Introduction

This paper shows the insights of funding done by startups and how factors. The aim of paper is to get a descriptive overview and a r funding and growth of newly launched startups. Another important p funding changes with time is an important aspect. Possible area of (Funding ecosystem and time relation, cities as a important factor important investors). Dataset we are using contains information of January 2015 to August 2017.The amount invested is in USD. Aggrega investors, funding type etc. is required to get an optimized resul preprocessing of data and overcome problem of missing data and unc Visualizations are done to find the anomalies and mining patterns some cities showing some abnormal behavior when it comes to fundin

Data is in comma seperated values (C.S.V) format

In [56]:
```
import numpy as np #used for scientific computation
import pandas as pd #used for data mugging and preprocessing
import matplotlib.pyplot as plt #data visualization library
from pandas import DataFrame as show # dataframe is the optimised st
import seaborn as sns # stastical visualization library
%matplotlib inline #used in jupyter notebook for interactive visuali
import squarify
```

```
UsageError: unrecognized arguments: #used in jupyter notebook for interactive visualizations
```

Data Formatting

Reading and making dataframe of the csv formated file
Csv file-(startup_funding.csv)- contains the information of all st 2017

In [3]:
```
df = pd.read_csv('startup_funding.csv')
```

Sample of our dataset (startup_funding.csv)

In [4]:
```
df.head(4)#display first 4 rows of dataframe
```

	SNo	Date	StartupName	IndustryVertical	SubVertical	CityLocation	InvestorsNa
0	0	01/08/2017	TouchKin	Technology	Predictive Care Platform	Bangalore	Kae Capital
1	1	02/08/2017	Ethinos	Technology	Digital Marketing Agency	Mumbai	Triton Investm Advisors
2	2	02/08/2017	Leverage Edu	Consumer Internet	Online platform for Higher Education Services	New Delhi	Kashyap Deor Anand Sankeshwar, Deepak Jain,..
3	3	02/08/2017	Zepo	Consumer Internet	DIY Ecommerce platform	Mumbai	Kunal Shah, LetsVenture, Anupam Mittal Hetal ...

In [5]:
```
df.tail(4)#display first 4 rows of dataframe
```

	SNo	Date	StartupName	IndustryVertical	SubVertical	CityLocation	Investo
2368	2368	29/01/2015	Graphene	NaN	NaN	NaN	KARSEM Fund
2369	2369	30/01/2015	Mad Street Den	NaN	NaN	NaN	Exfinity F GrowX V
2370	2370	30/01/2015	Simplotel	NaN	NaN	NaN	MakeMy
2371	2371	31/01/2015	couponmachine.in	NaN	NaN	NaN	UK base of Angel

Metadata

```
Columns and index information
```

In [6]:
```
print("Information of total number of non-empty columns")
print("------------------------------------------------")
print(df.info(null_counts=True))
```

```
Information of total number of non-empty columns
------------------------------------------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2372 entries, 0 to 2371
Data columns (total 10 columns):
SNo                 2372 non-null int64
Date                2372 non-null object
StartupName         2372 non-null object
IndustryVertical    2201 non-null object
SubVertical         1436 non-null object
CityLocation        2193 non-null object
InvestorsName       2364 non-null object
InvestmentType      2371 non-null object
AmountInUSD         1525 non-null object
Remarks             419 non-null object
dtypes: int64(1), object(9)
memory usage: 185.4+ KB
None
```

Dimentions of dataframe

In [7]:
```
print("Columns and their datatypes")
df.dtypes
```

```
Columns and their datatypes


SNo                  int64
Date                object
StartupName         object
IndustryVertical    object
SubVertical         object
CityLocation        object
InvestorsName       object
InvestmentType      object
AmountInUSD         object
Remarks             object
dtype: object
```

Cleaning Data

In [8]:
```
print("Columns and their datatypes")
df.dtypes #.dtypes are used to display datatypes of each column
```

```
Columns and their datatypes


SNo                 int64
Date               object
StartupName        object
IndustryVertical   object
SubVertical        object
CityLocation       object
InvestorsName      object
InvestmentType     object
AmountInUSD        object
Remarks            object
dtype: object
```

Repersentation of missing data

Since we can see that 'remarks' has the higher density of missing column
Here we can use estimated statistical values of available data of fill the missing values

Dataframe contains lots of NaN(null values)

In [9]:
```
print("Frequency count of missing values")
df.apply(lambda X:sum(X.isnull()))
#apply function is used to do mapping column-wise
#apply function can apply tranformations to each column individually
```

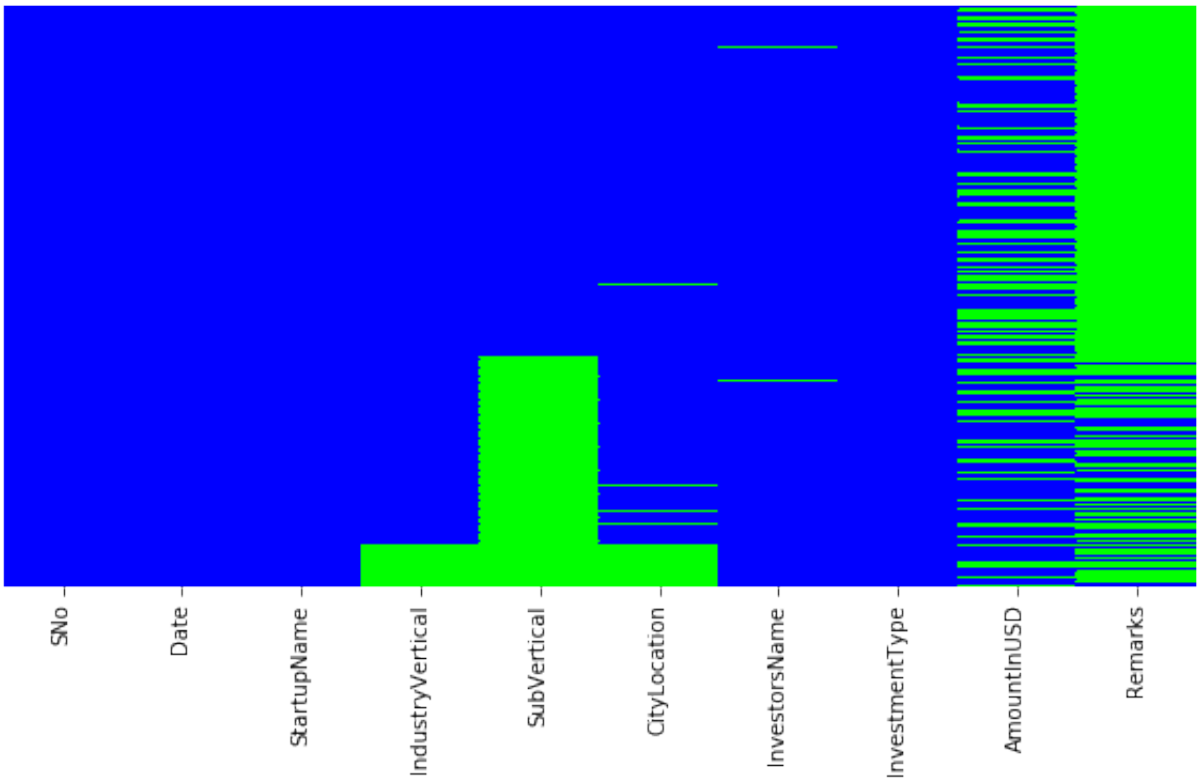
```
Frequency count of missing values


SNo                    0
Date                   0
StartupName            0
IndustryVertical     171
SubVertical          936
CityLocation         179
InvestorsName          8
InvestmentType         1
AmountInUSD          847
Remarks             1953
dtype: int64
```

