Data Sourcing

In [1]:
```
#import required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:
```
#Sourcing data
companies=pd.read_csv("companies.csv",encoding='latin-1')
rounds2 = pd.read_csv("rounds2.csv",encoding='latin-1')
mappings = pd.read_csv("mapping.csv",encoding='latin-1')
```

In [3]:
```
companies.head()
```

Out[3]:

	permalink	name	homepage_url	category_list	status	country_
0	/Organization/-Fame	#fame	http://livfame.com	Media	operating	
1	/Organization/-Qounter	:Qounter	http://www.qounter.com	Application Platforms|Real Time|Social Network...	operating	
2	/Organization/-The-One-Of-Them-Inc-	(THE) ONE of THEM,Inc.	http://oneofthem.jp	Apps|Games|Mobile	operating	
3	/Organization/0-6-Com	0-6.com	http://www.0-6.com	Curated Web	operating	
4	/Organization/004-Technologies	004 Technologies	http://004gmbh.de/en/004-interact	Software	operating	

In [4]:
```
rounds2.head()
```

Out[4]:

	company_permalink	funding_round_permalink	funding_round_type	funding_ro
0	/organization/-fame	/funding-round/9a01d05418af9f794eebff7ace91f638	venture	
1	/ORGANIZATION/-QOUNTER	/funding-round/22dacff496eb7acb2b901dec1dfe5633	venture	
2	/organization/-qounter	/funding-round/b44fbb94153f6cdef13083530bb48030	seed	
3	/ORGANIZATION/-THE-ONE-OF-THEM-INC-	/funding-round/650b8f704416801069bb178a1418776b	venture	
4	/organization/0-6-com	/funding-round/5727accaeaa57461bd22a9bdd945382d	venture	

In [5]:
```
mappings.head()
```

Out[5]:

	category_list	Automotive & Sports	Blanks	Cleantech / Semiconductors	Entertainment	Health	Manufacturing	Me
0	NaN	0	1	0	0	0	0	
1	3D	0	0	0	0	0	1	
2	3D Printing	0	0	0	0	0	1	
3	3D Technology	0	0	0	0	0	1	
4	Accounting	0	0	0	0	0	0	

Takeaways

1. companies dataframe has list of companies.
2. rounds2 dataframe has list of fundings, companies had raised.
3. mappings dataframe contains information of sectors associated with categories.

We can merge permalink column of companies and company_permalink column of rounds2.
We can create a derived column "Sector" based on category_list column of companies and mappings

Data Understanding of companies

In [6]:
```
companies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 66368 entries, 0 to 66367
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   permalink      66368 non-null  object
 1   name           66367 non-null  object
 2   homepage_url   61310 non-null  object
 3   category_list  63220 non-null  object
 4   status         66368 non-null  object
 5   country_code   59410 non-null  object
 6   state_code     57821 non-null  object
 7   region         58338 non-null  object
 8   city           58340 non-null  object
 9   founded_at     51147 non-null  object
dtypes: object(10)
memory usage: 5.1+ MB
```

In [7]:
```
#Understanding null values
100*(companies.isnull().sum()/len(companies))
```

Out[7]:
```
permalink         0.000000
name              0.001507
homepage_url      7.621143
category_list     4.743250
status            0.000000
country_code     10.483968
state_code       12.878194
region           12.099204
city             12.096191
founded_at       22.934245
dtype: float64
```

In [8]:
```
companies["status"].value_counts()
```

Out[8]:
```
operating    53034
closed        6238
acquired      5549
ipo           1547
Name: status, dtype: int64
```

Type *Markdown* and LaTeX: α^2

In [9]:
```
#Removing extra spaces and lower casing of permalink.
companies.permalink = companies.permalink.str.lower().str.strip()
```

In [10]:
```
#Removing extra spaces and lower casing of homepage_url
companies.homepage_url= companies.homepage_url.str.lower().str.strip()
```

In [11]:
```
#Removing extra spaces and lower casing of name
companies.name= companies.name.str.lower().str.strip()
```

In [12]:
```
#Correcting incorrect values in name
companies.name.loc[(companies.name=="#name?") | (companies.name.isnull()
```

In [13]:
```
companies.loc[(companies.name=="#name?") | (companies.name.isnull())].sh
```

Out[13]:
```
(0, 10)
```

In [14]:
```
#Understanding unique values
companies.describe()
```

Out[14]:

	permalink	name	homepage_url	category_list	status	country_code	s
count	66368	66368	61310	63220	66368	59410	
unique	66368	66038	61187	27296	4	137	
top	/organization/ce-interactive	roost	http://www.askforoffer.com	Software	operating	USA	
freq	1	4	5	3995	53034	37601	

Takeaways:

1. permalink, name and homepage_url represent about the company. **We need to further analyze these columns to find unique values.**
2. We want to find the countries to invest and category_list. These are target columns. **So we can drop the rows where country_code and category_list value is null.**
3. We can identify where to invest with-in the country if required. But For our case study (analysis), we are not interested in which city or region of a country the company should invest. Hence, can drop state_code, region and city columns, if required.
4. With Status column, we can further identify about the success rate of those investments. Like many companies are closed.
5. With founded_at feild we can check how old a company is.
6. Data set has 66368 rows and 10 columns. We need to find how many are unique rows.
7. founded_at column can be dropped as it has 22 percent nulls and This column is not needed for analysis

We need to find the reason for non unique name and homepage_url

In [15]:
```
# Non - unique  values of homepage_url
companies.loc[companies.duplicated(subset=["homepage_url"],keep=False)].
#As per data - For many companies homepage_url is same but name is diffe
#homepage_url
```

Out[15]:

	permalink	name	homepage_url	category_list	
7236	/organization/bittorrent	bittorrent	http://adcoock.com/sites/top-45-best-torrent-t...	Apps|Peer-to-Peer|Software	o
55148	/organization/stumbleupon	stumbleupon	http://adcoock.com/sites/top-45-best-torrent-t...	Content|Curated Web|Search	a
