Case Study

Startup funding dataset

by Abhinav Rawal

Abstract

This case study involves studying the dataset about startup fundings & gathering various insights about industries, investors, preferred locations and other trends using libraries like *numpy, pandas* as well as *matplotlib* for filtering and visualizing data respectively.

Contents

- Importing tools, libraries and unpacking data
- Data cleaning and analysis
- Section 1

  The respective frequencies of investments in the preferred cities namely Bangalore, Mumbai, and NCR(New Delhi, Noida and Gurgaon) visualized with a histogram.
- Section 2

  The top five investors based on frequency of investments
- Section 3

  The top five investors based on frequency of investments in unique companies.
- Section 4

  Top five investors based on unique seed or crowd funding investments
- Section 5

  Top five investors based on unique private equity investments

Importing libraries

Numpy and Pandas are imported to store, iterate, manipulate and search in the dataset. The module pyplot is imported from the Matplotlib library to create graphs of the data extracted.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Unpacking & cleaning data

Firstly, the *read_csv* function from pandas is used to read the dataset skipping the initial spaces in it and storing the file as a *dataframe* object in the variable *fund.*

We create a copy of this object in the variable *df* to perform manipulation. Then using the *fillna* function all the *NaN* entries in the dataset are replaced with the empty string '' character.

The *head()* command showed the initial entries in the dataset which helped visualize the form, type and structure of data without actually viewing the entire dataset.

```
fund = pd.read_csv('startup_funding.csv', skipinitialspace= True)
df = fund.copy()
df.fillna('', inplace = True)
df.head()
```

SNo	Date	StartupName	IndustryVertical	SubVertical	CityLocation	InvestorsName
0	01/08/2017	TouchKin	Technology	Predictive Care Platform	Bangalore	Kae Capital
1	02/08/2017	Ethinos	Technology	Digital Marketing Agency	Mumbai	Triton Investment Advisors
2	02/08/2017	Leverage Edu	Consumer Internet	Online platform for Higher Education Services	New Delhi	Kashyap Deorah, Anand Sankeshwar, Deepak Jain,...

Section - 1

The number of fundings in Mumbai, Bangalore and NCR.

To find all the cities and the forms they maybe present *unique()* function is used on column *CityLocations* and the following observations are made.

The above mention cities might be present with lowercase starting letters, New Delhi is also present as Delhi and cities might also be clubbed using the forward-slash symbol.

```
#Analysis of City Locations
sorted(df['CityLocation'].unique())
```

Now that we know the errors in city names from observation, a simple solution to check the preferred city names would be to find the lowercase string in the lowercase city location and then storing it in a frequency bucket.

fundings is the frequency bucket initialized with zeros. The *CityLocations* column is iterated, if the preferred city is found in the *city* string frequency bucket for that city is incremented.

```
#Data
cities = np.array(['Bangalore', 'Mumbai',
      'New Delhi', 'Gurgaon', 'Noida'])
fundings = np.zeros(5,dtype = int)

for city in df.CityLocation:
    city = city.lower()
    if 'bangalore' in city:
        fundings[0] += 1
    elif 'mumbai' in city:
        fundings[1] += 1
    elif 'delhi' in city:
        fundings[2] += 1
    elif 'gurgaon' in city:
        fundings[3] += 1
    elif 'noida' in city:
        fundings[4] += 1
```

Using the cities array which stores the names of the preferred cities and the funding frequency map we can plot the histogram keeping frequency count on the y axis and the city names on the x axis.

First we need to extract the data from the map and zip it the cities array using the zip function which is composed by sorted function to reverse sort the zipped array, which is again composed by the zip* function to unzip the sorted contents into y and x axis.

Figure and axis of the plot are obtained in the next line, a horizontal grid is then made and is kept below the histogram plot. Similarly the title as well the label for x and y axis are also set with the given parameters.