Here yellow bars repersent the null values(missing values)

x axis represents colums(features) of dataset

In [10]:
```
plt.figure(figsize=(10,5)) #plt is the object of matplot lib and .fi
sns.heatmap(df.isnull(),cmap='brg',yticklabels=False,cbar=False)#hea
plt.show()
```



Specifying format error

Some dates are not formated where generalized format is 'dd/mm/yy'

In [11]:
```
print("Here we can see in date column error- '.' is there instead of
df[df['Date']=='12/05.2015']['Date']
```

```
Here we can see in date column error- '.' is there instead of '/'


2103    12/05.2015
2104    12/05.2015
Name: Date, dtype: object
```

Amount in usd has a delemiter ',' which cannot be processed becaus
and null values(missing data)
datatype is String and alot of missing values in Amount given For

In [12]:
```
df['AmountInUSD'].head(5)#head(n) displays n rows
```

```
0    1,300,000
1          NaN
2          NaN
3      500,000
4      850,000
Name: AmountInUSD, dtype: object
```

Solving problem with missing data

WE CAN FILL THE MISSING NUMERICAL VALUES USING FOLLWING STATI

-BY MODE OF CENTRAL TENDENCY

- BACKWARD FILLING

-FORWARD FILLING

-INTERPOLATION(LINEAR)

Cleaning missing data and formatting

Cities and IndustryVertical columns are interpolated but not with

In [13]:
```
df['CityLocation']=df['CityLocation'].fillna(value='NotSpecific')
df['IndustryVertical']=df['IndustryVertical'].fillna(value='Other')
```

city column is having multiple city names for some records

In [14]:
```
import re#importing regular expressions
def convert_Slash(x):#converts citylocation where multiple citiescen
    x=x.lower()#converting  whole data to lower case to avoid dublic
    if   re.search('/',x):
        return x.split('/')[0].strip()#converting multiple citycentr
    else :
        return x.strip()# removing extra spaces from left and right
df['CityLocation']=df['CityLocation'].apply(convert_Slash)
```

Backup of dataframe

Deleting Insignificant columns

Here as we can see that 'Remarks column has very high missing data
column is not useful and will create bias in analysis
Serial number is not useful and hence deleted as data is organised

In [15]:
```
newdf=df.copy()#backup cleansed data
del newdf['Remarks']#remaks is deleted to overcome stability in anal
del newdf['SNo']
```

Investment type column has repeated values of categories

Categories have alphabetical error

This column has unformatted categories which results in repetition
words) Extra spaces in categories of Investment type will create d
example Here -'Seed Funding' and 'SeedFunding' are same and can ca

In [16]:
```
print('Different categories of Inverstment Type before cleansing and
newdf['InvestmentType'].value_counts().index# aggregating frequency
```

```
Different categories of Inverstment Type before cleansing and removing duplicacy in categori

Index(['Seed Funding', 'Private Equity', 'SeedFunding', 'Crowd Funding',
       'Crowd funding', 'PrivateEquity', 'Debt Funding'],
      dtype='object')
```

In [17]:
```
print('Different categories of Inverstment Type after cleansing and
newdf['InvestmentType']=newdf['InvestmentType'].astype(str).apply(la
newdf['InvestmentType'].value_counts().index
```

```
Different categories of Inverstment Type after cleansing and removing duplicacy in categorie

Index(['seedfunding', 'privateequity', 'crowdfunding', 'nan', 'debtfunding'], dtype='object'
```

Fomatting dates to time series

In [18]:
```
n funding dataframe
5','13/042015' where backslash (/) is missing or at wrong position


ɔiton to date column using apply() which maps u.d.f to each record of


e format and to_datetime() is used to convert the datatype of date col
```

In [19]:
```
print('processed datatype of Date column')
newdf.dtypes['Date']
```

```
processed datatype of Date column


dtype('<M8[ns]')
```

In [20]:
```
newdf['InvestmentType'].head(4)
```

```
0    privateequity
1    privateequity
2      seedfunding
3      seedfunding
Name: InvestmentType, dtype: object
```

Preprocessing number of investors for each startup

In [21]:
```
# feature engineerng
def calculate_n_investors(x):#function to calculate record wise numb
    if  re.search(',',x) and x!='empty':
        return len(x.split(','))
    elif x!='empty':
        return 1
    else:
        return -1
newdf['numberofinvestors']=newdf['InvestorsName'].replace(np.NaN,'em
```

In [22]:
```
n_inv2=newdf

n_inv=newdf['InvestorsName']
n_inv.fillna(value='None',inplace=True)
listed_n_inv=n_inv.apply(lambda x: x.lower().strip().split(','))
investors=[]
for i in listed_n_inv:
    for j in i:
        if(i!='None' or i!=''):
            investors.append(j.strip())
unique_investors=list(set(investors))
```

In [23]:
```
investors=pd.Series(investors)
unique_investors=pd.Series(unique_investors)
```

In [24]:
```
investors=list(investors[investors!=''])
unique_investors=list(unique_investors[unique_investors!=''])
```

In [25]:
```
for i in range(len(unique_investors)):
    for j in range(len(investors)):
        if(re.search(unique_investors[i],investors[j])):
            investors[j]=unique_investors[i]
```

