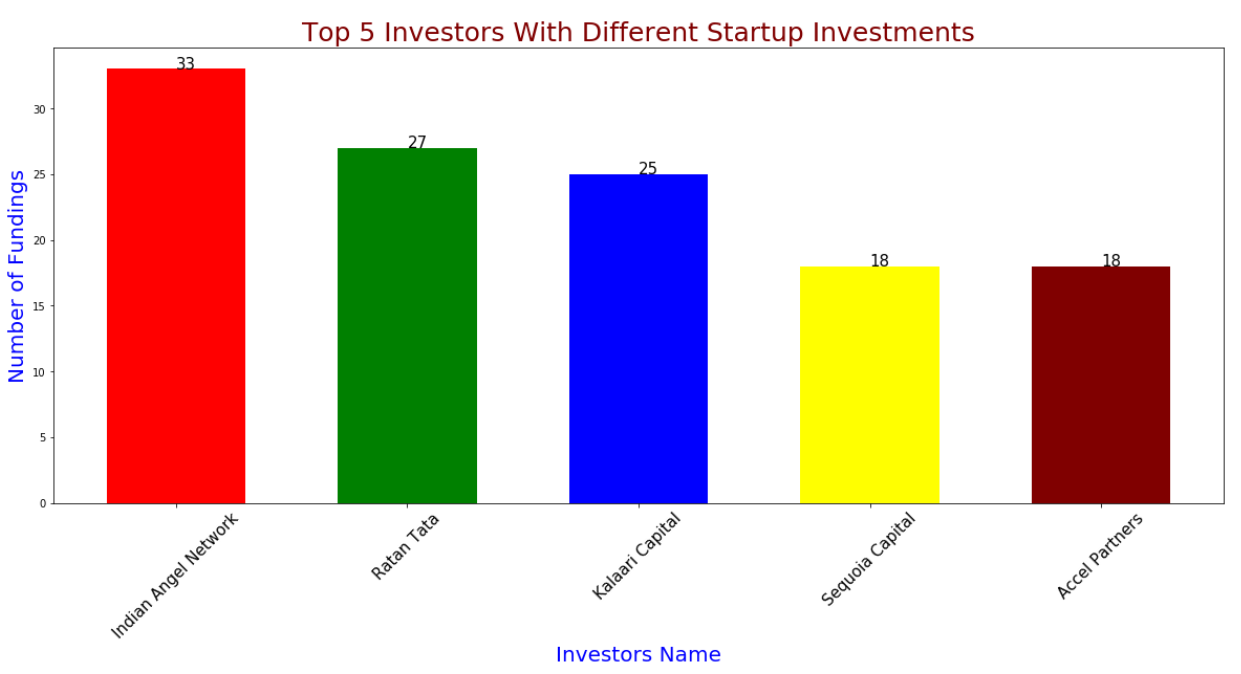
Ratan Tata 27

Kalaari Capital 25

Sequoia Capital 18

Accel Partners 18

On a Bar Graph-



The code output simply tells us the top 5 investors that we have(according to the number of

investments they make in different companies).

Now for the outcome that we have reached to, first we use our Pandas library and use the read\_csv() function to read our startup.csv file into a variable name say “df”. After that we drop the rows

containing NaN values in InvestorsName. Now that there are errors in the startup names, we will

simply just take care of the main start up companies that we have with us - Ola, Flipkart, Oyo and

Paytm. After that our aim is to create a dictionary that will have the keys as the names of investors

and the values as the companies in which they have invested. For that we will call a function

“createDictionary” that we will define passing it the arguments as (df.InvestorsName,df.StartupName).

Now let’s define our function createDictionary. Now we will declare the variable “index” which will

store the index values for our Investor Names columns so that we can iterate over the indexes. Now we will create an empty dictionary say “dictionary “. After that we will have a few set of conditions,

for a 1)single investor name and for 2)multiple investor names in our iteration. The logic for both 1)

and 2) is same, we will first check whether our investor name is present in the dictionary, if yes then

we will check if the startup name for that investor is present or not(if not we will append the value to

the investors name), and if no then we will create an empty list for that investors name and append

the value of the startup name to the investor names empty list. This was the main logic uptil here to

create a dictionary with keys as investors name and the values will be lists containing all the unique

startups they have invested in. Now we will simply return this dictionary to its function calling. After

that all the values of this dictionary will be changed to the length of the value for each key(which was earlier a list). Now we will create a DataFrame using the keys and values and then use the

sort\_values function to sort them according to column 0 and in descending order. Now the only task

left is to print the investor names and the number of investments which can be handled by the loops.

And in the end we can plot the bar graph on the same lines like in the question 1. Bar Graph will be

Plotted as investors names against no. of investments.

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.

