Indian Startups

Project - Case Study 2

Q1 Approach:

I start with importing the files and reading the data. I drop all the NA values in “City Location” because that will help us categorise the startups based on the city. Then I update the errors in the names of important cities and select Indian location for startups with multiple locations, all of this is done in the updateCity function which is applied to the “CityLocation” column. Once my city data is updated, I then select the possible cities and only keep their data in the data frame, and all other cities are discarded. This is done in the chooseCity function. Next I can just get the Number of times all the cities are present in the data frame using value_counts function. And this is all the date that we need to plot the graph and draw an inference.

Q2 Approach:

I start with importing the files and reading the data. I drop all the NA values in “InvestorsName” and drop all the ‘undisclosed investors’ (and all its variations) in the removeUndisclosed function. Once that is done, I create a dictionary to store investments done by each investor. The dictionary is filled in the getInvestors function, which is called on all the rows in InvestorsName column. Once the dictionary is filled, we can convert it into a data frame and sort the data frame. Then all we have to do is to show the top 5 investors from the data frame, i.e. top 5 rows.

Q3 Approach:

I start with importing the files and reading the data. I drop all the NA values in “InvestorsName” and drop all the ‘undisclosed investors’ (and all its variations) in the removeUndisclosed function. Then I rectify all the errors in the startup names. Then I create a dictionary to store all the companies someone has invested in, i.e. it is structurally similar to {InvestorName : [‘Startup1’, ‘Startup2’, …] }. This is filled in the getUniqueInvestors function, which is applied on all the rows. Once the dictionary is filled, We can convert the InvestorName and the length on unique investments into another data frame, and sort it. Once it is sorted, we can show the top 5 rows.

Q4 Approach:

I start with importing the files and reading the data. I drop all the NA values in “InvestorsName” and drop all the ‘undisclosed investors’ (and all its variations) in the removeUndisclosed function. After that I rectify all the errors in “Investment Type” column and then update the data frame to contain only those rows which have Investment type as “Seed Funding” or “Crowd Funding” . Then I rectify all the errors in the startup names. Then I create a dictionary to store all the companies someone has invested in, i.e. it is structurally similar to {InvestorName : [‘Startup1’, ‘Startup2’, …] }. This is filled in the getUniqueInvestors function, which is applied on all the rows. Once the dictionary is filled, We can convert the InvestorName and the length on unique investments into another data frame, and sort it. Once it is sorted, we can show the top 5 rows.

Q5 Approach:

I start with importing the files and reading the data. I drop all the NA values in “InvestorsName” and drop all the ‘undisclosed investors’ (and all its variations) in the removeUndisclosed function. After that I rectify all the errors in “Investment Type” column and then update the data frame to contain only those rows which have Investment type as “Private Equity” . Then I rectify all the errors in the startup names. Then I create a dictionary to store all the companies someone has invested in, i.e. it is structurally similar to {InvestorName : [‘Startup1’, ‘Startup2’, …] }. This is filled in the getUniqueInvestors function, which is applied on all the rows. Once the dictionary is filled, We can convert the InvestorName and the length on unique investments into another data frame, and sort it. Once it is sorted, we can show the top 5 rows.