FILLING MISSING VALUES IN AmountInUSD

AmountinUSD column is formatted to integer

In [26]:
```
def convert_AmountInUSD(x):
    if re.search(',',x):
        return (x.replace(',',''))
    return x
newdf['AmountInUSD']=newdf[newdf['AmountInUSD'].notnull()]['AmountIn
```

In [27]:
```
newdf['AmountInUSD']=round(newdf['AmountInUSD'].fillna(np.mean(newdf
newdf['AmountInUSD']=newdf['AmountInUSD'].astype('int')
```

FILLING MISSING VALUES IN InvestmentType

In [28]:
```
#filling missing valuse in InvestmetnType
newdf['InvestmentType'].fillna(method='bfill',inplace=True)#backward
```

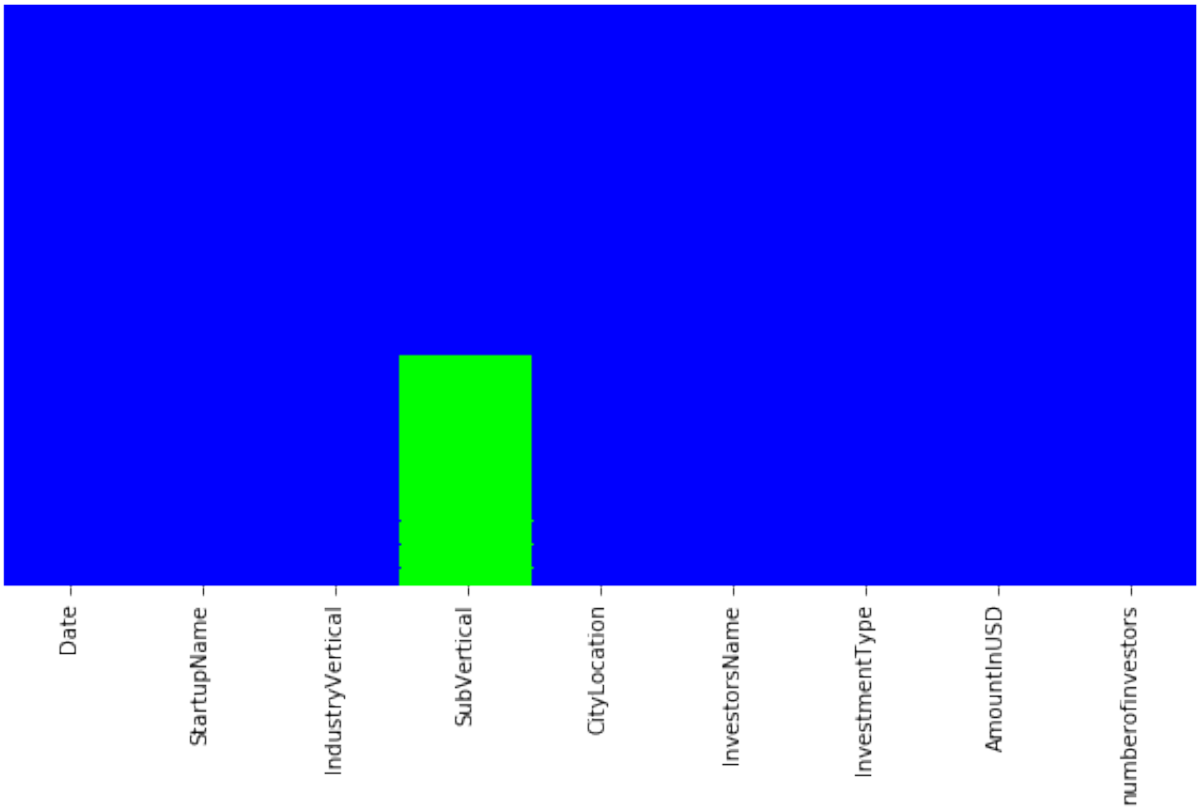
Converting data into lower case to avoid duplicacy

In [29]:
```
newdf.iloc[:,[1,2,3,4,6]]=newdf.iloc[:,[1,2,3,4,6]].applymap(lambda
```

In [30]:
```
def check(x):
    if(pd.notnull(x)):
        return x.lower()
newdf.iloc[:,3]=newdf.iloc[:,3].apply(check)
```

Checking for NaN values after Cleansing

In [31]:
```
plt.figure(figsize=(10,5))
sns.heatmap(newdf.isnull(),cmap='brg',yticklabels=False,cbar=False)
plt.show()
```



Removing ambigous records(startup names like flipkart and flipkart.com

In [32]:
```
unique_startup_name=list(newdf['StartupName'].unique())
startupname=list(newdf['StartupName'])
```

In [33]:
```
for i in range(len(unique_startup_name)):
    for j in range(len(startupname)):
        if(re.search(unique_startup_name[i],startupname[j])):
            startupname[j]=unique_startup_name[i]
```

In [34]:
```
newdf['StartupName']=startupname
```

In [35]:
```
newdf.head(10)
```

	Date	StartupName	IndustryVertical	SubVertical	CityLocation
0	2017-08-01	touchkin	technology	predictivecareplatform	bangalore
1	2017-08-02	ethinos	technology	digitalmarketingagency	mumbai
2	2017-08-02	leverageedu	consumerinternet	onlineplatformforhighereducationservices	newdelhi
3	2017-08-02	zepo	consumerinternet	diyecommerceplatform	mumbai
4	2017-08-02	click2clinic	consumerinternet	healthcareserviceaggregator	hyderabad
5	2017-07-01	billionloans	consumerinternet	peertopeerlendingplatform	bangalore
6	2017-07-03	ecolibriumenergy	technology	energymanagementsolutionsprovider	ahmedabad
7	2017-07-04	droom	ecommerce	onlinemarketplaceforautomobiles	gurgaon
8	2017-07-05	jumbotail	ecommerce	onlinemarketplaceforfoodandgrocery	bangalore
9	2017-07-05	moglix	ecommerce	b2bmarketplaceforindustrialproducts	noida

In [64]:
```
print(newdf['StartupName'].nunique())
```

```
1792
```

Top 10 startups had most funding

paytm and flipkart were on the top of the run

In [65]:
```
tp10fund=show(newdf.groupby('StartupName')['AmountInUSD'].sum().sort
tp10fund.head(10)
```

StartupName	AmountInUSD
paytm	2364062146
flipkart	2259700000
ola	2001391292
snapdeal	700000000
oyo	661062146
quikr	230000000
delhivery	215000000
cartrade	212031073
foodpanda	210000000
shopclues	207700000