Answer – The top 5 investors with investments in different companies which are of the Investment

Types- Seed Funding or Crowd Funding are 1) Indian Angel Network(31 investments), 2) Group of

Angel Investors(14 investments), 3) Kae Capital(10 investments), 4) YouWeCan Ventures(9 investments), 5) Ratan Tata(8 investments)

Justification – The code has the following output:

Indian Angel Network 31

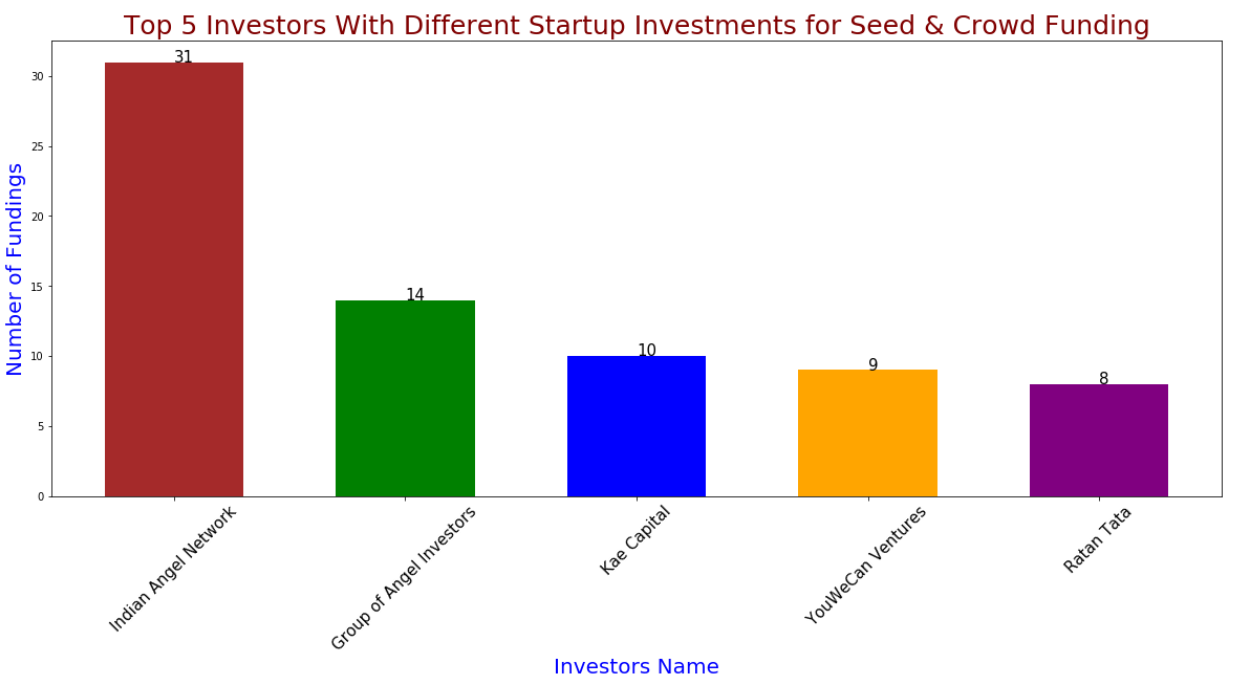
Group of Angel Investors 14

Kae Capital 10

YouWeCan Ventures 9

Ratan Tata 8

On a Bar Graph-



The code output simply tells us the top 5 investors that we have(according to the number of

investments they make in different companies for the Investment Types- Seed Funding and Crowd

Funding).

Now for the outcome that we have reached to, first we use our Pandas library and use the read\_csv() function to read our startup.csv file into a variable name say “df”. Now we will simply replace the

errors in Invest Type and and Startup Name column which can be visualised by looking at the unique

values for each column. Now we will create a copy of df into lets say “df1”. After that we drop the rows

containing NaN values in InvestorsName for df1. After that our aim is to create a dictionary that will

have the keys as the names of investors and the values as the companies in which they have

invested. For that we will call a function “createDictionary” that we will define passing it the

arguments as (df1.InvestorsName,df1.StartupName). Now let’s define our function createDictionary. Now we will declare the variable “index” which will store the index values for our Investor Names

columns so that we can iterate over the indexes. Now we will create an empty dictionary say

“dictionary “. After that we will have a few set of conditions, for a 1)single investor name and for

2)multiple investor names in our iteration. The logic for both 1) and 2) is same, we will first check

whether our investor name is present in the dictionary, if yes then we will check if the startup name for that investor is present or not(if not we will append the value to the investors name), and if no then we will create an empty list for that investors name and append the value of the startup name to the

investor names empty list. This was the main logic uptil here to create a dictionary with keys as

investors name and the values will be lists containing all the unique startups they have invested in.

Now we will simply return this dictionary to its function calling. After that all the values of this

dictionary will be changed to the length of the value for each key(which was earlier a list). Now we will create a DataFrame using the keys and values and then use the sort\_values function to sort them

according to column 0 and in descending order. Now the only task left is to print the investor names

and the number of investments which can be handled by the loops and the investor names which are undisclosed will not be printed by using appropriate conditions. In the end, we can plot the bar graph, which is optional as it is not mentioned in the question.