57990	/organization/thought-network-s-a-s	thought network s.a.s	http://app.thotz.co/	Apps|Digital Media|Internet|Software	
57986	/organization/thotz	thotz	http://app.thotz.co/	Content|Information Services|Visualization	
6824	/organization/bincode-entertainment	bincode entertainment	http://bincode-entertainment.com/	NaN	
6823	/organization/bincode	bincode	http://bincode-entertainment.com/	NaN	o
8388	/organization/brave-new-coin	brave new coin	http://bravenewcoin.com	Financial Services	o
56715	/organization/techemy-ltd	techemy ltd	http://bravenewcoin.com	Big Data|Bitcoin|FinTech	o
12868	/organization/confluent	confluent	http://confluent.io/	Big Data|Enterprise Software|Technology	o
12869	/organization/confluent-oblix-oracle	confluent (oblix / oracle)	http://confluent.io/	Computers|Software	o

In [16]:
```
# Non - unique  values of name
companies.loc[companies.duplicated(subset=["name"],keep=False)].sort_val
#As per data - For many companies name is same but homepage_url. Hence,
#but they are different
```

Out[16]:

	permalink	name	homepage_url	category_list	sta
281	/organization/3divaz-2	3divaz	http://www.3divaz.ch/home	NaN	clo
282	/organization/3divaz-3	3divaz	http://www.3divaz.ch/home	NaN	opera
1526	/organization/adtena	adtena	http://adtena.com/	Ad Targeting|Advertising|Mobile Advertising	opera
1527	/organization/adtena-2	adtena	http://adtena.com	NaN	clo
1995	/organization/agora-3	agora	http://www.agora.io/	Mobile|Mobile Software Tools|VoIP	opera
...	...	...	...	...	
65758	/organization/zenbox-2	zenbox	http://zenbox.us	Software	opera
				Curated Web|Health	

In [17]:
```
# Non - unique  records  (having same homepage_url and name)
companies.loc[companies.duplicated(subset=["name","homepage_url"],keep=F
```

Out[17]:

	permalink	name	homepage_url	cat
281	/organization/3divaz-2	3divaz	http://www.3divaz.ch/home	
282	/organization/3divaz-3	3divaz	http://www.3divaz.ch/home	
4168	/organization/ardian	ardian	http://www.ardian.com	Investment Ma
4169	/organization/ardian-inc	ardian	http://www.ardian.com	H
4481	/organization/arvegenix	arvegenix	http://www.arvegenix.com/	Fuels|Nutrition|Oil|F
4482	/organization/arvegenix-2	arvegenix	http://www.arvegenix.com/	Industrial|Oil|F
13751	/organization/credo-semiconductor	credo semiconductor	http://www.credosemi.com/	Semi
13752	/organization/credo-	credo	http://www.credosemi.com/	Semi

In [18]:
```
# Non - unique  records  (having same homepage_url, name and country coo
companies.loc[companies.duplicated(subset=["name","homepage_url","countr
```

Out[18]:

	permalink	name	homepage_url	category_list	s
4481	/organization/arvegenix	arvegenix	http://www.arvegenix.com/	Fuels|Nutrition|Oil|Renewable Energies	ope
4482	/organization/arvegenix-2	arvegenix	http://www.arvegenix.com/	Industrial|Oil|Renewable Energies	ope
23168	/organization/global-fashion-group	global fashion group	http://global-fashion-group.com/	Fashion	ope
23169	/organization/global-fashion-group-	global fashion group	http://global-fashion-group.com/	E-Commerce|Fashion	ope
26411	/organization/i-tech-2	i-tech	http://itech.wanye.cc/	NaN	c
26412	/organization/i-tech-3	i-tech	http://itech.wanye.cc/	Technology	ope
40447	/organization/ofixu	ofixu	http://www.ofixu.com	NaN	ope
40448	/organization/ofixu-2	ofixu	http://www.ofixu.com	Business Services|Office Space|Professional Se...	ope
48850	/organization/roost	roost	https://roost.com/	Local Based Services|Real Estate|Storage	acc
48851	/organization/roost-6	roost	https://roost.com/	Storage	ope
55154	/organization/stupsr	stupsr	NaN	NaN	c
55155	/organization/stupsr-2	stupsr	NaN	NaN	c
60011	/organization/twistlock	twistlock	https://www.twistlock.io/	Enterprises|Information Technology|Services	ope
60012	/organization/twistlock-2	twistlock	https://www.twistlock.io/	Security	ope
62899	/organization/wacai	wacai	http://www.wacai.com	Software	ope
62919	/organization/waicai	wacai	http://www.wacai.com	Finance|FinTech	ope

In [19]:
```
#To get the Unique company count count
companies.drop_duplicates(subset=["name","homepage_url"]).shape
```

Out[19]: (66350, 10)

Takeaway:

1. For many companies, either homepage_url is wrong or null.
2. **Duplicate companies -** The companies having same name and same or null homepage_url and same or null country code.
3. We will delete those records which has same name, homepage_url and country_code but different category_list. As this data is ambigioius.

4. There are 66350 unique companies.

Data Understanding of round2

In [20]:
```
rounds2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 114949 entries, 0 to 114948
Data columns (total 6 columns):
 #   Column                   Non-Null Count   Dtype  
---  ------                   --------------   -----  
 0   company_permalink        114949 non-null  object 
 1   funding_round_permalink  114949 non-null  object 
 2   funding_round_type       114949 non-null  object 
 3   funding_round_code       31140 non-null   object 
 4   funded_at                114949 non-null  object 
 5   raised_amount_usd        94959 non-null   float64
dtypes: float64(1), object(5)
memory usage: 5.3+ MB
```

In [21]:
```
#Understanding null values
100*(rounds2.isnull().sum()/len(rounds2))
```

Out[21]:
```
company_permalink           0.000000
funding_round_permalink     0.000000
funding_round_type          0.000000
funding_round_code         72.909725
funded_at                   0.000000
raised_amount_usd          17.390321
dtype: float64
```

In [22]:
```
rounds2.funding_round_type.value_counts()
```

Out[22]:
```
venture                  55494
seed                     30524
debt_financing            6895
angel                     6094
undisclosed               4897
equity_crowdfunding       3257
private_equity            2285
grant                     2200
convertible_note          1817