DETAILS OF TOP 10 STARTUPS AS PER THE FUND GENERATED

Below is the details about the top 10 startups on different Dates

In [66]:

```
def find(x):
    if x in tp10fund.head(10).index:
        return True
    return False

n=newdf[newdf['StartupName'].apply(find)]
print('amount funded on top 10 startups')
n.describe().iloc[:,0]
```

```
amount funded on top 10 startups


count    5.800000e+01
mean     1.562232e+08
std      2.859792e+08
min      1.470000e+05
25%      1.203107e+07
50%      5.750000e+07
75%      1.362500e+08
max      1.400000e+09
Name: AmountInUSD, dtype: float64
```

Which kind of investment did the top10 startups got

```
The top 10 investments were pivate equtiy and hence seed funding i
```

AS THE TOP 10 FUNDNDING AMOUNT ARE RECEIVED THROUGH PRIVATE
10 STARTUPS AS PER PRIVATE EQUITY BELOW THE PRIVATE EQUITY CO
COUNT OF INVESTMENT TYPE

In [67]:
```
pd.crosstab(n['StartupName'],columns=n['InvestmentType']).sort_value
```

InvestmentType	privateequity	seedfunding
StartupName		
ola	16	7
oyo	6	2
paytm	6	0
flipkart	5	0
cartrade	3	0
delhivery	3	0
quikr	3	0
shopclues	3	0
foodpanda	2	0
snapdeal	2	0

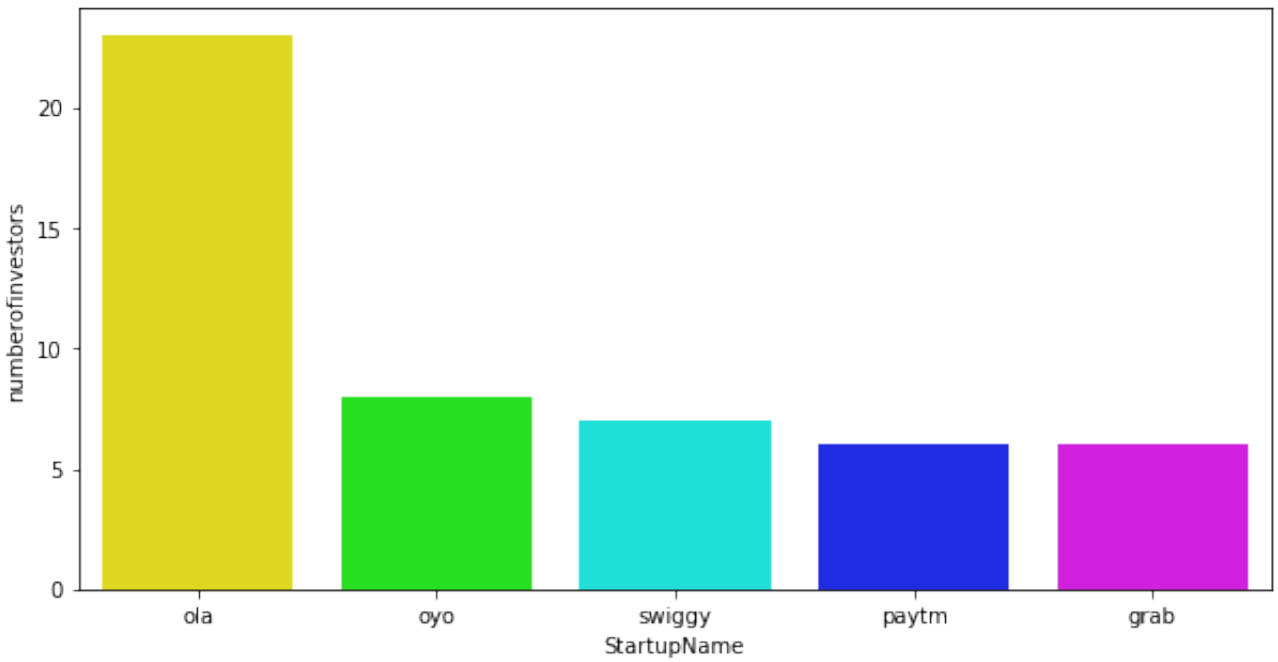
Insights regarding the best startup in terms of funding amount

In [68]:
```
newdf[newdf['StartupName']=='paytm']
```

	Date	StartupName	IndustryVertical	SubVertical	CityLocation	Inv
158	2017-05-18	paytm	ecommerce	mobilewallet&ecommerceplatform	bangalore	Softl
266	2017-03-03	paytm	ecommerce	ecommercemarketplace	bangalore	Aliba
821	2016-08-30	paytm	ecommerce	mobilewallet&ecommerceplatform	bangalore	Med
1787	2015-09-29	paytm	e-commerce&m-commerceplatform	None	newdelhi	Aliba Fina
2218	2015-03-13	paytm	other	None	notspecific	Rata
2276	2015-02-05	paytm	other	None	notspecific	Ant Serv

Companies with most number of investors

In [69]:
```
#Companies with most number of investors
cmi=show(newdf.groupby('StartupName')['numberofinvestors'].count().s
fig=plt.figure(figsize=(10,5))
sns.barplot(y='numberofinvestors',x='StartupName',data=cmi.reset_ind
plt.show()
cmi.head(10)
```

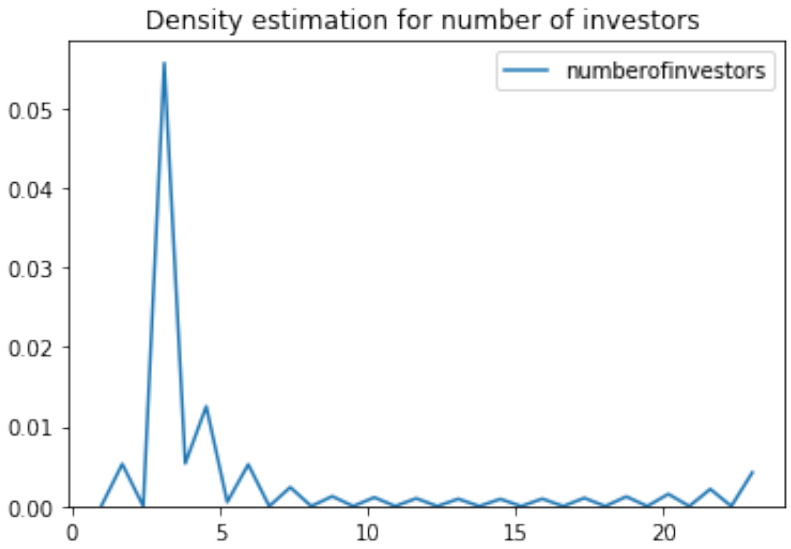


StartupName	numberofinvestors
ola	23
oyo	8
swiggy	7
paytm	6
grab	6
urbanclap	6
medinfi	5
stalkbuylove	5
lenskart	5
faircent	5

