Data Analysis Project

A company ABC wants to invest in startups and other companies.The CEO of the company wants to understand the global trends in investments so that he can make the investment effectively.

Business Understanding

ABC has two major constraints for investments:

- It wants to invest between 5 to 15 million USD per round of investment
- It only wants to invest in English-speking countries (Countries with english as one of their one of the official languages.).

The objective is to find the best sectors, countries, and suitable investment type for making investments. Here best means where the number of investors is greater.

Data Understanding

The data is real world data taken from crunchbase.com and contains three files

1. Mapping file : Contains main eight sectors and their subsectors.
2. Companies file : Contains different companies from all over the world with their basic information such as sector,country of origin etc.
3. Rounds file : Contains all the sub sectors along with companies details.

Methodology

First we have to load the data into our IPython notebook. Before that we need our modules.
We are going to use :

- Pandas
- Numpy
- Matplotlib
- Seaborn

This is data analysis and cleaning project so we are not going to use scikit learn.
First we have to clean the data therefore we will look into the data and as much information we can.

We used companies.head() to see the first rows of the companies data file.
Here we have 10 columns.
As you can see the permalinks contains different case letters, so we will need to convert them into one case. Therefore we converted them into smaller case.
Similarly, we peeked into rounds data set.

```
[12]: rounds['company_permalink'] = rounds['company_permalink'].str.lower()
      rounds.head()
```

[12]:

	company_permalink	funding_round_permalink	funding_round_type	funding_round_code	funded_at	raised_amount_usd
0	/organization/-fame	/funding-round/9a01d05418af9f794eebff7ace91f638	venture	B	05-01-2015	10000000.0
1	/organization/-qounter	/funding-round/22dacff496eb7acb2b901dec1dfe5633	venture	A	14-10-2014	NaN
2	/organization/-qounter	/funding-round/b44fbb94153f6cdef13083530bb48030	seed	NaN	01-03-2014	700000.0
3	/organization/-the-one-of-them-inc-	/funding-round/650b8f704416801069bb178a1418776b	venture	B	30-01-2014	3406878.0
4	/organization/0-6-com	/funding-round/5727accaeaa57461bd22a9bdd945382d	venture	A	19-03-2008	2000000.0

We have company permalinks in rounds data too. So we can first check whether they all are the same as in companies dataset.

```
[14]: rounds.loc[~rounds['company_permalink'].isin(companies['permalink']), :]
```

[14]:

	company_permalink	funding_round_permalink	funding_round_type	funding_round_code	funded_at	raised_amount_usd
29597	/organization/e-cãbica	/funding-round/8491f74869e4fe8ba9c378394f8fbdea	seed	NaN	01-02-2015	NaN
31863	/organization/energystone-games-çµç³æ„æ	/funding-round/b89553f3d2279c5683ae93f45a21cfe0	seed	NaN	09-08-2014	NaN
45176	/organization/huizuche-com-æ ç§ÿè½¦	/funding-round/8f8a32dbeeb0f831a78702f83af78a36	seed	NaN	18-09-2014	NaN
58473	/organization/magnet-tech-ç£ç³ç§æ	/funding-round/8fc91fbb32bc95e97f151dd0cb4166bf	seed	NaN	16-08-2014	1625585.0
101036	/organization/tipcat-interactive-æ²èÿä¿¡æ¯ç...	/funding-round/41005928a1439cb2d706a43cb661f60f	seed	NaN	06-09-2010	NaN
109969	/organization/weiche-tech-åè½¦ç§æ	/funding-round/f74e457f838b81fa0b29649740f186d8	venture	A	06-09-2015	NaN
113839	/organization/zengame-ç¦æ„ç§æ	/funding-round/6ba28fb4f3eadf5a9c6c81bc5dde6cdf	seed	NaN	17-07-2010	NaN

Here we use isin function to check which permalinks are not in the intersection of both the dataset.
As you can see there are some weird characters in the company-permalink.
This is because of the python's encoding. As the companies dataset is .txt file so it may have done some encoding differently.
To fix this we have to encode the file to utf-8 and then decode it to ascii.
This way we can fix this problem. But you want to see the proper explanation go to this stack overflow link.
https://stackoverflow.com/questions/45871731/removing-special-characters-in-a-pandas-dataframe.

```
[79]: rounds['company_permalink'] = rounds.company_permalink.str.encode('utf-8').str.decode('ascii', 'ignore')
      rounds.loc[~rounds['company_permalink'].isin(companies['permalink']), :]
```

```
6]: companies.head()
```

6]:

	permalink	name	homepage_url	category_list	status	country_code	state_code	region	city	founded_at
0	/Organization/-Fame	#fame	http://livfame.com	Media	operating	IND	16	Mumbai	Mumbai	NaN
1	/Organization/-Qounter	:Qounter	http://www.qounter.com	Application Platforms|Real Time|Social Network...	operating	USA	DE	DE - Other	Delaware City	04-09-2014
2	/Organization/-The-One-Of-Them-Inc-	(THE) ONE of THEM,Inc.	http://oneofthem.jp	Apps|Games|Mobile	operating	NaN	NaN	NaN	NaN	NaN
3	/Organization/0-6-Com	0-6.com	http://www.0-6.com	Curated Web	operating	CHN	22	Beijing	Beijing	01-01-2007
4	/Organization/004-Technologies	004 Technologies	http://004gmbh.de/en/004-interact	Software	operating	USA	IL	Springfield, Illinois	Champaign	01-01-2010

```
7]: companies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 66368 entries, 0 to 66367
Data columns (total 10 columns):
permalink       66368 non-null object
name            66367 non-null object
homepage_url    61310 non-null object
category_list   63220 non-null object
status          66368 non-null object
country_code    59410 non-null object
state_code      57821 non-null object
region          58338 non-null object
city            58340 non-null object
founded_at      51147 non-null object
dtypes: object(10)
memory usage: 5.1+ MB
```

We will use the same technique on the companies dataset.
I think now the issue with the encoding is resolved.Now we can move on with our data cleaning.