post_ipo_equity            638
product_crowdfunding       410
non_equity_assistance      191
post_ipo_debt              152
secondary_market            95
Name: funding_round_type, dtype: int64
```

In [23]:
```
#Removing extra spaces and makeing correct case of company_permalink.
rounds2.company_permalink = rounds2.company_permalink.str.lower().str.st
```

In [24]:
```
#Removing extra spaces and makeing correct case of funding_round_permali
rounds2.funding_round_permalink = rounds2.funding_round_permalink.str.lo
```

In [25]:
```
rounds2.head()
```

Out[25]:

	company_permalink	funding_round_permalink	funding_round_type	funding_ro
0	/organization/-fame	/funding-round/9a01d05418af9f794eebff7ace91f638	venture	
1	/organization/-qounter	/funding-round/22dacff496eb7acb2b901dec1dfe5633	venture	
2	/organization/-qounter	/funding-round/b44fbb94153f6cdef13083530bb48030	seed	
3	/organization/-the-one-of-them-inc-	/funding-round/650b8f704416801069bb178a1418776b	venture	
4	/organization/0-6-com	/funding-round/5727accaeaa57461bd22a9bdd945382d	venture	

In [26]:
```
rounds2.describe(include="all")
```

Out[26]:

	company_permalink	funding_round_permalink	funding_round_type	fundir
count	114949	114949	114949	
unique	66370	114949	14	
top	/organization/solarflare	/funding-round/e198e1213ce19f1fd70153f1eccb79da	venture	
freq	19	1	55494	
mean	NaN	NaN	NaN	
std	NaN	NaN	NaN	
min	NaN	NaN	NaN	
25%	NaN	NaN	NaN	
50%	NaN	NaN	NaN	
75%	NaN	NaN	NaN	
max	NaN	NaN	NaN	

Takeaways:

1. permalink - It represents the companies. **We need to further analyze these columns to find unique companies.**
2. funding_round_permalink - It represents unique funding round.
3. funding_round_type - Type of the funding.
4. funding_round_code - This coulumn has 72 % null va;ues. We can remove this column.
5. funded_at - When the funding happened.

6. raised_amount_usd - How much amount is raised. This is a target column. We will delete the records which has null values.

In [27]:
```
#To get the Unique company count count
rounds2.drop_duplicates(subset=["company_permalink"]).shape
```

Out[27]: (66370, 6)

In [28]:
```
# Non - unique companies
rounds2.loc[rounds2.duplicated(subset=["company_permalink"],keep=False)]
```

Out[28]:

	company_permalink	funding_round_permalink	funding_round_type	fund
1	/organization/-qounter	/funding-round/22dacff496eb7acb2b901dec1dfe5633	venture	
2	/organization/-qounter	/funding-round/b44fbb94153f6cdef13083530bb48030	seed	
7	/organization/0ndine-biomedical-inc	/funding-round/2b9d3ac293d5cdccbecff5c8cb0f327d	seed	
8	/organization/0ndine-biomedical-inc	/funding-round/954b9499724b946ad8c396a57a5f3b72	venture	
9	/organization/0xdata	/funding-round/383a9bd2c04f7038bb543ccef5ba3eae	seed	
...	...	...	...	
114938	/organization/zzish	/funding-round/34b560f672bebeb339a5efa3b27eae5d	grant	
114943	/organization/zzzzapp-com	/funding-round/6ba41360588bc6e3f77e9b50a0ebfafa	seed	
114944	/organization/zzzzapp-com	/funding-round/8f6d25b8ee4199e586484d817bceda05	convertible_note	
114942	/organization/zzzzapp-com	/funding-round/22ef2fafb4d20ac3aa4b86143dbf6c8e	seed	
114945	/organization/zzzzapp-com	/funding-round/ff1aa06ed5da186c84f101549035d4ae	seed	

72456 rows × 6 columns

In [29]:
```
#To get companies, which are not present in round2
rounds2.loc[~rounds2.company_permalink.isin(companies.permalink)]
```

Out[29]:

	company_permalink	funding_round_permalink	funding_round_type	fu
29597	/organization/e-cãbica	/funding-round/8491f74869e4fe8ba9c378394f8fbdea	seed	
31863	/organization/energystone-games-çˆµç³æ¸¸æ÷	/funding-round/b89553f3d2279c5683ae93f45a21cfe0	seed	
45176	/organization/huizuche-com-æ ç§ÿè½¦	/funding-round/8f8a32dbeeb0f831a78702f83af78a36	seed	
58473	/organization/magnet-tech-ç£ˆç³ç§æ¯	/funding-round/8fc91fbb32bc95e97f151dd0cb4166bf	seed	
101036	/organization/tipcat-interactive-æ²èÿä¿¡æ¯ç...	/funding-round/41005928a1439cb2d706a43cb661f60f	seed	
109969	/organization/weiche-tech-åè½¦ç§æ	/funding-round/f74e457f838b81fa0b29649740f186d8	venture	
113839	/organization/zengame-ç¦æ¸¸ç§æ	/funding-round/6ba28fb4f3eadf5a9c6c81bc5dde6cdf	seed	

Takeaways:

1. There are 66370 unique companies and some companies are not present in companies dataframe.
2. Many companies has multiple funding rounds.
3. Some companies of round2 are not present in companies

Data processing and cleaning of round2

In [30]:
```
#Renaming company_permalink to permalink
rounds2.rename(columns={'company_permalink': 'permalink'}, inplace=True)
```

In [31]:
```
#Deleting rows with null raised_amount_usd
rounds2 = rounds2.loc[~rounds2.raised_amount_usd.isnull()]
```

In [32]:
```
#Dropping funding_round_code
rounds2 = rounds2.drop(['funding_round_code'], axis = 1)
```

In [33]:
```
#Converting values in millions USD
rounds2.raised_amount_usd=rounds2.raised_amount_usd/1000000
```

In [34]:
```
rounds2.head()
```

Out[34]:

	permalink	funding_round_permalink	funding_round_type	funded_a
0	/organization/-fame	/funding-round/9a01d05418af9f794eebff7ace91f638	venture	05-01-201
2	/organization/-qounter	/funding-round/b44fbb94153f6cdef13083530bb48030	seed	01-03-201
3	/organization/-the-one-of-them-inc-	/funding-round/650b8f704416801069bb178a1418776b	venture	30-01-201
4	/organization/0-6-com	/funding-round/5727accaeaa57461bd22a9bdd945382d	venture	19-03-200
6	/organization/01games-technology	/funding-round/7d53696f2b4f607a2f2a8cbb83d01839	undisclosed	01-07-201

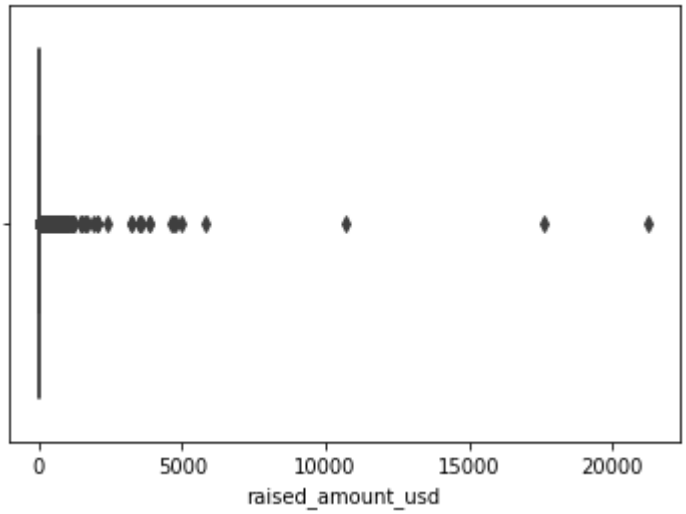
In [35]:
```
#Understanding null values
100*(rounds2.isnull().sum()/len(rounds2))
```

Out[35]:
```
permalink                 0.0
funding_round_permalink   0.0
funding_round_type        0.0
funded_at                 0.0
raised_amount_usd         0.0
dtype: float64
```

In [36]:
```
#Identifying outliers in raised_amount_usd
sns.boxplot(rounds2.raised_amount_usd)
```

```
/home/rahulg/.local/lib/python3.6/site-packages/seaborn/_decorators.p
y:43: FutureWarning: Pass the following variable as a keyword arg: x.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  FutureWarning
```

Out[36]: <AxesSubplot:xlabel='raised_amount_usd'>



We should get rid of values more than 500 million USD

In [37]:
```
#Identifying fundings with 0
rounds2.loc[rounds2.raised_amount_usd>=500].shape
```

Out[37]: (160, 5)

In [38]:
```
#Removing investments with more than 500 million USD
rounds2=rounds2.loc[rounds2.raised_amount_usd<500]
```

In [39]:
```
#Identifying outliers in raised_amount_usd
sns.boxplot(rounds2.raised_amount_usd)
```

```
/home/rahulg/.local/lib/python3.6/site-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

Out[39]:
```
<AxesSubplot:xlabel='raised_amount_usd'>
```

In [40]:
```
#Identifying fundings with 0
rounds2.loc[rounds2.raised_amount_usd>=100].shape
```

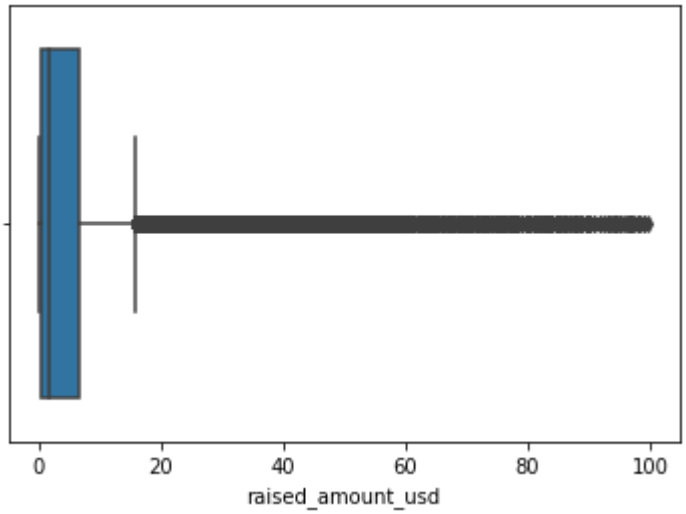
Out[40]:
```
(1127, 5)
```

In [41]:
```
#Removing investments with more than 100 million USD
rounds2=rounds2.loc[rounds2.raised_amount_usd<100]
```

In [42]:
```
sns.boxplot(rounds2.raised_amount_usd)
```

```
/home/rahulg/.local/lib/python3.6/site-packages/seaborn/_decorators.p
y:43: FutureWarning: Pass the following variable as a keyword arg: x.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  FutureWarning
```

Out[42]: <AxesSubplot:xlabel='raised_amount_usd'>



In [43]:
```
rounds2.shape
```

Out[43]: (93672, 5)

Data processing and cleaning of companies

In [44]:
```
#Deleting rows with null category_list
companies = companies.loc[~companies.category_list.isnull()]
```

In [45]:
```
#Deleting rows with null country_code
companies = companies.loc[~companies.country_code.isnull()]
```

In [46]:
```
#To delete ambigious data - having same name,country_code and homepage_u
companies=companies.drop_duplicates(subset=["name","homepage_url","count
```

In [47]:
```
#Dropping founded_at
companies = companies.drop(['founded_at'], axis = 1)
```

In [48]:
```
#correcting category list as per business rule - first category from lis
companies.category_list=companies.category_list.apply(lambda x: x.lower(
```

In [49]:
```
#Understanding null values of companies
100*(companies.isnull().sum()/len(companies))
```

Out[49]:
```
permalink        0.000000
name             0.000000
homepage_url     5.773956
category_list    0.000000
status           0.000000
country_code     0.000000
state_code       2.657715
region           1.797764
city             1.794304
dtype: float64
```

Home page url, state_code, region and city columns have null values.

1. We can either impute www..com in homepage_url and drop the column.
2. We can either impute modde of state, region and city per country. or drop these columns.

   In this case study, we are not analyzing data based on these columns. Hence, dropping these columns.

In [50]:
```
companies = companies.drop(['homepage_url',"state_code","region","city"]
```

In [51]:
```
#Understanding null values of companies
100*(companies.isnull().sum()/len(companies))
```

Out[51]:
```
permalink        0.0
name             0.0
category_list    0.0
status           0.0
country_code     0.0
dtype: float64
```

Data processing and cleaning of mappings

In [52]:
```
mappings.head()
```

Out[52]:

	category_list	Automotive & Sports	Blanks	Cleantech / Semiconductors	Entertainment	Health	Manufacturing	Me
0	NaN	0	1	0	0	0	0	
1	3D	0	0	0	0	0	1	
2	3D Printing	0	0	0	0	0	1	
3	3D Technology	0	0	0	0	0	1	
4	Accounting	0	0	0	0	0	0	

In [53]:
```
# store the value and id variables in two separate arrays

# store the value variables in one Series
value_vars = list(mappings.columns[1:])

# take the setdiff() to get the rest of the variables
id_vars = list(np.setdiff1d(mappings.columns, value_vars))

print(value_vars, "\n")
print(id_vars)
```

```
['Automotive & Sports', 'Blanks', 'Cleantech / Semiconductors', 'Enter
tainment', 'Health', 'Manufacturing', 'News, Search and Messaging', 'O
thers', 'Social, Finance, Analytics, Advertising']

['category_list']
```

In [54]:
```
#converting wide data set to long data set.
mappings = pd.melt(mappings, id_vars = id_vars, value_vars = value_vars)
```

In [55]:
```
#Understanding null values
100*(mappings.isnull().sum()/len(mappings))
```

Out[55]:
```
category_list    0.145349
variable         0.000000
value            0.000000
dtype: float64
```

In [56]:
```
#removing null category and values with 0
mappings=mappings.loc[(~mappings.category_list.isnull())]
```

In [57]:
```
#Understanding null values
100*(mappings.isnull().sum()/len(mappings))
```

Out[57]:
```
category_list    0.0
variable         0.0
value            0.0
dtype: float64
```

In [58]:
```
#Geting values
mappings.value.value_counts()
```

Out[58]:
```
0    5496
1     687
Name: value, dtype: int64
```

In [59]:
```
#removing 0 values
mappings=mappings.loc[mappings.value==1]
```

In [60]:
```
#Removing extra spaces and lower casing of category_list
mappings.category_list = mappings.category_list.str.lower().str.strip()
```

In [61]:
```
#Geting values
mappings.value.value_counts()
```

Out[61]:
```
1    687
Name: value, dtype: int64
```

In [62]:
```
#Dropping value
mappings = mappings.drop(['value'], axis = 1)
```

In [63]:
```
#To check all values in category_list are unique
mappings.describe()
```

Out[63]:

	category_list	variable
count	687	687
unique	687	8
top	group sms	Others
freq	1	195

In [64]:
```
mappings.head()
```

Out[64]:

	category_list	variable
8	adventure travel	Automotive & Sports
14	aerospace	Automotive & Sports
45	auto	Automotive & Sports
46	automated kiosk	Automotive & Sports
47	automotive	Automotive & Sports