Here we can see that kernel density of startups having two and three nu

In [70]:
```
#Here we can see that kernel density of startups having two and thre
sns.kdeplot(data=cmi.reset_index()['numberofinvestors'],gridsize=20,
plt.title('Density estimation for number of investors ')
plt.show()
```

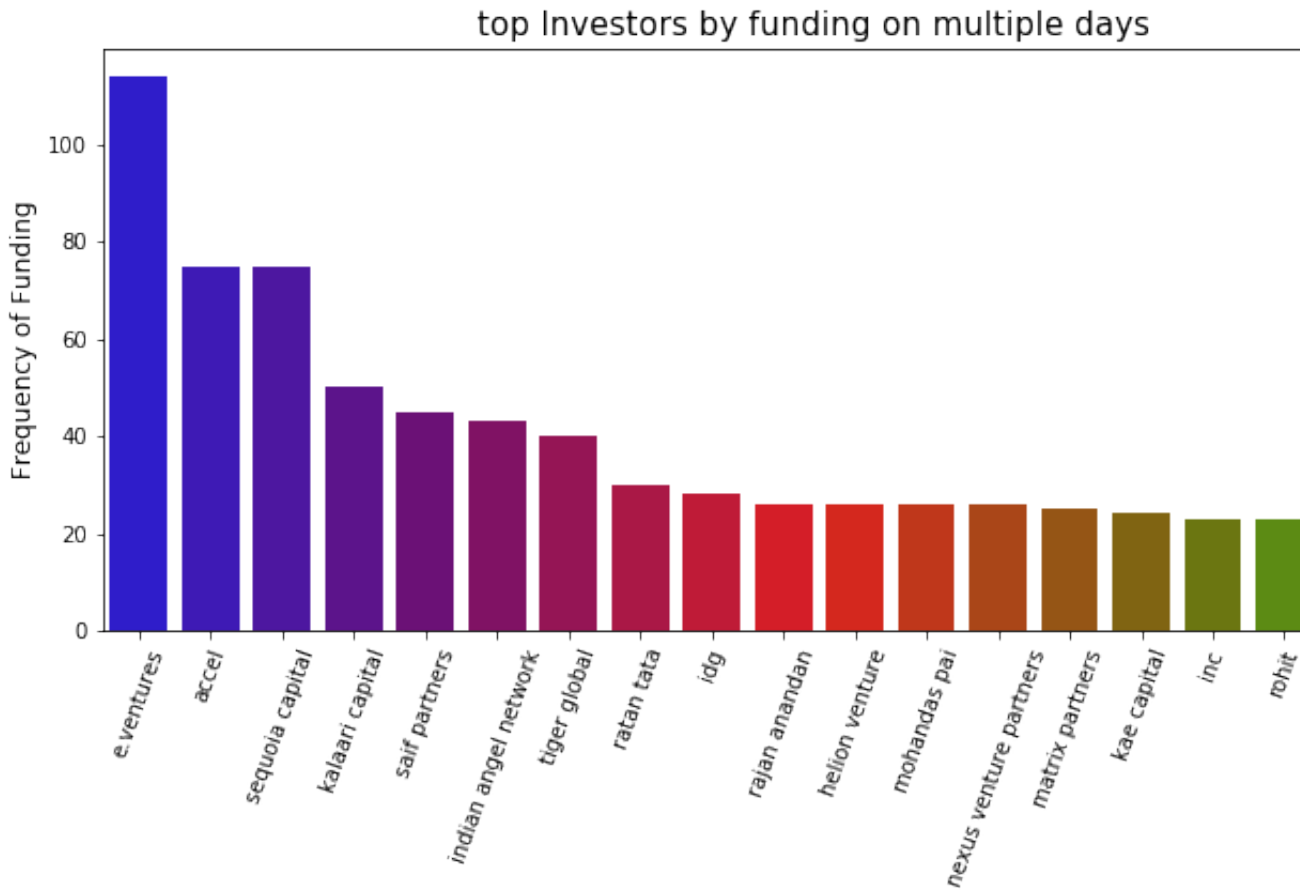
```
/miniconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a nc
ndexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will
np.array(seq)]`, which will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



Investors with most funding frequency

In [71]:
```
cinvestors=show(investors)[0].value_counts()[2:]
cinvestors.head(10)
print("Top Investors in Frequency ")
plt.figure(figsize = (12,5))
bar= sns.barplot(x=cinvestors.index[:20],y=cinvestors.values[:20],pa
bar.set_xticklabels(bar.get_xticklabels(),rotation=70)
bar.set_title("top Investors by funding on multiple days ", fontsize
bar.set_xlabel("", fontsize=12)
bar.set_ylabel("Frequency of Funding", fontsize=12)
plt.show()
```