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.

Answer – The top 5 investors with investments in different companies which are of the Investment

Type- Private Equity are 1) Kalaari Cpital(19 investments), 2) Ratan Tata(19 investments), 3) Tiger

Golbal(17 investments), 4) Sequoia Capital(15 investments) and 5) Accel Partners(15 investments)

Justification – The code has the following output:

Kalaari Capital 19

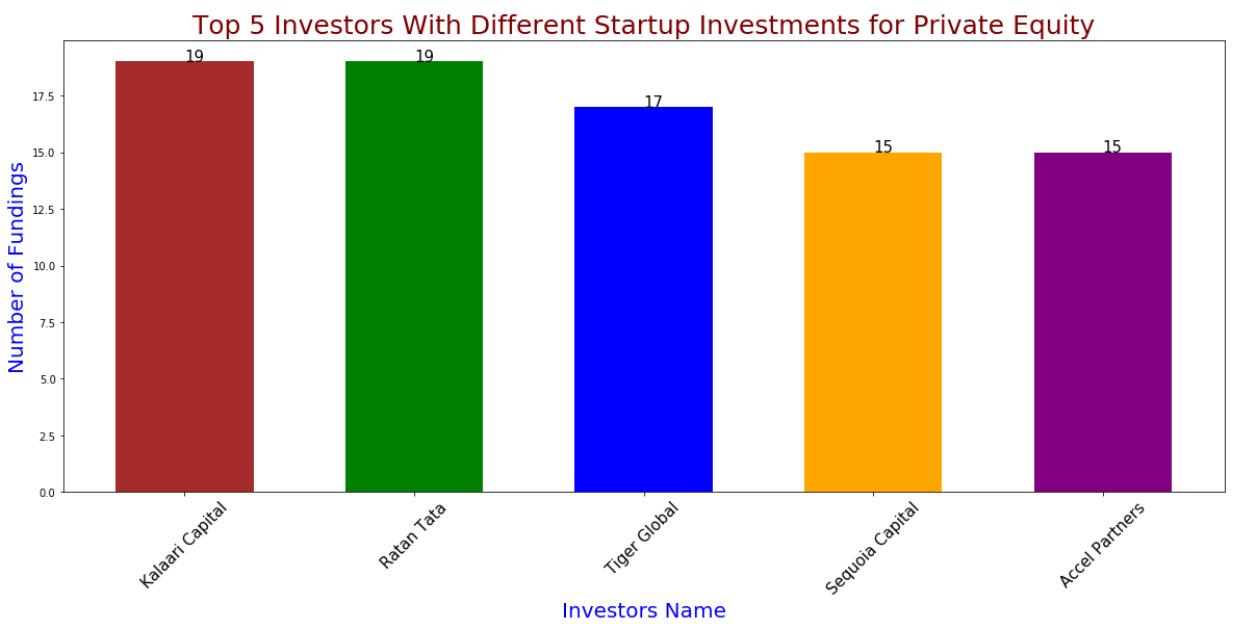
Ratan Tata 19

Tiger Global 17

Sequoia Capital 15

Accel Partners 15

On a Bar Graph-



The code output simply tells us the top 5 investors that we have(according to the number of

investments they make in different companies for the Investment Type- Private Equity)

Now for the outcome that we have reached to, first we use our Pandas library and use the read\_csv() function to read our startup.csv file into a variable name say “df”. Now we will simply replace the

errors in Invest Type and and Startup Name column which can be visualised by looking at the unique

values for each column. Now we will create a copy of df into lets say “df1”. After that we drop the rows

containing NaN values in InvestorsName for df1. After that our aim is to create a dictionary that will

have the keys as the names of investors and the values as the companies in which they have

invested. For that we will call a function “createDictionary” that we will define passing it the arguments as (df1.InvestorsName,df1.StartupName). Now let’s define our function createDictionary. Now we will declare the variable “index” which will store the index values for our Investor Names columns so that we can iterate over the indexes. Now we will create an empty dictionary say “dictionary “. After that

we will have a few set of conditions, for a 1)single investor name and for 2)multiple investor names in our iteration. The logic for both 1) and 2) is same, we will first check whether our investor name is

present in the dictionary, if yes then we will check if the startup name for that investor is present or not(if not we will append the value to the investors name), and if no then we will create an empty list for

that investors name and append the value of the startup name to the investor names empty list. This was the main logic uptil here to create a dictionary with keys as investors name and the values will be lists containing all the unique startups they have invested in. Now we will simply return this dictionary

to its function calling. After that all the values of this dictionary will be changed to the length of the

value for each key(which was earlier a list). Now we will create a DataFrame using the keys and

values and then use the sort\_values function to sort them according to column 0 and in descending order. Now the only task left is to print the investor names and the number of investments which can be handled by the loops and the investor names which are undisclosed will not be printed by using

appropriate conditions. In the end, we can plot the bar graph, which is optional as it is not mentioned

in the question.