It’s time to check for the missing values and it seems that we have a lot of them.

```
[6]: companies.isnull().sum()
```

```
[6]: permalink            0
     name                 1
     homepage_url      5058
     category_list     3148
     status               0
     country_code      6958
     state_code        8547
     region            8030
     city              8028
     founded_at       15221
     dtype: int64
```

This is for companies dataset. Let’s check for the rounds dataset also.

```
rounds.isnull().sum()
```

```
company_permalink              0
funding_round_permalink        0
funding_round_type             0
funding_round_code         83809
funded_at                      0
raised_amount_usd          19990
dtype: int64
```

Since there is no missing values in the permalink or comapany_permalink columns let’s merge two datasets into one. This way it will be easier to clean the data.

```
[8]: master = pd.merge(companies, rounds, how="inner", left_on="permalink", right_on="company_permalink")
     master.head()
```

Now we will drop the company_permalink because the permalink and company_permalink are same.
We will again check for the null values in our main dataset(let’s call our new dataset as master) using isnull() and sum() function.
Looking at the data funding_round_code, hompage_url,founded_at,state_code,region and city are needed according to our business objective so we will drop these

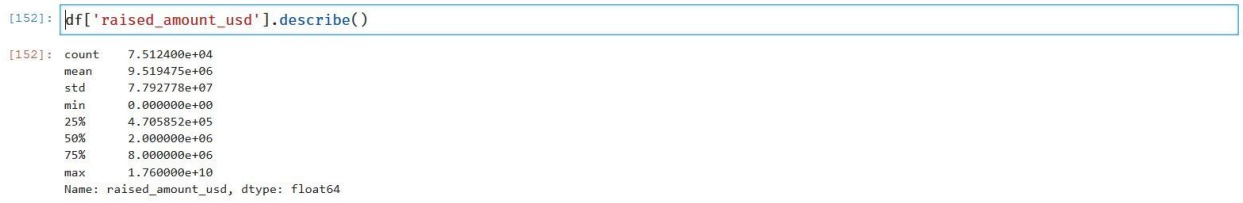
columns.But note the raised_amount_usd, country_code and category_list are useful so we need to clean them properly.
After all the cleaning we would save our master dataset into other csv file so we can directly use that file for our analysis.

Now It's time for analysis

Let's take another IPython notebook for our analysis this way we can keep our work clean and sorted.
First we will import our required libraries that we mentioned in the beginning of the post, along with our master csv file.
We only need four main funding types so we will use only that data which contains these funding types.
These four main funding types are :

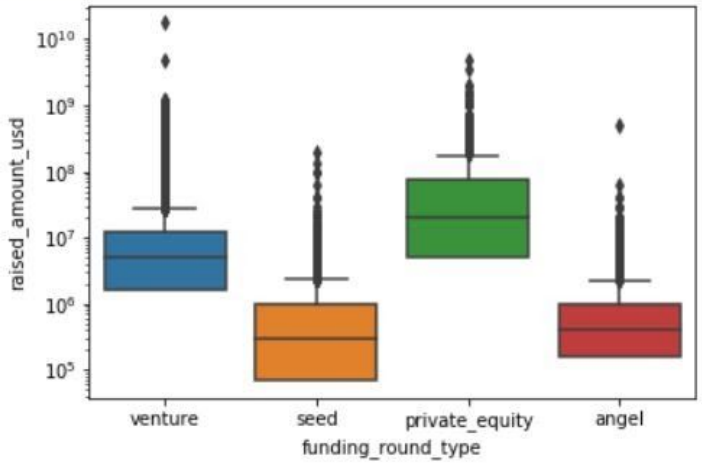
- Venture
- Angel
- Seed
- Private_equity

We need to compute the amount that we can spend for each funding type. We can either choose the mean or the median. Let's have a look at the raised_amount_usd column to get a sense.



Let's also look for raised amount and funding type.

```
[153]: sns.boxplot(x='funding_round_type', y='raised_amount_usd', data=df)
       plt.yscale('log')
       plt.show()
```



The median investment for type ‘private_equity’ is 20M which is beyond ABC’s investment range whereas the median of type ‘venture’ is 5M which is in the range.
Now let’s compare total amount across the countries.

```
[156]: df = df[df.funding_round_type=="venture"]
       country_wise_total = df.groupby('country_code')['raised_amount_usd'].sum().sort_values(ascending=False)
       print(country_wise_total)
```

```
[157]: top_9_countries = country_wise_total[:9]
       top_9_countries
```

```
[157]: country_code
       USA    4.200680e+11
       CHN    3.933892e+10
       GBR    2.007281e+10
       IND    1.426151e+10
       CAN    9.482218e+09
       FRA    7.226851e+09
       ISR    6.854350e+09
       DEU    6.306922e+09
       JPN    3.167647e+09
       Name: raised_amount_usd, dtype: float64
```

Now these are the top nine countries in terms of investment amount in venture type.

Among these nine countries USA,IND and GBR are top three english speaking countries.
Now it’s time for our mapping data file to come into play. It contains different sub-sectors with their main sectors.
We will merge our mapping and master file into one.
And applying the above tactics we can see that the USA will the country to invest in with Others as the sector to go for.