In [65]:
```
#To check the distribution of sectors
mappings.variable.value_counts()
```

Out[65]:
```
Others                                    195
Social, Finance, Analytics, Advertising   153
Entertainment                              89
News, Search and Messaging                 72
Health                                     63
Cleantech / Semiconductors                 53
Manufacturing                              40
Automotive & Sports                        22
Name: variable, dtype: int64
```

In [66]:
```
#Renaming variable to sector
mappings.rename(columns={'variable': 'sector'}, inplace=True)
```

In [67]:
```
mappings.head()
```

Out[67]:

	category_list	sector
8	adventure travel	Automotive & Sports
14	aerospace	Automotive & Sports
45	auto	Automotive & Sports
46	automated kiosk	Automotive & Sports
47	automotive	Automotive & Sports

Merging companies and round2

In [68]:
```
#To merge companies and round2
master_frame=pd.merge(companies,rounds2,how="inner",on="permalink")
```

In [69]:
```
master_frame.head()
```

Out[69]:

	permalink	name	category_list	status	country_code	fundi
0	/organization/-fame	#fame	media	operating	IND	round/9a01d05418af
1	/organization/-qounter	:qounter	application platforms	operating	USA	round/b44fbb94153f6c
2	/organization/0-6-com	0-6.com	curated web	operating	CHN	round/5727accaeaa574
3	/organization/01games-technology	01games technology	games	operating	HKG	round/7d53696f2b4f60
4	/organization/0ndine-biomedical-inc	ondine biomedical inc.	biotechnology	operating	CAN	round/2b9d3ac293d5

Adding sector column in master frame

In [70]:
```
# Identifying records of categories which are not part of mappings data
master_frame.loc[~(master_frame.category_list.isin(mappings.category_lis
```

Out[70]: (6314, 9)

There are many categories which are not present in mappings. As we don't know the sector details. Hence, removing these records.

In [71]:
```
#To merge the sector with master frame and remove the records of categor
master_frame=pd.merge(master_frame,mappings)
```

In [72]:
```
#Verify whether data is deleted
master_frame.loc[~(master_frame.category_list.isin(mappings.category_lis
```

Out[72]: (0, 10)

In [73]:
```
master_frame.head()
```

Out[73]:

	permalink	name	category_list	status	country_code	funding_round
0	/organization/-fame	#fame	media	operating	IND	round/9a01d05418af9f794eebf
1	/organization/90min	90min	media	operating	GBR	round/21a2cbf6f2fb2a1c2a61e
2	/organization/90min	90min	media	operating	GBR	round/bd626ed022f5c66574b1a
3	/organization/90min	90min	media	operating	GBR	round/fd4b15e8c97ee2ffc0accc
4	/organization/a-dance-for-me	a dance for me	media	operating	USA	round/9ab9dbd17bf010c79d841

Univariate Analysis

In [74]:
```
#Sector Distribution
master_frame.sector.value_counts(normalize=True)*100
```

Out[74]:
```
Others                                    24.478659
Social, Finance, Analytics, Advertising   18.295408
Cleantech / Semiconductors                17.534849
News, Search and Messaging                15.048214
Health                                     8.256269
Entertainment                              7.294458
Manufacturing                              6.927759
Automotive & Sports                        2.164385
Name: sector, dtype: float64
```

In [75]:
```
#Country Distribution
(master_frame.country_code.value_counts(normalize=True)*100).head(15)
```

Out[75]:
```
USA    69.660341
GBR     5.639994
CAN     2.969392
CHN     2.086600
IND     1.793982
FRA     1.693974
ISR     1.560629
ESP     1.254429
DEU     1.165533
AUS     0.743274
RUS     0.644500
SWE     0.629684
IRL     0.614868
SGP     0.590174
NLD     0.579062
Name: country_code, dtype: float64
```

In [76]:
```
#Funding Round Distribution
master_frame.funding_round_type.value_counts(normalize=True)*100
```

Out[76]:
```
venture                  54.599780
seed                     23.742793
debt_financing            7.331498
angel                     4.949810
grant                     2.227353
private_equity            1.638413
undisclosed               1.508772
convertible_note          1.484079
equity_crowdfunding       1.279123
post_ipo_equity           0.607460
product_crowdfunding      0.397565
post_ipo_debt             0.139518
non_equity_assistance     0.071611
secondary_market          0.022224
Name: funding_round_type, dtype: float64
```

In [77]:
```
#Company status Distribution
master_frame.status.value_counts(normalize=True)*100
```

Out[77]:
```
operating    77.843764
acquired     11.120714
closed        6.611682
ipo           4.423839
Name: status, dtype: float64
```

In [78]:
```
#raised_amount_usd Distribution
master_frame.raised_amount_usd.describe()
```

Out[78]:
```
count    80993.000000
mean         6.247229
std         11.352483
min          0.000000
25%          0.350000
50%          1.700000
75%          7.000000
max         99.800000
Name: raised_amount_usd, dtype: float64
```

In [79]:
```
#Percentage investment between 3m and 15m
(master_frame.raised_amount_usd.loc[(master_frame.raised_amount_usd>=3)
                                    (master_frame.raised_amount_usd<=15)
```

Out[79]:
```
29.505018952255135
```

Take Aways

1. Most investors invests in USA.
2. More than 50 percent of companies has raised "venture" type funding.
3. Around 77 percent companies are operating currently. And about 6 percent are closed.
4. About 50 percent fundings has value ranging between .35 million and 7 million USD.
5. Reprsentative funding amount is 1.7 million USD
6. Most favourable sectors for investment are (Social, Finance, Analytics, Advertising), Others and Cleantech / Semiconductors.
7. 29.5 percent investments are in range 3 million and 15 million

Bivariate Analysis

In [80]:
```
#To identify representative investment under all funding round type
amount_round=pd.pivot_table(master_frame, values ='raised_amount_usd',
                            index =['funding_round_type'], aggfunc = np.
```

In [81]:
```
funding_type_median=amount_round.sort_values(by="raised_amount_usd",asce
```