```
Top Investors in Frequency
```



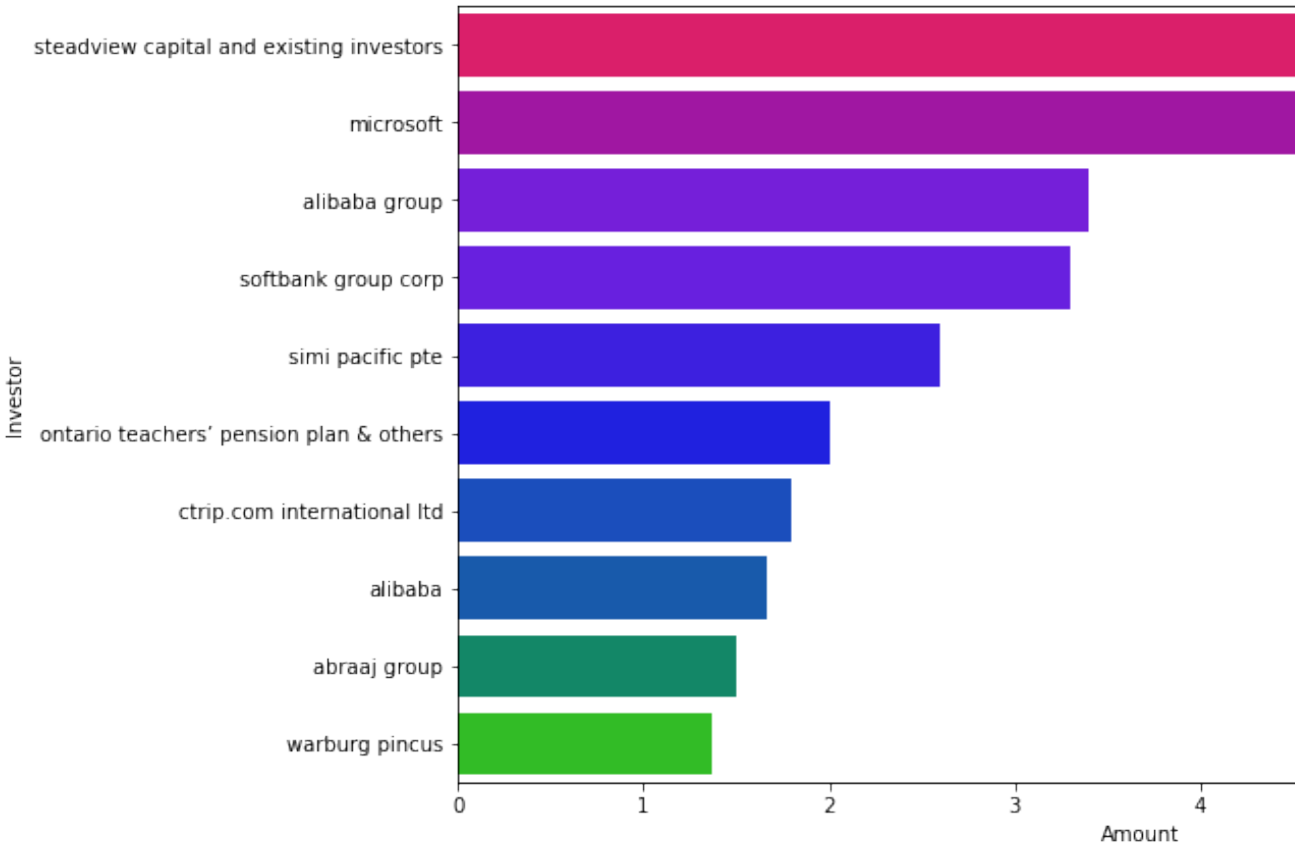
Top 10 Investors with highest funding amount

In [47]:
```
Investor_amount=pd.Series(d,name='Amount')

Investor_amount=show(Investor_amount,).reset_index().groupby('index'
Investor_amount=show(Investor_amount).reset_index()
Investor_amount.columns=["Investor","Amount"]
```

In [72]:
```
print('Top 10 Most funded Investors')
plt.figure(figsize=(12,7))
sns.barplot(y='Investor',x='Amount',data=Investor_amount.head(10),pa
print(Investor_amount.head(10))
plt.show()
```

```
Top 10 Most funded Investors
                                  Investor        Amount
0  steadview capital and existing investors  7.000000e+08
1                                microsoft  4.666667e+08
2                           alibaba group   3.400000e+08
3                      softbank group corp  3.300000e+08
4                         simi pacific pte  2.600000e+08
5   ontario teachers' pension plan & others  2.000000e+08
6              ctrip.com international ltd   1.800000e+08
7                                  alibaba  1.666667e+08
8                             abraaj group  1.500000e+08
9                           warburg pincus  1.370000e+08
```

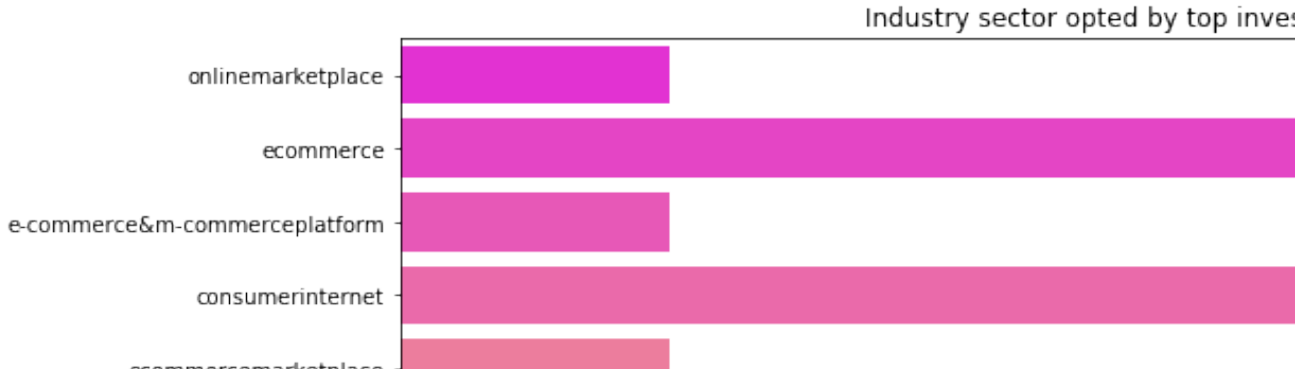


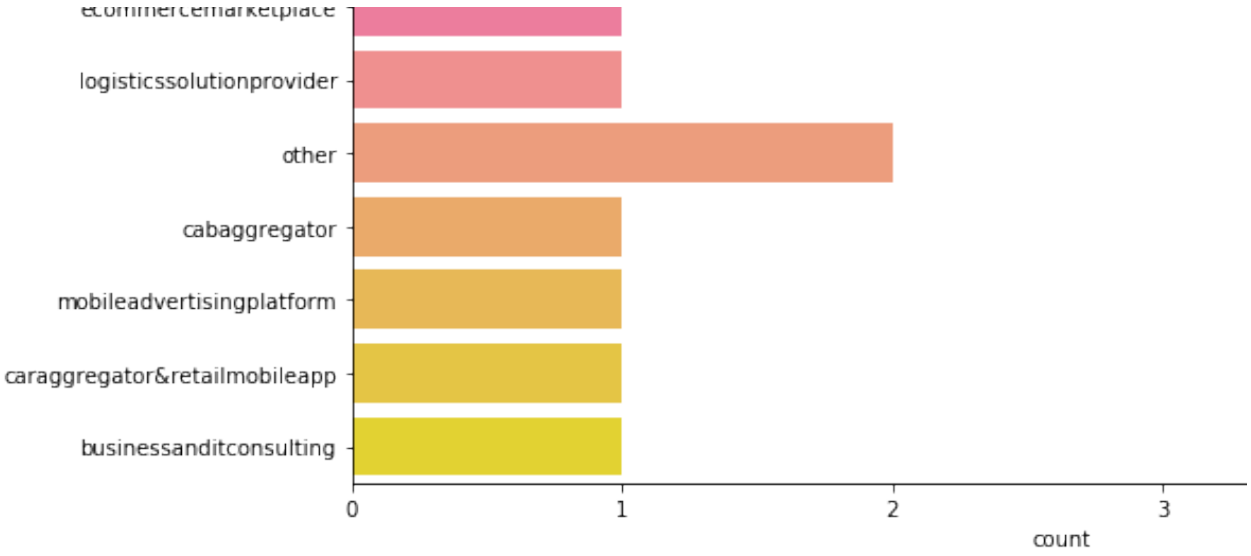
Which industry vertical opted by top investors

In [51]:
```
top_industry_vertical={}
for i in Investor_amount['Investor'].head(20):
    for j in range(len(listed_n_inv)):
        if i in listed_n_inv[j]:
            top_industry_vertical[i]=newdf['IndustryVertical'][j]
```

In [74]:
```
plt.figure(figsize=(12,7))
sns.countplot(y=pd.Series(top_industry_vertical),palette='spring')
plt.title('Industry sector opted by top investors' )
print('top investor\'s favourite Industry  ')
print(pd.Series(top_industry_vertical))
plt.show()
```