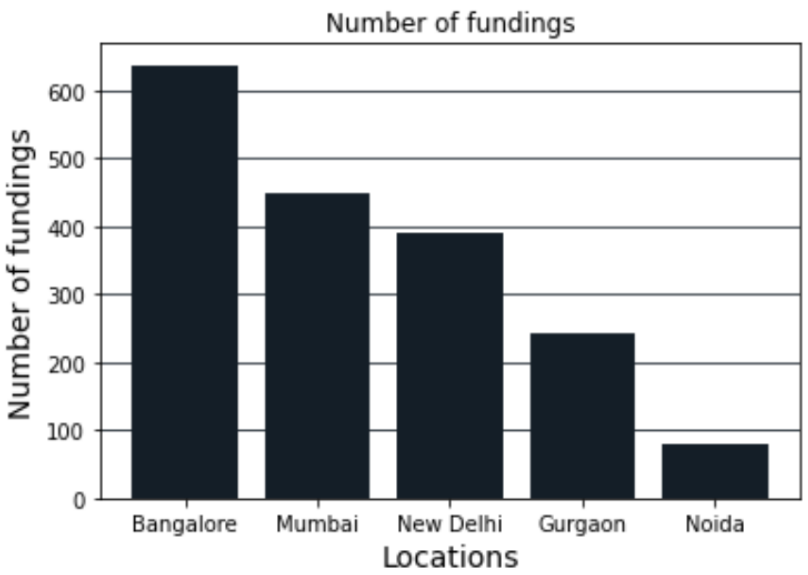
The x and y array obtained in the first line are then passed as parameters to form the histogram, the histogram plot is ready and is shown as below, clearly indicating that Bangalore has the highest frequency of fundings.

```
#Graph
y,x = zip(*sorted(list( zip(fundings,cities)), reverse =True))

fig,ax = plt.subplots()
ax.grid(color = '#141E27', axis = 'y')
ax.set_axisbelow(True)
ax.set_title(label = 'Number of fundings')
ax.bar(x,y, color = '#141E27')

ax.set_xlabel('Locations', fontsize = 14)
ax.set_ylabel('Number of fundings', fontsize = 14)

plt.show()
print(x[0],'had the maximum number of fundings.')
```



Section - 2

The top five investors based on frequency of investments

The InvestorsName column in analyzed for the unique names and undisclosed investors that maybe present using the unique function and running a loop to check for its form in the dataset.

```
#Analysis of Investor names
x = df.InvestorsName.unique()
for name in x:
    if('undisclosed' in name.lower()):
        print(name)
```

A frequency map for the investors is initialized and the InvestorsName column is iterated over; if a non-empty string is present, it's split with delimiter as a comma, the spaces are stripped off each split, if the name split thus obtained is non-empty and disclosed the frequency bucket for the investor is incremented.

```
#Data
investors = {}
for names in df.InvestorsName:
    if names == '':continue
    for name in names.split(','):
        name = name.strip()
        if name == '':continue
        if 'Undisclosed' in name or 'undisclosed' in name:continue

        if name in investors:
            investors[name] += 1
        else:
            investors[name] = 1
```

The data from the map is extracted using the DataFrame function and the top five investors are found using the nlargest function

```
freq = pd.DataFrame(investors.items(), columns = ['Name','Frequency'])
for investor in freq.nlargest(5,'Frequency').Name : print(investor)
```

```
Sequoia Capital
Accel Partners
Kalaari Capital
SAIF Partners
Indian Angel Network
```

Section - 3

The top five investors based on frequency of unique investments

The StartupName column in analyzed for the errors in the major startups using the unique function and running a loop to check for its form in the dataset by finding the lowercase names in the string of the column entries.

The duplicates for the given companies are present in the following forms-

Oyo	Oyo Rooms, OyoRooms, Oyorooms, OYO Rooms
Flipkart	Flipkart.com
Ola	Ola Cabs, Olacabs
Paytm	Paytm Marketplace

```
#Analysis of Startup Names
x = df.StartupName.unique().astype(str)
startups = ['oyo', 'flipkart', 'ola','paytm']
for j in startups:
    for i in x:
        if j in i.lower():print(i)
```

A frequency map for the investors is initialized as freq as well as a dictionary of investors containing key: value pair as investor name and the set of companies they have invested in.

For each entry in InvestorsName and StartupName, firstly, the company name errors are corrected (if any), then the name string is split as done previously, if the company is not present in the investors set it is added and the respective frequency bucket is updated for the investor name otherwise not.

```
freq,investors = {},{}
for names,company in zip(df.InvestorsName,df.StartupName):
    if names == '':continue
    if company == '':continue

    if   company == 'Flipkart.com':company = 'Flipkart'
    elif company == 'Paytm Marketplace' :company = 'Paytm'
    elif company == 'Ola Cabs' or company == 'Olacabs':company = 'Ola'
    elif company in {'Oyo Rooms','OyoRooms','Oyorooms','OYO Rooms'}:
        company = 'Oyo'
```

The split of names, extraction of data as well as finding the top five investors are done in the similar way as in the previous section.

```
    for name in names.split(','):
        name = name.strip()
        if name == '': continue
        if 'Undisclosed' in name or 'undisclosed' in name:continue

        if name in freq:
            if company not in investors[name]:
                investors[name].add(company)
                freq[name] += 1

        else:
            investors[name] = {company}
            freq[name] = 1

f = pd.DataFrame(freq.items(), columns = ['Name','Frequency'])
for investor in f.nlargest(5,'Frequency').Name : print(investor)
```

```
Sequoia Capital
Accel Partners
Kalaari Capital
Indian Angel Network
Blume Ventures
```

Section - 4

Top five investors based on unique seed or crowd funding investments

The InvestmentType column in analyzed for the errors in the types of investments using the unique function

The duplicates for the given investment types are present in the following forms-

Private Equity	PrivateEquity
Seed Funding	SeedFunding
Crowd funding	Crowd Funding

```
#Analysis of Investment Types
df.InvestmentType.unique()
```

A frequency map for the investors as freq as well as a dictionary of investors are initialized as before.

For each entry in InvestorsName. StartupName and InvestmentType, if the funding type is not empty or based on Private equity or Debit funding the errors in company names as well as Investment types are corrected and the frequency maps are created as before and in the similar way the top five investors are printed after extracting them from the map.

```
freq,investors = {},{}
for names,company,funding in zip(df.InvestorsName,
                    df.StartupName,df.InvestmentType):

    if names == '':continue
    if company == '':continue
    if funding in {'','Private Equity',
        'PrivateEquity', 'Debt Funding'}:continue

    if   company == 'Flipkart.com':company = 'Flipkart'
    elif company == 'Paytm Marketplace' :company = 'Paytm'
    elif company == 'Ola Cabs' or company == 'Olacabs':
        company = 'Ola'
    elif company in {'Oyo Rooms','OyoRooms',
                     'Oyorooms','OYO Rooms'}:company = 'Oyo'
```

```
    for name in names.split(','):
        name = name.strip()
        if name == '': continue
        if 'Undisclosed' in name or 'undisclosed' in name:continue

        if name in freq:
            if company not in investors[name]:
                investors[name].add(company)
                freq[name] += 1

        else:
            investors[name] = {company}
            freq[name] = 1

f = pd.DataFrame(freq.items(), columns = ['Name','Frequency'])
for investor in f.nlargest(5,'Frequency').Name : print(investor)
```

```
Indian Angel Network
Rajan Anandan
LetsVenture
Anupam Mittal
Kunal Shah
```

Section - 5

Top five investors based on unique private equity investments

Since, it is needed to find the top investors based unique private equity investments rather than seed and crowd funding (as done in the previous section) we only need to change the funding condition as marked in the code below.

```
freq,investors = {},{}
for names,company,funding in zip(df.InvestorsName,
                    df.StartupName,df.InvestmentType):

    if names == '':continue
    if company == '':continue
    if funding not in {'Private Equity','PrivateEquity'}:continue

    if   company == 'Flipkart.com':company = 'Flipkart'
    elif company == 'Paytm Marketplace' :company = 'Paytm'
    elif company == 'Ola Cabs' or company == 'Olacabs':company = 'Ola'
    elif company in {'Oyo Rooms','OyoRooms','Oyorooms','OYO Rooms'}:
        company = 'Oyo'

    for name in names.split(','):
        name = name.strip()
        if name == '': continue
        if 'Undisclosed' in name or 'undisclosed' in name:continue

        if name in freq:
            if company not in investors[name]:
                investors[name].add(company)
                freq[name] += 1

        else:
            investors[name] = {company}
            freq[name] = 1

f = pd.DataFrame(freq.items(), columns = ['Name','Frequency'])
for investor in f.nlargest(5,'Frequency').Name : print(investor)
```

```
Sequoia Capital
Accel Partners
Kalaari Capital
Blume Ventures
SAIF Partners
```