In [82]:
```
funding_type_median
```

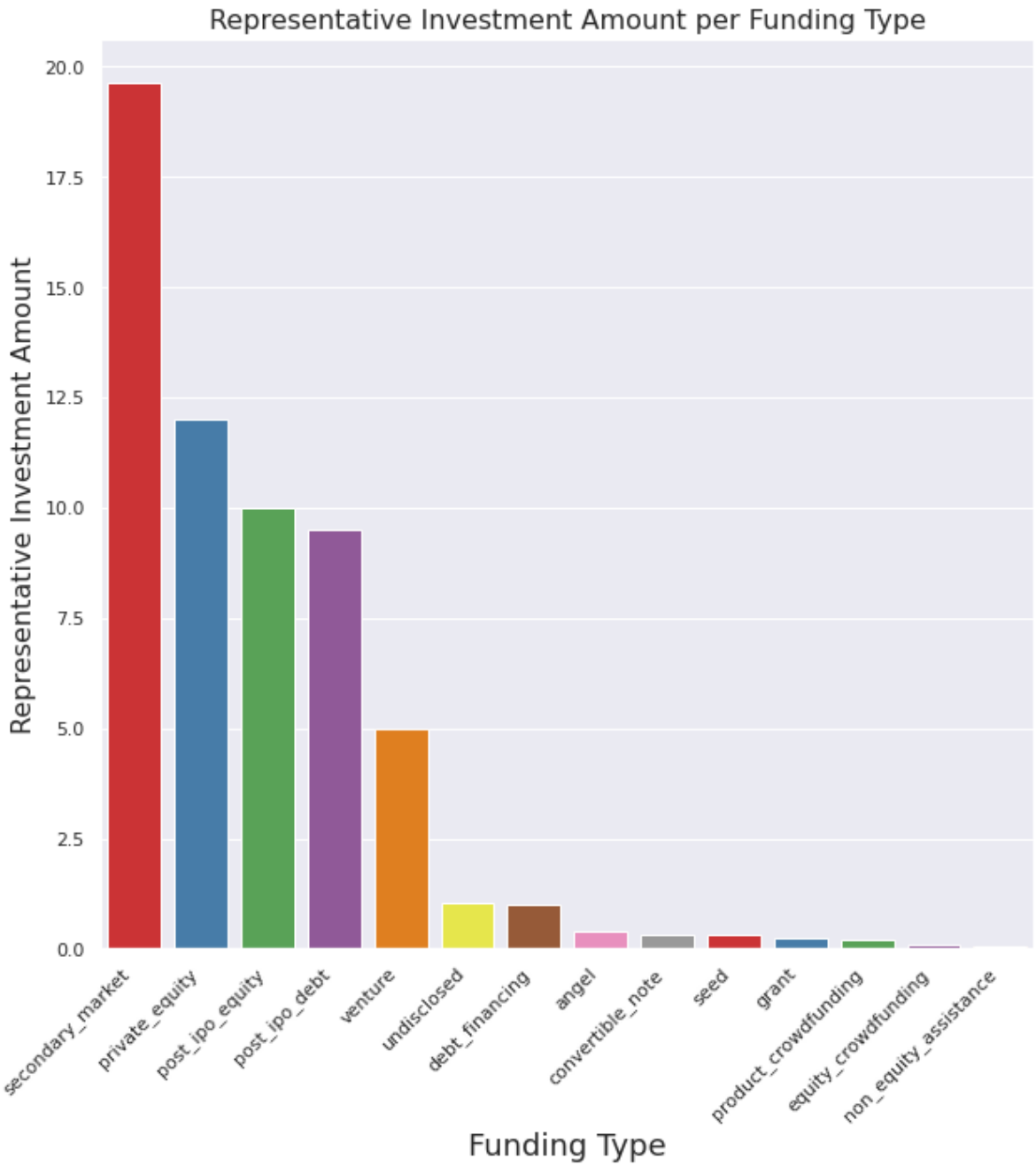
Out[82]:

funding_round_type	raised_amount_usd
secondary_market	19.650000
private_equity	12.000000
post_ipo_equity	10.000000
post_ipo_debt	9.500000
venture	5.000000
undisclosed	1.031499
debt_financing	1.000000
angel	0.410000
convertible_note	0.300000
seed	0.300000
grant	0.223975
product_crowdfunding	0.200000
equity_crowdfunding	0.088969
non_equity_assistance	0.060000

In [83]:
```
funding_type_median.reset_index(level=0, inplace=True)
```

In [84]:
```
sns.set_style=("whitegrid")
sns.set(rc={'figure.figsize':(10,10)})
chart=sns.barplot(x="funding_round_type",y="raised_amount_usd",data=func
chart.set_xticklabels(chart.get_xticklabels(), rotation=45, horizontalal
plt.xlabel('Funding Type', fontsize=18)
plt.ylabel('Representative Investment Amount', fontsize=16)
plt.title("Representative Investment Amount per Funding Type",fontsize=1
```

Out[84]: Text(0.5, 1.0, 'Representative Investment Amount per Funding Type')



private_equity, venture, post_ipo_debt and post_ipo_equity are having representative investment amount between 3 and 15 USD. But, more than 50 percent of companies has raised "venture" type funding. Hence we will choose venture as investment type.

In [85]:
```
#Keeping only venture type records in data set.
master_frame=master_frame.loc[master_frame.funding_round_type=='venture'
```

In [86]:
```
master_frame.shape
```

Out[86]: (44222, 10)

In [87]:
```
#To identify top 9 countries with highest funding amount
highest_round_country=pd.pivot_table(master_frame, values ='raised_amoun
                                     index =['country_code'], aggfunc =
```

In [88]:
```
#Top 9 countries with highest funding amount
top9=highest_round_country.sort_values(by="raised_amount_usd",ascending=
```

In [89]:
```
top9
```

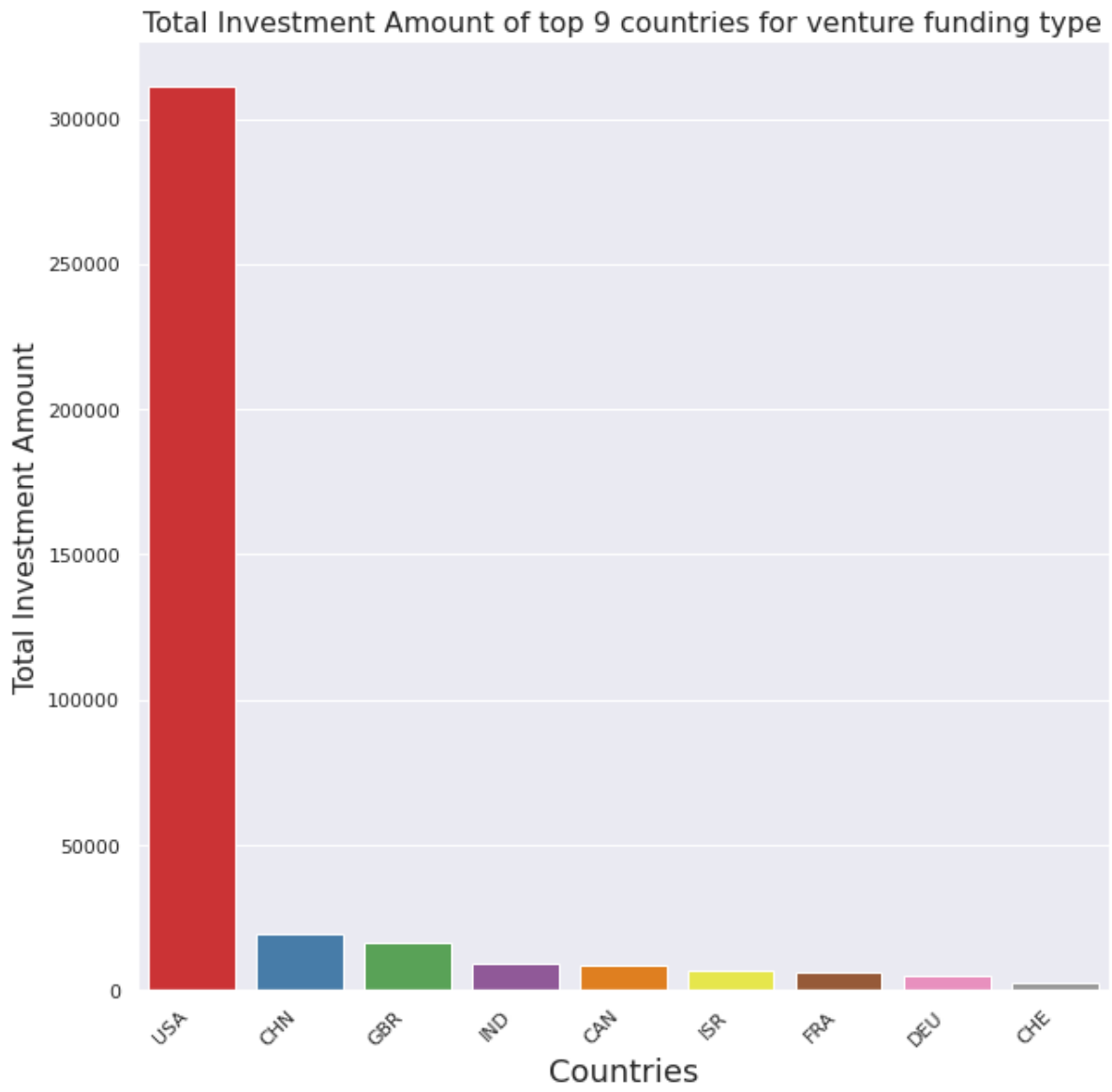
Out[89]:

country_code	raised_amount_usd
USA	310894.274778
CHN	19530.436127
GBR	15995.494685
IND	8867.281237
CAN	8550.620524
ISR	6370.700477
FRA	5861.700436
DEU	4973.133571
CHE	2731.657869

In [90]:
```
top9.reset_index(level=0, inplace=True)
```

In [166]:
```
chart=sns.barplot(x="country_code",y="raised_amount_usd",data=top9,palet
chart.set_xticklabels(chart.get_xticklabels(), rotation=45, horizontalal
plt.xlabel('Countries', fontsize=18)
plt.ylabel('Total Investment Amount', fontsize=16)
plt.title("Total Investment Amount of top 9 countries for venture fundir
```

Out[166]:
```
Text(0.5, 1.0, 'Total Investment Amount of top 9 countries for venture
funding type')
```



Top 3 English speaking countries are USA, GBR and IND

In [92]:
```
#Divinng data based on countries
c1=master_frame.loc[master_frame.country_code=='USA']
c2=master_frame.loc[master_frame.country_code=='GBR']
c3=master_frame.loc[master_frame.country_code=='IND']
```

In [93]:
```
#Total investments of USA
c1.shape
```

Out[93]: (33229, 10)

In [94]:
```
#Total investments of GBR
c2.shape
```

Out[94]: (1903, 10)

In [95]:
```
#Total investments of IND
c3.shape
```

Out[95]: (727, 10)

In [96]:
```
#Getting number of investments per sector of USA
c1.sector.value_counts()
```

Out[96]:
```
Others                                    8169
Cleantech / Semiconductors                7772
Social, Finance, Analytics, Advertising   5105
News, Search and Messaging                4258
Health                                    3247
Manufacturing                             2439
Entertainment                             1744
Automotive & Sports                        495
Name: sector, dtype: int64
```

In [141]:
```
#Getting top 3 sectors of USA
top_sector_c1 = set(pd.DataFrame(c1.sector.value_counts().head(3)).index
top_sector_c1
```

Out[141]:
```
{'Cleantech / Semiconductors',
 'Others',
 'Social, Finance, Analytics, Advertising'}
```

In [98]:
```
#Getting number of investments per sector of GBR
c2.sector.value_counts()
```

Out[98]:
```
Others                                    505
Cleantech / Semiconductors                430
Social, Finance, Analytics, Advertising   315
News, Search and Messaging                238
Entertainment                             132
Manufacturing                             121
Health                                    118
Automotive & Sports                        44
Name: sector, dtype: int64
```

In [139]:
```
#Getting top 3 sectors of GBR
top_sector_c2 = set(pd.DataFrame(c2.sector.value_counts().head(3)).index
top_sector_c2
```

Out[139]:
```
{'Cleantech / Semiconductors',
 'Others',
 'Social, Finance, Analytics, Advertising'}
```

In [100]:
```
#Getting number of investments per sector of IND
c3.sector.value_counts()
```

Out[100]:
```
Others                                    270
News, Search and Messaging                129
Social, Finance, Analytics, Advertising    76
Entertainment                              75
Manufacturing                              54
Cleantech / Semiconductors                 53
Health                                     42
Automotive & Sports                        28
Name: sector, dtype: int64
```

In [140]:
```
#Getting top 3 sectors of IND
top_sector_c3 = set(pd.DataFrame(c3.sector.value_counts().head(3)).index
top_sector_c3
```

Out[140]:
```
{'News, Search and Messaging',
 'Others',
 'Social, Finance, Analytics, Advertising'}
```

In [144]:
```
top_sectors=list(top_sector_c1.union(top_sector_c2, top_sector_c3))
top_sectors
```

Out[144]:
```
['Others',
 'Cleantech / Semiconductors',
 'News, Search and Messaging',
 'Social, Finance, Analytics, Advertising']
```

In [121]:
```
ount = master_frame.pivot_table(values="permalink", index="country_code"
```