```
top investor's favourite Industry
steadview capital and existing investors              onlinemarketplace
microsoft                                                    ecommerce
alibaba group                                 e-commerce&m-commerceplatform
softbank group corp                                   consumerinternet
simi pacific pte                                      consumerinternet
ontario teachers' pension plan & others                      ecommerce
ctrip.com international ltd                           consumerinternet
alibaba                                           ecommercemarketplace
abraaj group                                                 ecommerce
warburg pincus                                 logisticssolutionprovider
rocket internet ag & others                                      other
dst global                                                cabaggregator
tiger global & other investors                                   other
tennenbaum capital partners & others               mobileadvertisingplatform
baillie gifford                               caraggregator&retailmobileapp
clairvest group                                       consumerinternet
naspers                                                      ecommerce
chryscapital                                    businessanditconsulting
softbank vision fund                                  consumerinternet
mediatek inc.                                                ecommerce
dtype: object
```





Total amount of funding recieved as per investment type

In [76]:
```
newdf.groupby('InvestmentType').sum()['AmountInUSD']
```

```
InvestmentType
crowdfunding          155768
debtfunding          7800000
nan                 12031073
privateequity    20882511447
seedfunding       7635207019
Name: AmountInUSD, dtype: int64
```

Total number of funding recieved as per investment type

In [75]:
```
newdf['InvestmentType'].value_counts()
```

```
seedfunding      1301
privateequity    1067
crowdfunding        2
nan                 1
debtfunding         1
Name: InvestmentType, dtype: int64
```

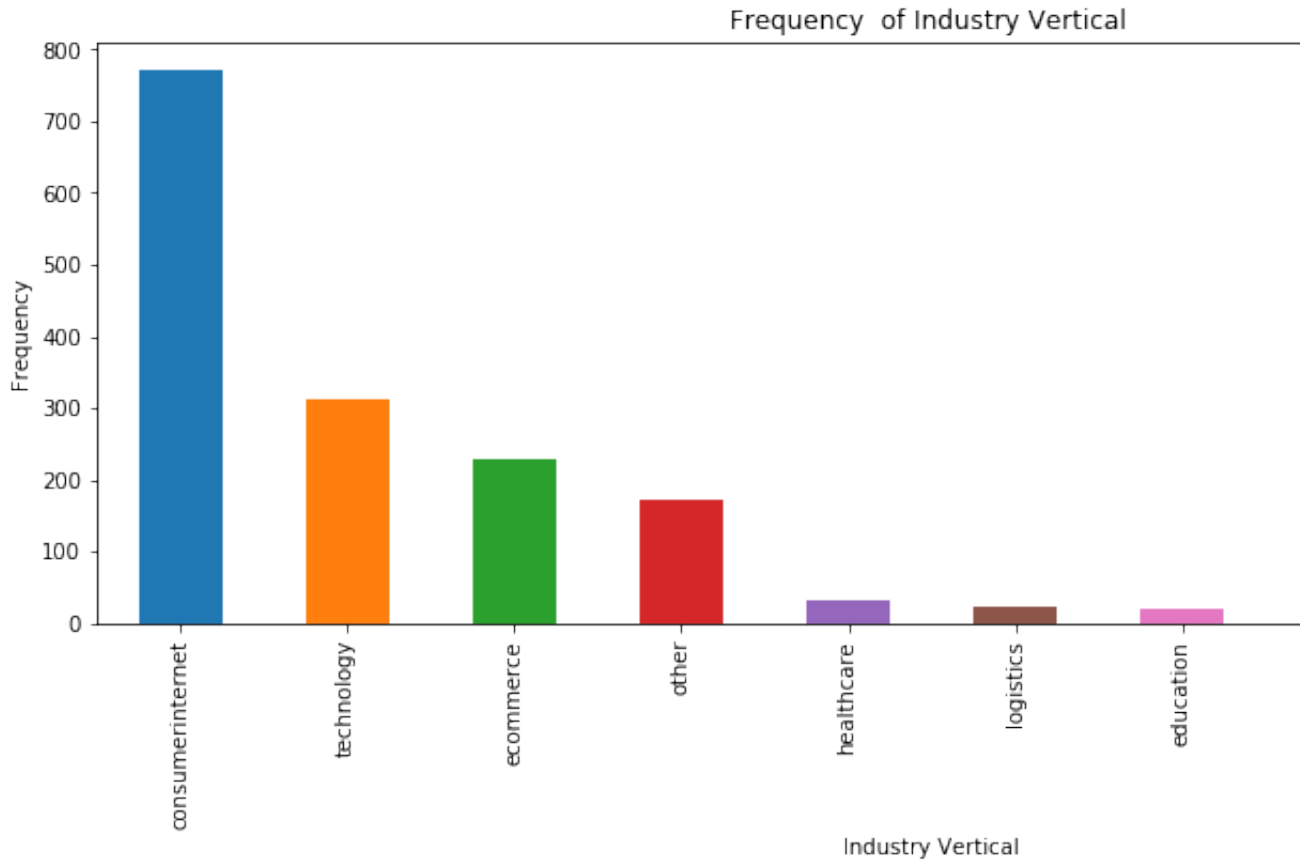
Top 10 industry sectors with most funding

visualization of the growth rate of each sector (industry vertical)

Here we can see that consumer internet sector has more funding (as
Here we can see that consumer internet sector has most frequency
It also seen that debt funding and crowd funding are negligible as
Investment types

In [79]:
```
plt.figure(figsize=(14,5))
iv=newdf['IndustryVertical'].value_counts().head(10)
iv.plot.bar()