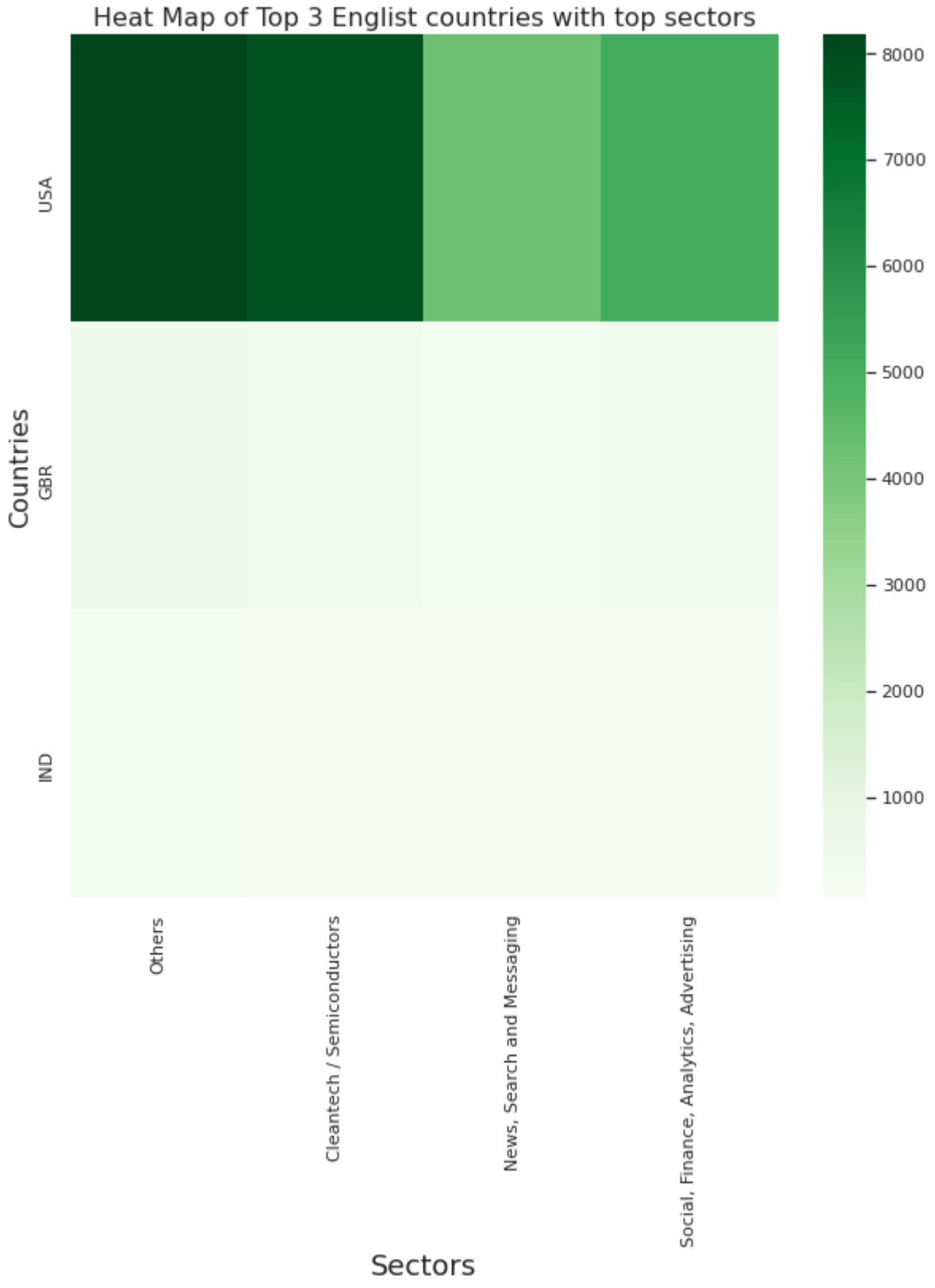
In [146]:
```
sector_details = pivot_count.loc[["USA","GBR","IND"],top_sectors]
sector_details
```

Out[146]:

sector	Others	Cleantech / Semiconductors	News, Search and Messaging	Social, Finance, Analytics, Advertising
country_code				
USA	8169.0	7772.0	4258.0	5105.0
GBR	505.0	430.0	238.0	315.0
IND	270.0	53.0	129.0	76.0

In [165]:
```
#To generate head map
sns.heatmap(sector_details,cmap="Greens")
plt.xlabel('Sectors', fontsize=18)
plt.ylabel('Countries', fontsize=16)
plt.title("Heat Map of Top 3 Englist countries with top sectors",fontsi
```

Out[165]: Text(0.5, 1.0, 'Heat Map of Top 3 Englist countries with top sectors')



In [102]:
```
#Getting company having maximum investments in top 3 sectors of USA
pd.pivot_table(c1.loc[c1.sector.isin(top_sector_c1)], values ='raised_am
               index =["sector","name",'permalink'], aggfunc = np.sum).s
```

Out[102]:

			raised_amount_usd
sector	**name**	**permalink**	
Social, Finance, Analytics, Advertising	**appnexus**	**/organization/appnexus**	285.671856
	stripe	**/organization/stripe**	278.000000
Cleantech / Semiconductors	**alien technology**	**/organization/alien-technology**	265.000000
	relypsa	**/organization/relypsa**	255.729847
Others	**force10 networks**	**/organization/force10-networks**	255.067782
...	**...**	**...**	...
Social, Finance, Analytics, Advertising	**linkmeglobal**	**/organization/linkmeglobal**	0.001000
	sevenlunches	**/organization/sevenlunches**	0.000291
	sentic technologies inc	**/organization/sentic-technologies-inc**	0.000001
Others	**promisec**	**/organization/promisec**	0.000000
Cleantech / Semiconductors	**cosmosid**	**/organization/cosmosid**	0.000000

11074 rows × 1 columns

In [103]:
```
#Getting company having maximum investments in top 3 sectors of GBR
pd.pivot_table(c2.loc[c2.sector.isin(top_sector_c2)], values ='raised_am
               index =["sector","name",'permalink'], aggfunc = np.sum).s
```

Out[103]:

			raised_amount_usd
sector	**name**	**permalink**	
Others	**farfetch**	**/organization/farfetch**	194.500000
Social, Finance, Analytics, Advertising	**powa technologies**	**/organization/powa-technologies**	176.700000
Cleantech / Semiconductors	**circassia**	**/organization/circassia**	144.630999
	biovex	**/organization/biovex**	133.314585
	kymab	**/organization/kymab**	120.400000
...	**...**	**...**	...
Social, Finance, Analytics, Advertising	**paperfold**	**/organization/paperfold**	0.056695
Others	**socii**	**/organization/socii**	0.054000
Social, Finance, Analytics, Advertising	**scaleogy**	**/organization/scaleogy**	0.050000

In [138]:
```
#Getting company having maximum investments in top 3 sectors of IND
pd.pivot_table(c3.loc[c3.sector.isin(top_sector_c3)], values ='raised_am
               index =["sector","name",'permalink'], aggfunc = np.sum).s
```

Out[138]:

			raised_amount_usd
sector	**name**	**permalink**	
News, Search and Messaging	**quikr**	**/organization/quikr-india**	196.000
Others	**snapdeal**	**/organization/snapdeal**	177.000
	myntra	**/organization/myntra**	158.750
	delhivery	**/organization/delhivery**	127.500
News, Search and Messaging	**freecharge**	**/organization/freecharge**	113.000
...	**...**	**...**	...
Others	**sudiksha**	**/organization/sudiksha**	0.075
	experifun	**/organization/experifun**	0.075
	egully	**/organization/egully**	0.050
News, Search and Messaging	**zify - instant carpooling app**	**/organization/zify**	0.040
Social, Finance, Analytics, Advertising	**securesight technologies**	**/organization/securesight-technologies**	0.010

322 rows × 1 columns

Conclusion

Country = USA
Amount = 5 million USD
Sector = Cleantech / Semiconductors