plt.title('Frequency  of Industry Vertical ')
plt.ylabel('Frequency')
plt.xlabel('Industry Vertical')
plt.show()
```

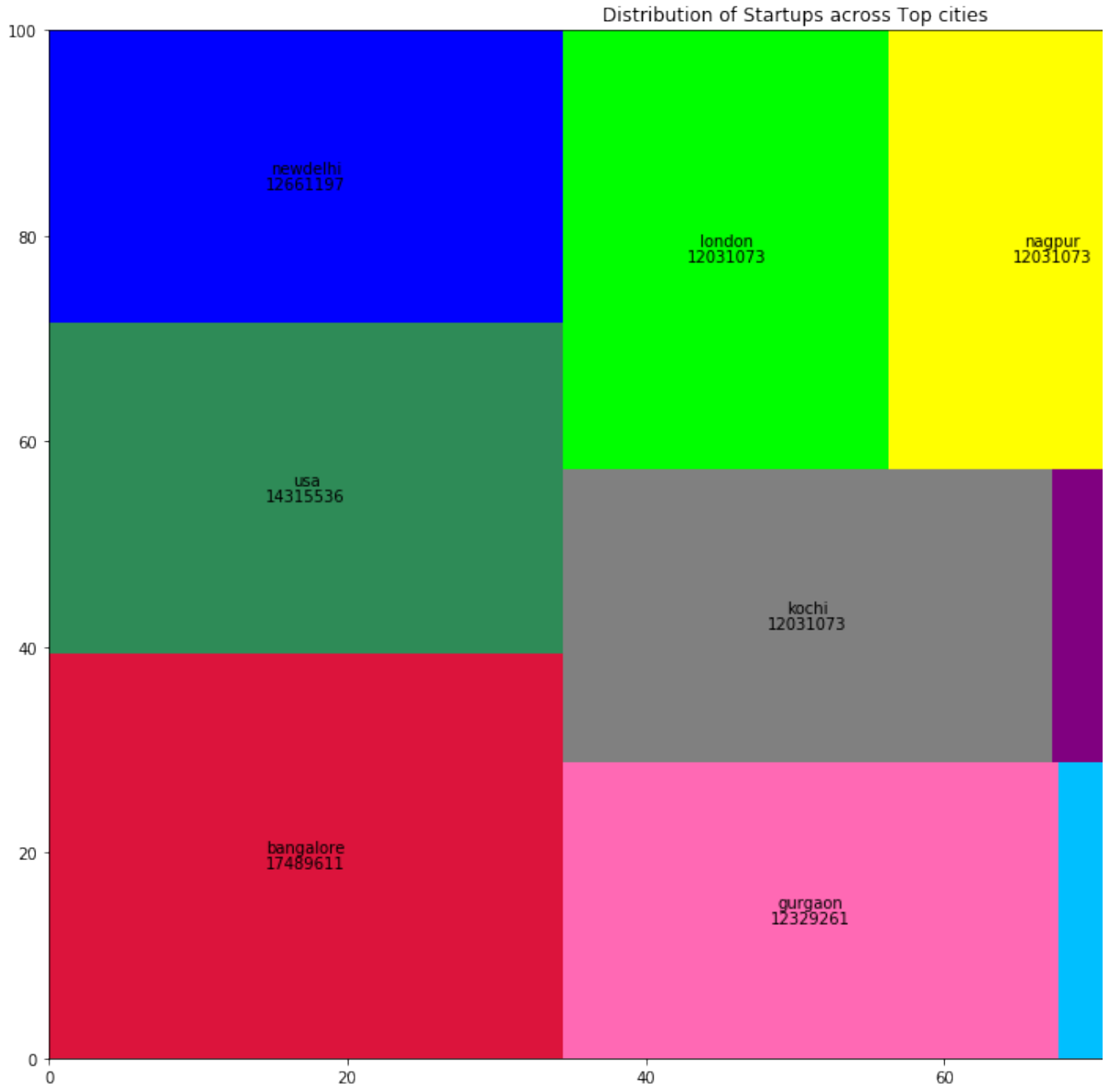


Most preferrable cities as per Investment on startups

Insights provided shows that Banglore has the most average funding Amount

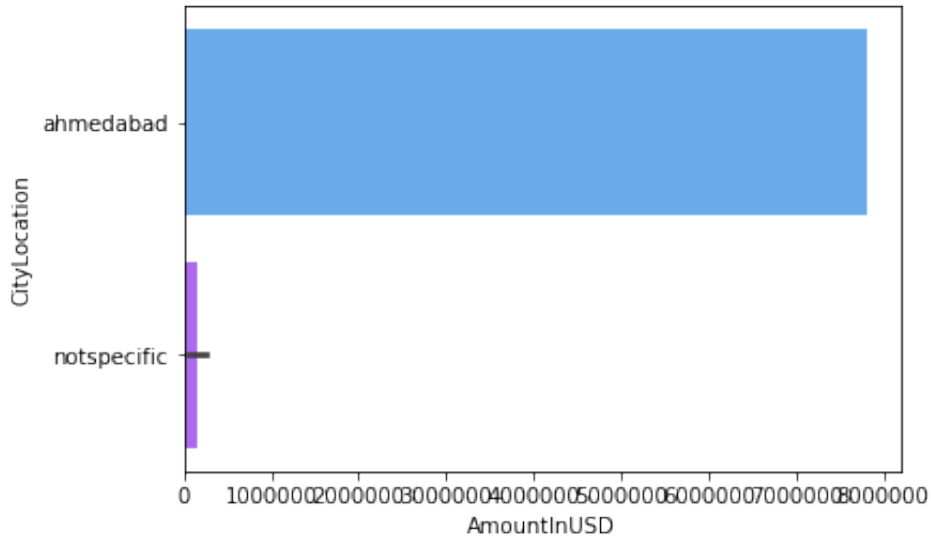
In [82]:
```
plt.figure(figsize=(17,12))
mean_amount = newdf.groupby('CityLocation').mean()["AmountInUSD"].as
squarify.plot(sizes=mean_amount.values,label=mean_amount.index, valu
plt.title('Distribution of Startups across Top cities')
```

```
Text(0.5, 1.0, 'Distribution of Startups across Top cities')
```



Here we can see that Ahmedabad is a Market place for dept-funding

In [89]:
```
sns.barplot(y='CityLocation',x='AmountInUSD',data=newdf[(newdf['Inve
#average investment in banglore is most
plt.show()
#amehdabad is the market place for dept funding
```



In [ ]: