Shark Tank India Investments Analysis

Shark Tank India is an Indian Hindi-language business reality television series that airs on Sony Entertainment Television. The show is the Indian franchise of the American show Shark Tank. It shows entrepreneurs making business presentations to a panel of investors or sharks, who decide whether to invest in their company. The first season of Shark Tank India premiered on 20 December 2021 and concluded on 4 February 2022

Libraries required for the Analysis

In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import plotly.offline as pyo
import plotly.graph_objs as go
import plotly.express as px
```

In [3]:
```
df = pd.read_csv("Shark Tank India.csv")
df.head()
```

Out[3]:

	Season Number	Episode Number	Episode Title	Pitch Number	Startup Name	Industry	Business Description	Company Website	Number of Presenters	Male Presenters	...	Aman Investment Amount	Aman Investment Equity	Aman Debt Amount	Peyush Investment Amount	Peyush Investment Equity	Peyush Debt Amount	Gha Investm Amo
0	1	1	Badlegi Business Ki Tasveer	1	BluePine Foods	Food	Frozen Momos	https://bluepinefoods.com/	3	2.0	...	25.0	5.33	NaN	NaN	NaN	NaN	N
1	1	1	Badlegi Business Ki Tasveer	2	Booz Scooters	Electrical Vehicles	Renting e-bike for mobility in private spaces	https://www.boozup.net/	1	1.0	...	0.0	0.00	NaN	NaN	NaN	NaN	N
2	1	1	Badlegi Business Ki Tasveer	3	Heart up my Sleeves	Beauty/Fashion	Detachable Sleeves	https://heartupmysleeves.com/	1	NaN	...	0.0	0.00	NaN	NaN	NaN	NaN	N
3	1	2	Insaan, Ideas Aur Sapne	4	Tagz Foods	Food	Healthy Potato Chips Snacks	https://tagzfoods.com/	2	2.0	...	0.0	0.00	NaN	NaN	NaN	NaN	N
4	1	2	Insaan, Ideas Aur Sapne	5	Head and Heart	Education	Brain Development Course	https://thehnh.in/	4	1.0	...	NaN	NaN	NaN	NaN	NaN	NaN	N

5 rows × 50 columns

Data Types of the Columns

In [5]:
```
df.dtypes
```

```
Namita Investment Equity     float64
Namita Debt Amount           float64
Anupam Investment Amount     float64
Anupam Investment Equity     float64
Anupam Debt Amount           float64
Vineeta Investment Amount    float64
Vineeta Investment Equity    float64
Vineeta Debt Amount          float64
Aman Investment Amount       float64
Aman Investment Equity       float64
Aman Debt Amount             float64
Peyush Investment Amount     float64
Peyush Investment Equity     float64
Peyush Debt Amount           float64
Ghazal Investment Amount     float64
Ghazal Investment Equity     float64
Ghazal Debt Amount           float64
Number of sharks in deal     float64
dtype: object
```

In [16]:
```
# No of Episodes unique gives the list of items where as nuniquew gives the num of unique number
print(df["Episode Number"].unique())
print(df["Episode Number"].nunique())
```

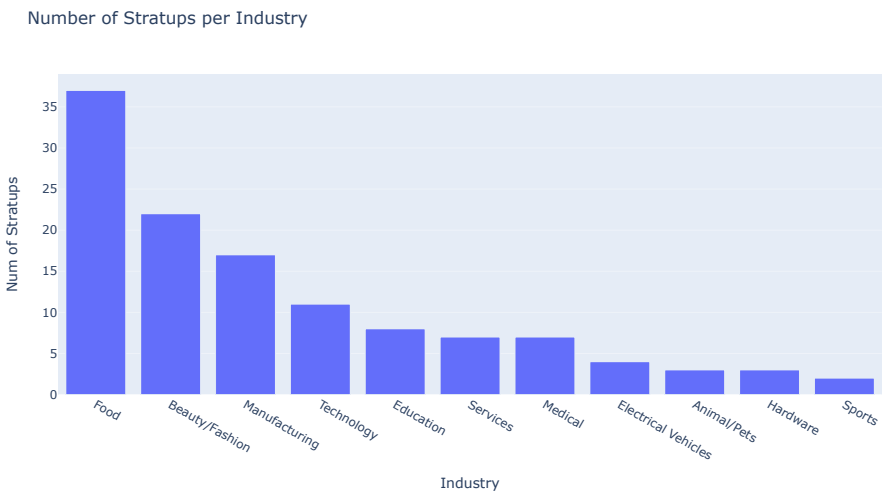
```
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
 25 26 27 28 29 30 31 32 33 34 35 36]
36
```

Number of Stratups from different Industry

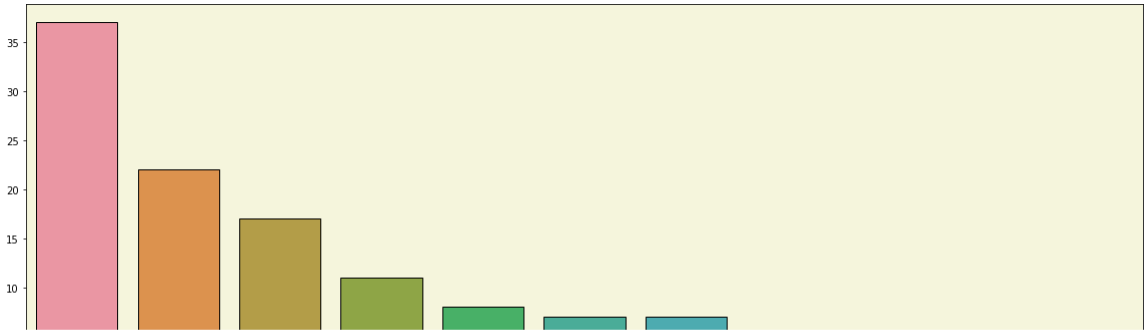
In [8]:
```
print("Number of Stratups from Different Industries", '\n','-'*45, '\n', df['Industry'].value_counts())
```

```
Number of Stratups from Different Industries
 ---------------------------------------------
 Food                  37
Beauty/Fashion         22
Manufacturing          17
Technology             11
Education               8
Services                7
Medical                 7
Electrical Vehicles     4
Animal/Pets             3
Hardware                3
Sports                  2
Name: Industry, dtype: int64
```

In [9]:
```
fig = go.Figure()
fig.add_trace(go.Bar(x=x.index, y =x.values))
fig.update_layout(title='Number of Stratups per Industry', xaxis_title = "Industry", yaxis_title= "Num of Stratups")
```



In [85]:
```
plt.figure(figsize=(20,7))
sns.barplot(x=x.index, y=x.values, ec='k').set_facecolor("beige")
plt.tick_params(axis = 'x', labelrotation=45)

```



In [94]:
```
df.columns
```

Out[94]:
```
Index(['Season Number', 'Episode Number', 'Episode Title', 'Pitch Number',
       'Startup Name', 'Industry', 'Business Description', 'Company Website',
       'Number of Presenters', 'Male Presenters', 'Female Presenters',
       'Couple Presenters', 'Pitchers Average Age', 'Started in',
       'Pitchers City', 'Pitchers State', 'Yearly Revenue', 'Monthly Sales',
       'Gross Margin', 'Original Ask Amount', 'Original Ask Equity',
       'Valuation Requested', 'Received Offer', 'Accepted Offer',
       'Total Deal Amount', 'Total Deal Equity', 'Total Deal Debt',
       'Valuation Offered', 'Ashneer Investment Amount',
       'Ashneer Investment Equity', 'Ashneer Debt Amount',
       'Namita Investment Amount', 'Namita Investment Equity',
       'Namita Debt Amount', 'Anupam Investment Amount',
       'Anupam Investment Equity', 'Anupam Debt Amount',
       'Vineeta Investment Amount', 'Vineeta Investment Equity',
       'Vineeta Debt Amount', 'Aman Investment Amount',
       'Aman Investment Equity', 'Aman Debt Amount',
       'Peyush Investment Amount', 'Peyush Investment Equity',
       'Peyush Debt Amount', 'Ghazal Investment Amount',
       'Ghazal Investment Equity', 'Ghazal Debt Amount',
```

Which State has highest number stratups in Shark Tank

In [98]:
```
df['Pitchers State'].value_counts()
```

Out[98]:
```
Maharashtra               26
Delhi                     13
Gujarat                   11
Karnataka                  6
Telangana                  5
West Bengal                4
Uttar Pradesh              4
Jammu & Kashmir            2
Tamil Nadu                 2
Kerala                     2
Punjab                     2
Bihar                      1
Rajasthan                  1
Madhya Pradesh             1
Haryana                    1
Karnataka, West Bengal     1
Punjab, Delhi              1
Uttarakhand                1
Goa                        1
Maharashtra, Delhi         1
Name: Pitchers State, dtype: int64
```

How many Starups Affected/Rejected the offer?

In [10]:
```
df['Received Offer'].replace(1, "Offer Recieved", inplace=True)
df['Received Offer'].replace(0, "Offer Not Recieved", inplace=True)
df['Accepted Offer'].replace(0, "Offer Declined", inplace=True)
df['Accepted Offer'].replace(1, "Offer Accepted", inplace=True)
p = df['Received Offer'].value_counts()
u = df['Accepted Offer'].value_counts()
print("Number of Stratups Recieved Offer: ", p[0])
print("Number of Stratups Not Recieved Offer: ", p[1])
print("Number of Stratups Accepted Offer: ", u[0])
print("Number of Stratups Not Accepted Offer: ", u[1])

plt.figure(figsize=(15,8))
ax1 = plt.subplot(221)
p.plot(kind='bar', color = ['green', 'red'])
plt.xticks(rotation = 0)


ax2 = plt.subplot(222)
p.plot(kind='pie', autopct ='%.2f', explode = (0,0.05), colors=(['green', 'red']))

ax3 = plt.subplot(223)
u.plot(kind='bar', color = ['green', 'red'])
plt.xticks(rotation = 0)

ax4 = plt.subplot(224)
u.plot(kind='pie', autopct ='%.2f', explode = (0,0.05), colors=(['green', 'red']))
```

```
Number of Stratups Recieved Offer:  88
Number of Stratups Not Recieved Offer:  33
Number of Stratups Accepted Offer:  67
Number of Stratups Not Accepted Offer:  21
```

Out[10]: <AxesSubplot:ylabel='Accepted Offer'>

In [140]:
```
1
```

```
Number of Stratups Recieved Offer:  88
Number of Stratups Not Recieved Offer:  33
Number of Stratups Accepted Offer:  67
Number of Stratups Not Accepted Offer:  21
```

In [149]:
```
plt.figure(figsize=(15,8))
ax1 = plt.subplot(221)
p.plot(kind='bar', color = ['green', 'red'])
plt.xticks(rotation = 0)


ax2 = plt.subplot(222)
p.plot(kind='pie', autopct ='%.2f', explode = (0,0.05), colors=(['green', 'red']))

ax3 = plt.subplot(223)
u.plot(kind='bar', color = ['green', 'red'])
plt.xticks(rotation = 0)

ax4 = plt.subplot(224)
u.plot(kind='pie', autopct ='%.2f', explode = (0,0.05), colors=(['green', 'red']))
```

Out[149]: <AxesSubplot:ylabel='Accepted Offer'>

In [154]:
```
df['Valuation Offered']
```

Out[154]: 56977.0

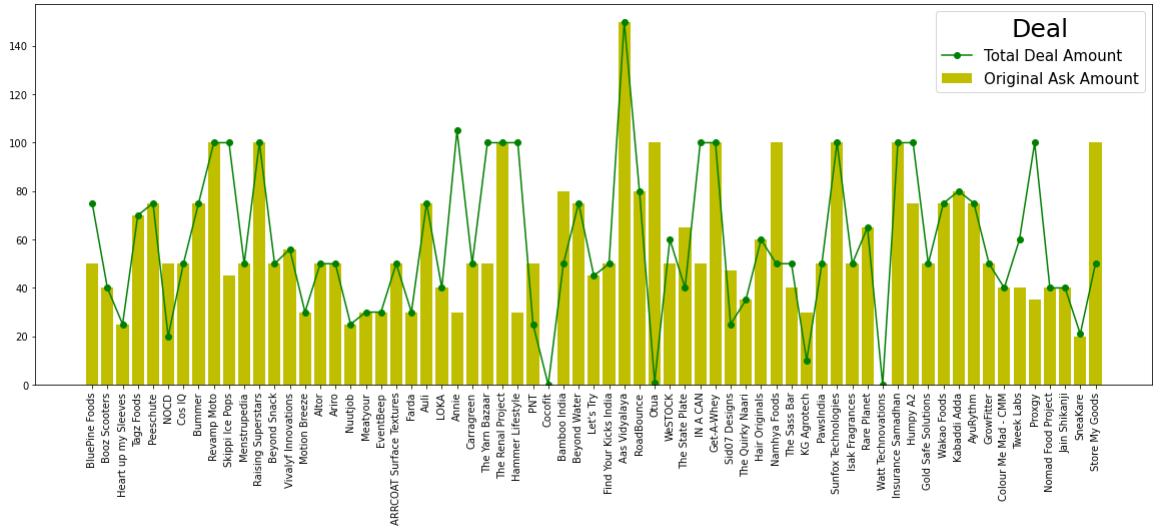
In [ ]:
```
1
```

Was their a bargin on deal amount?

In [16]:
```
# Removing null value records
v = df.loc[(df['Valuation Offered'].isnull()==False)]

# Changing the data type to int
v["Valuation Offered"]=v["Valuation Offered"].astype(np.int64)

#Plotting the garph
plt.figure(figsize=(20,7))
plt.bar(v['Startup Name'], v['Original Ask Amount'], color='y', label='Original Ask Amount')
plt.plot(v['Startup Name'], v["Total Deal Amount"], marker = 'o', color='g', label='Total Deal Amount')
plt.tick_params(axis = 'x', labelrotation=90)
plt.xlabel("Stratup", labelpad=25, fontsize=30)
plt.legend(fontsize=15, title="Deal", title_fontsize=25)
```

```
C:\Users\Premk\AppData\Local\Temp/ipykernel_10588/3309688150.py:5: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
```

Out[16]: <matplotlib.legend.Legend at 0x1797981ca90>

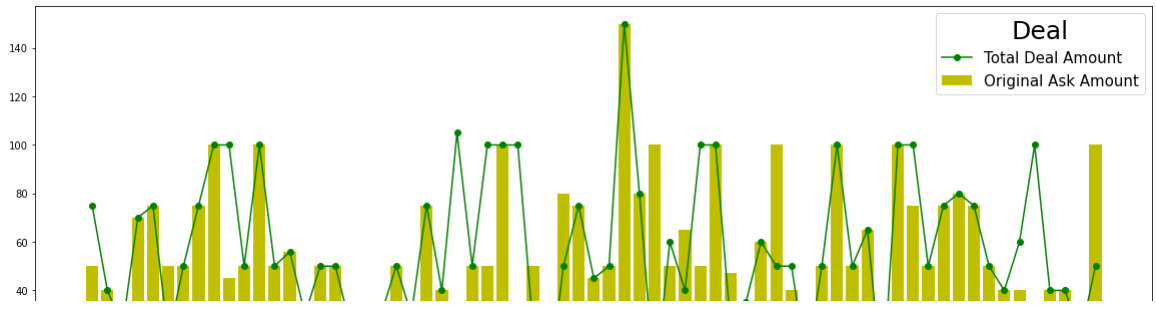


In [14]:
```
v["Valuation Offered"]=v["Valuation Offered"].astype(np.int64)
```

```
C:\Users\Premk\AppData\Local\Temp/ipykernel_10588/1212485624.py:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
```

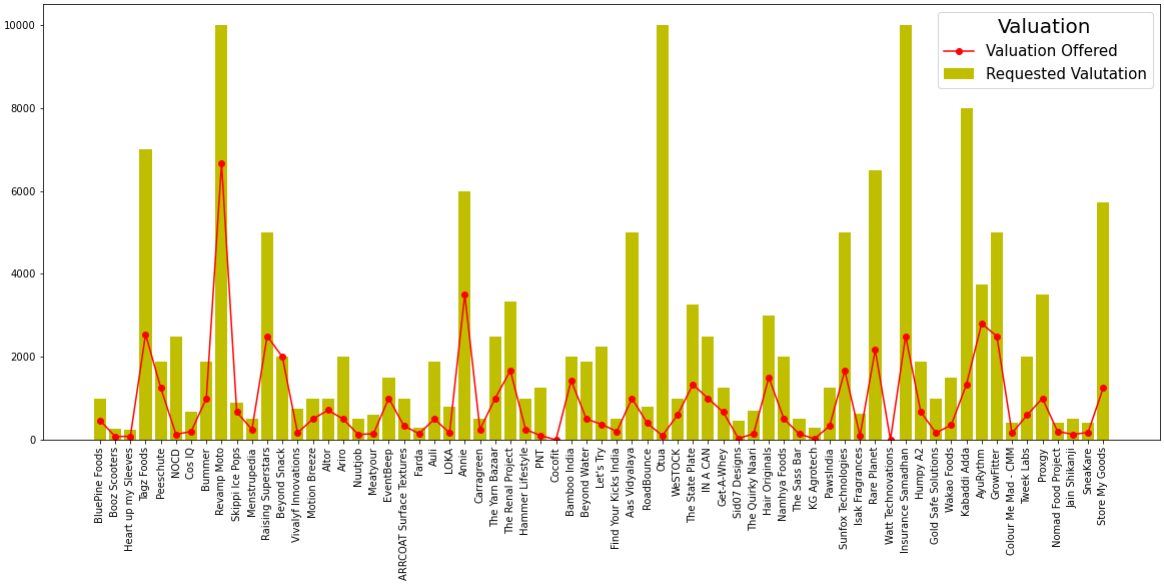
In [15]:
```
1
```

Out[15]: <matplotlib.legend.Legend at 0x1797963a6d0>



Valuation requested vs Valuation Offered

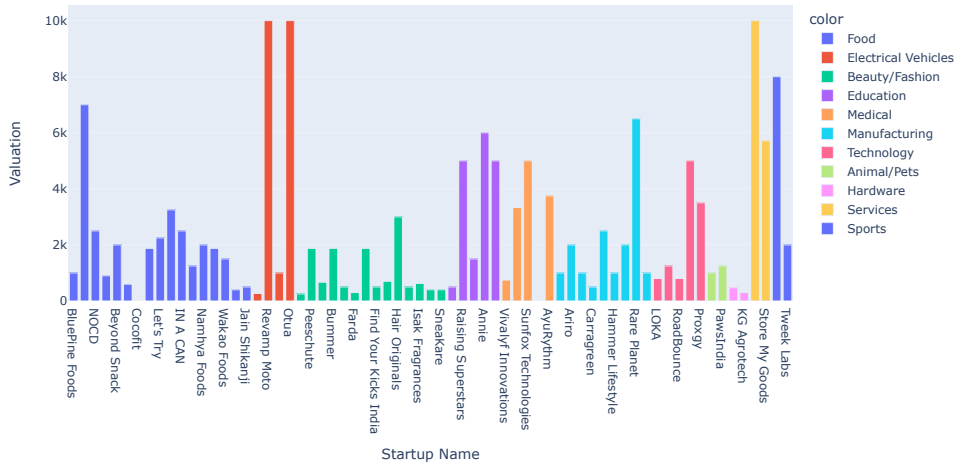
In [17]:
```
plt.figure(figsize=(20,8))
plt.bar(v['Startup Name'], v['Valuation Requested'], label ='Requested Valutation', color='y')
plt.plot(v['Startup Name'], v['Valuation Offered'], label='Valuation Offered', color='r', marker='o')
plt.legend(fontsize=15, title="Valuation", title_fontsize=20)
plt.tick_params(axis='x', labelrotation=90)
```



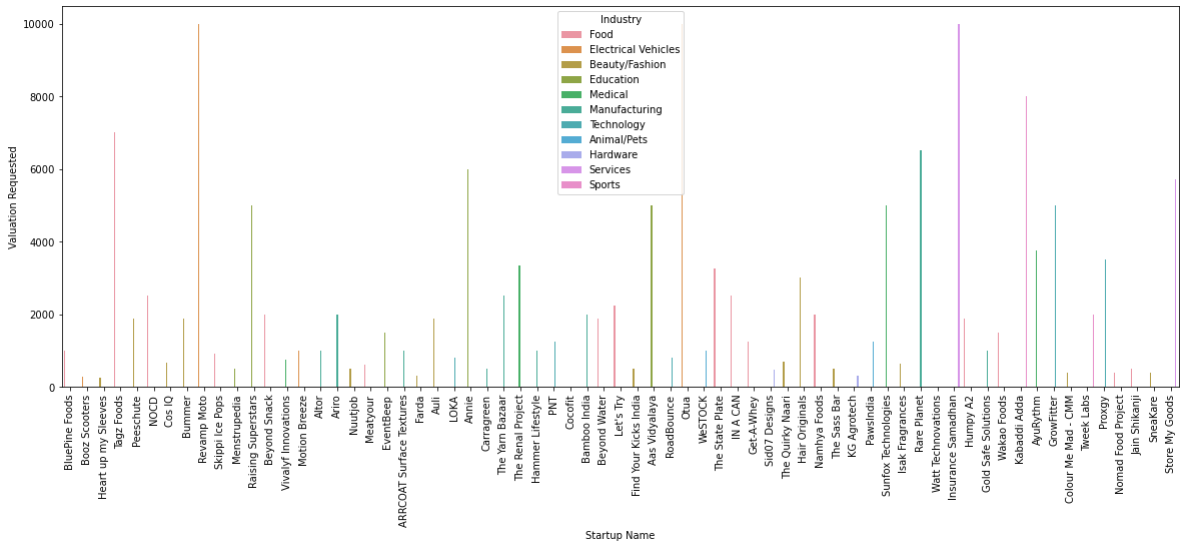
In [18]:
```
fig = go.Figure()
fig.add_trace(go.Bar(x=v['Startup Name'], y=v['Valuation Requested']))
fig.add_trace(go.Line(x=v['Startup Name'], y=v['Valuation Offered']))
fig.update_layout(title="Valuation Game", xaxis_title="Stratup", yaxis_title = "Valuation", showlegend=False)
```

```
C:\Users\Premk\anaconda3\lib\site-packages\plotly\graph_objs\_deprecations.py:378: DeprecationWarning:

plotly.graph_objs.Line is deprecated.
Please replace it with one of the following more specific types
  - plotly.graph_objs.scatter.Line
  - plotly.graph_objs.layout.shape.Line
  - etc.
```

In [19]:
```
px.bar(x=v['Startup Name'], y=v['Valuation Requested'], color=v['Industry'], labels={"y":'Valuation', "x":"Startup Name"})
```



In [236]:
```
plt.figure(figsize=(20,7))
sns.barplot(x='Startup Name', y='Valuation Requested', hue='Industry',data=v)
plt.tick_params(axis = 'x', labelrotation=90)
```



Number of investments by Sharks

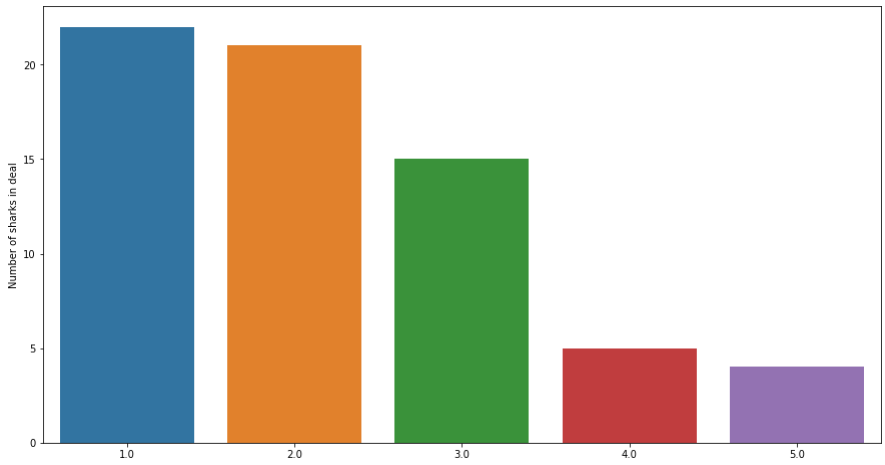
In [241]:
```
Ashneer_amount=df.loc[(df["Ashneer Investment Amount"].isnull()==False)&(df["Ashneer Investment Amount"]!=0)]
Namita_amount=df.loc[(df["Namita Investment Amount"].isnull()==False)&(df["Namita Investment Amount"]!=0)]
Anupam_amount=df.loc[(df["Anupam Investment Amount"].isnull()==False)&(df["Anupam Investment Amount"]!=0)]
Vineeta_amount=df.loc[(df["Vineeta Investment Amount"].isnull()==False)&(df["Vineeta Investment Amount"]!=0)]
Aman_amount=df.loc[(df["Aman Investment Amount"].isnull()==False)&(df["Aman Investment Amount"]!=0)]
Peyush_amount=df.loc[(df["Peyush Investment Amount"].isnull()==False)&(df["Peyush Investment Amount"]!=0)]
Ghazal_amount=df.loc[(df["Ghazal Investment Amount"].isnull()==False)&(df["Ghazal Investment Amount"]!=0)]

print("-"*60,"\n","Ashneer invested in",len(Ashneer_amount),"number of business in the season.")
print("Namita invested in",len(Namita_amount),"number of business in the season.")
print("Anupam invested in",len(Anupam_amount),"number of business in the season.")
print("Vineeta invested in",len(Vineeta_amount),"number of business in the season.")
print("Aman invested in",len(Aman_amount),"number of business in the season.")
print("Peyush invested in",len(Peyush_amount),"number of business in the season.")
print("Ghazal invested in",len(Ghazal_amount),"number of business in the season.","\n","-"*60)
```

```
------------------------------------------------------------
 Ashneer invested in 21 number of business in the season.
Namita invested in 24 number of business in the season.
Anupam invested in 24 number of business in the season.
Vineeta invested in 16 number of business in the season.
Aman invested in 29 number of business in the season.
Peyush invested in 28 number of business in the season.
Ghazal invested in 7 number of business in the season.
 ------------------------------------------------------------
```

Number of Sharks in a deal

In [37]:
```
d4=df["Number of sharks in deal"].value_counts()
plt.figure(figsize=(15,8))
sns.barplot(x=d4.index, y=d4)
```

Out[37]:
```
<AxesSubplot:ylabel='Number of sharks in deal'>
```

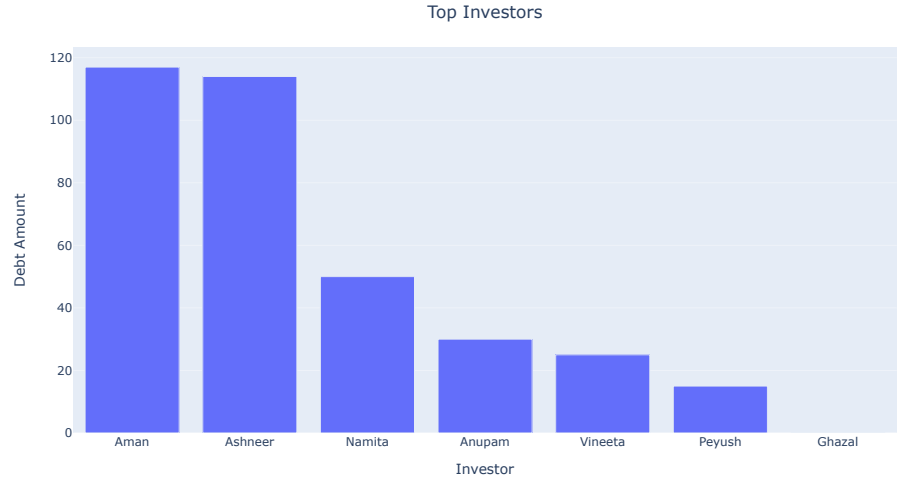


Total Equity Investments by Sharks

In [38]:
```
invest = {"Aman": df['Aman Investment Amount'].sum(),
          "Ashneer": df['Ashneer Investment Amount'].sum(),
          "Namita": df['Namita Investment Amount'].sum(),
          "Anupam": df['Anupam Investment Amount'].sum(),
          "Vineeta": df['Vineeta Investment Amount'].sum(),
          "Peyush": df['Peyush Investment Amount'].sum(),
          "Ghazal": df['Ghazal Investment Amount'].sum()}

df1 = pd.DataFrame.from_dict(invest, orient ='index')
df1 = df1.reset_index()
df1.rename(columns={'index':'Investor', 0:"Investment Amount (M)"}, inplace=True)
df1.sort_values(by='Investment Amount (M)', ascending=False)
fig = px.bar(x=df1['Investor'], y=df1['Investment Amount (M)'].sort_values(ascending=False))
fig.update_layout(title="Top Investors", xaxis_title='Investor', yaxis_title='Investment', title_x = 0.5)
```

Total Debt Investment by Sharks

In [39]:
```
Debt = {"Aman": df['Aman Debt Amount'].sum(),
          "Ashneer": df['Ashneer Debt Amount'].sum(),
          "Namita": df['Namita Debt Amount'].sum(),
          "Anupam": df['Anupam Debt Amount'].sum(),
          "Vineeta": df['Vineeta Debt Amount'].sum(),
          "Peyush": df['Peyush Debt Amount'].sum(),
          "Ghazal": df['Ghazal Debt Amount'].sum()}

df2 = pd.DataFrame.from_dict(Debt, orient ='index')
df2 = df2.reset_index()
df2.rename(columns={'index':'Investor', 0:"Investment Amount (M)"}, inplace=True)
fig = px.bar(x=df2['Investor'], y=df2['Investment Amount (M)'].sort_values(ascending=False))
fig.update_layout(title="Top Investors", xaxis_title="Investor", yaxis_title='Debt Amount', title_x=0.5)
```



How many stratups accepted Debt

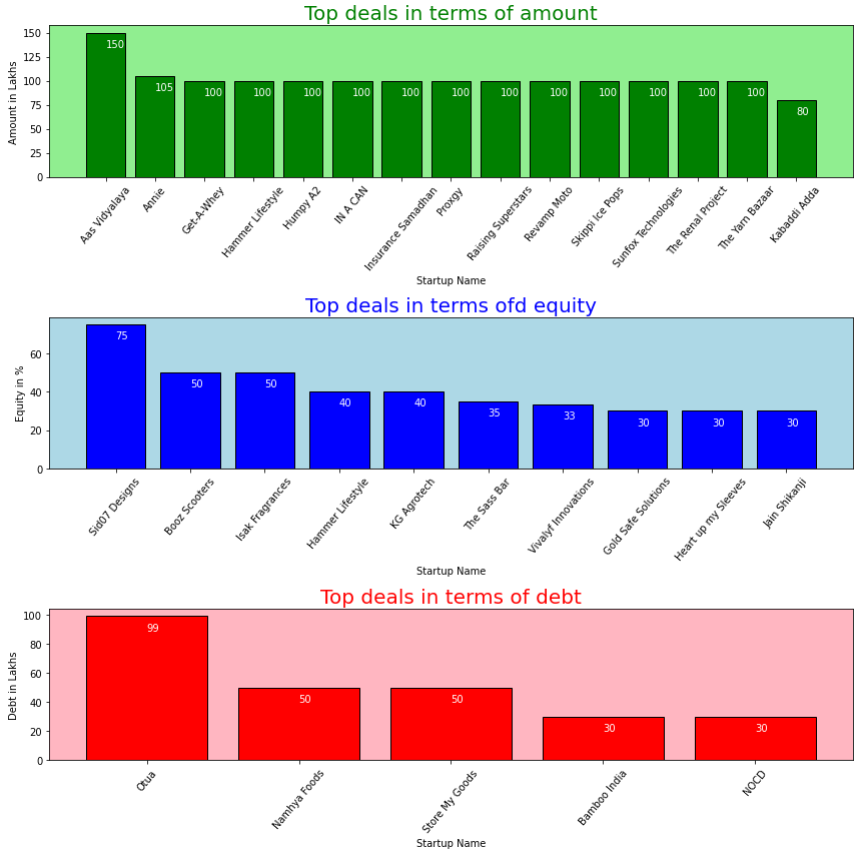
In [4]:
```
print(' Deal in which debt accepted:', '\n', '-'*80, '\n', df[['Startup Name', "Total Deal Amount", "Total Deal Equity", "Total Deal Debt"]].loc[df["Total Deal Debt"].
```

```
 Deal in which debt accepted:
 --------------------------------------------------------------------------------
         Startup Name  Total Deal Amount  Total Deal Equity  Total Deal Debt
8               NOCD               20.0               15.0             30.0
43               PNT               25.0               25.0             25.0
45      Bamboo India               50.0                3.5             30.0
55              Otua                1.0                1.0             99.0
61   The State Plate               40.0                3.0             25.0
65     Sid07 Designs               25.0               75.0             22.0
71      Namhya Foods               50.0               10.0             50.0
76       KG Agrotech               10.0               40.0             20.0
119   Store My Goods               50.0                4.0             50.0
 --------------------------------------------------------------------------------
```

Top deals by Stratups

In [42]:

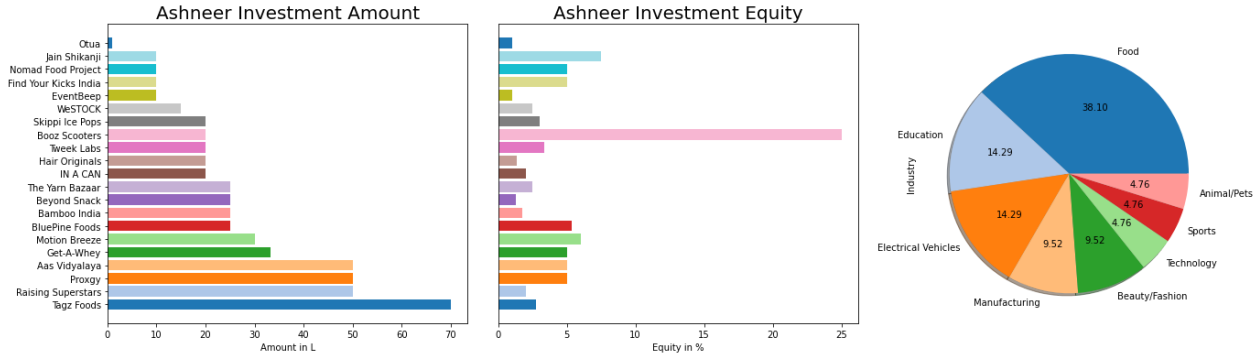
```
Ammount=df.groupby("Startup Name")["Total Deal Amount"].max().nlargest(15)
Equity=df.groupby("Startup Name")["Total Deal Equity"].max().nlargest(10)
Debt=df.groupby("Startup Name")["Total Deal Debt"].max().nlargest(5)
short_data=[Ammount,Equity,Debt]

plots=[1,2,3]
face=['lightgreen', 'lightblue', 'lightpink']
barcolor=['g','b', 'r']
Title=["Top deals in terms of amount", 'Top deals in terms ofd equity', 'Top deals in terms of debt']
ylabel=["Amount in Lakhs", "Equity in %", "Debt in Lakhs"]

plt.figure(figsize=(12,12))
for i,j,k,l,m,n in zip(short_data,plots,face,barcolor,Title,ylabel):
    plt.subplot(3,1,j).set_facecolor(k)
    plt.bar(i.index, i, color=l,ec='k')
    plt.xticks(rotation = 50)
    plt.xlabel("Startup Name", color="k")
    plt.ylabel(n, color="k")
    plt.title(m, color=l, fontsize=20)
    for x, y in enumerate(i):
        plt.annotate(round(y), (x,y-max(i)*0.1), color='w')

plt.tight_layout() #The tight_layout() function in pyplot module of matplotlib library is used to automatically adjust subplot parameters to give specified padding
```

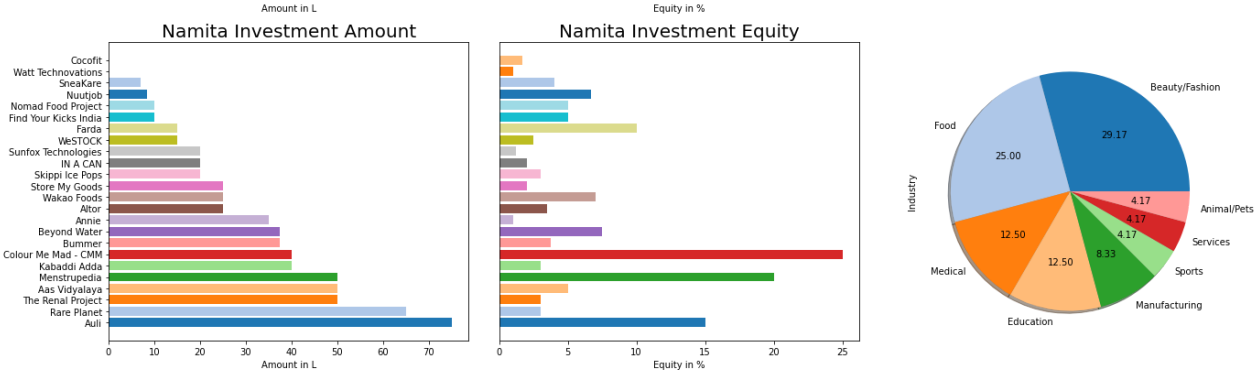


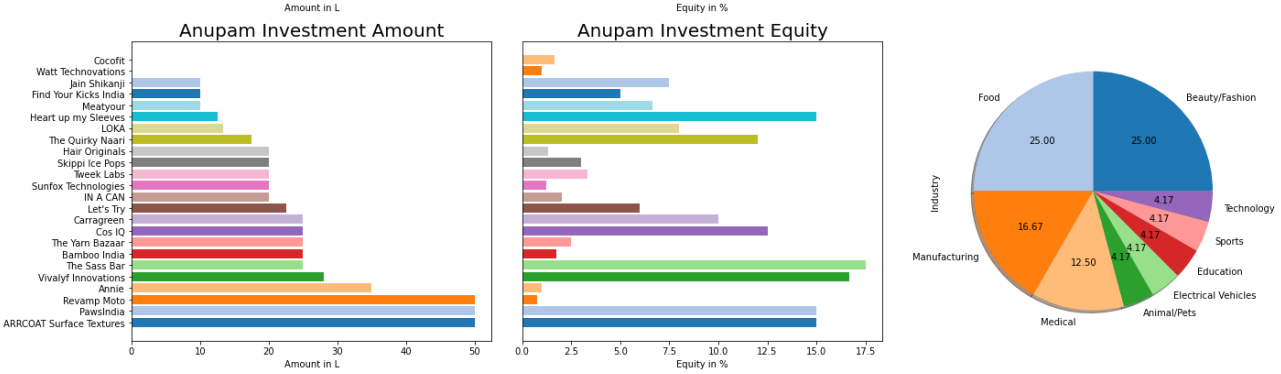
Overview of Sharks Investment in terms of Amount, Equity and Debt

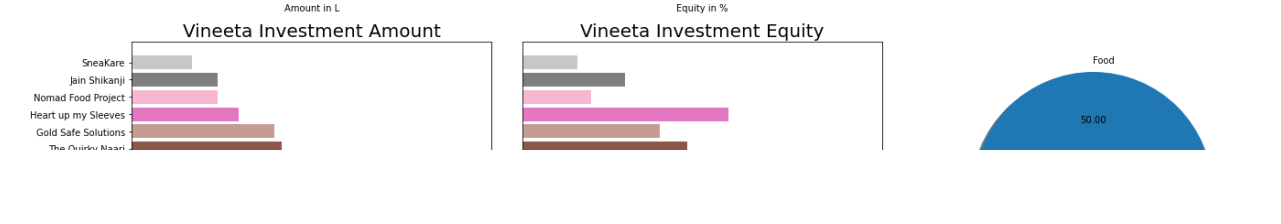
In [43]:

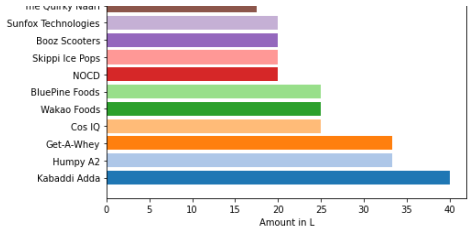
```
Name = ['Ashneer_amount', 'Namita_amount', 'Anupam_amount', 'Vineeta_amount', 'Aman_amount', 'Peyush_amount', 'Ghazal_amount']
amount = ['Ashneer Investment Amount', 'Namita Investment Amount', 'Anupam Investment Amount', 'Vineeta Investment Amount', 'Aman Investment Amount', 'Peyush Investmer
for i,j in zip(Name, amount):
    i = df.sort_values(j, ascending = False).loc[(df[j].isnull()==False)& (df[j]>0)]

# Sorting Sharks Investmetns
Ashneer_amount = df.sort_values("Ashneer Investment Amount",ascending=False).loc[(df['Ashneer Investment Amount'].isnull()==False) & (df["Ashneer Investment Amount"] >
Namita_amount = df.sort_values("Namita Investment Amount",ascending=False).loc[(df['Namita Investment Amount'].isnull()==False) & (df["Namita Investment Amount"] > 0)]
Anupam_amount = df.sort_values("Anupam Investment Amount",ascending=False).loc[(df['Anupam Investment Amount'].isnull()==False) & (df["Anupam Investment Amount"] > 0)]
Vineeta_amount = df.sort_values("Vineeta Investment Amount",ascending=False).loc[(df['Vineeta Investment Amount'].isnull()==False) & (df["Vineeta Investment Amount"] >
Aman_amount = df.sort_values("Aman Investment Amount",ascending=False).loc[(df['Aman Investment Amount'].isnull()==False) & (df["Aman Investment Amount"] > 0)]
Peyush_amount = df.sort_values("Peyush Investment Amount",ascending=False).loc[(df['Peyush Investment Amount'].isnull()==False) & (df["Peyush Investment Amount"] > 0)]
Ghazal_amount = df.sort_values("Ghazal Investment Amount",ascending=False).loc[(df['Ghazal Investment Amount'].isnull()==False) & (df["Ghazal Investment Amount"] > 0)]

Ashneer = Ashneer_amount
Namita = Namita_amount
Anupam = Anupam_amount
Vineeta = Vineeta_amount
Aman = Aman_amount
Peyush = Peyush_amount
Ghazal = Ghazal_amount
plots=0
sorted_Data=[Ashneer,Namita,Anupam,Vineeta,Aman,Peyush,Ghazal]
col_name1=["Ashneer Investment Amount","Namita Investment Amount","Anupam Investment Amount","Vineeta Investment Amount","Aman Investment Amount","Peyush Investment Am
col_name2=["Ashneer Investment Equity","Namita Investment Equity","Anupam Investment Equity","Vineeta Investment Equity","Aman Investment Equity","Peyush Investment Eq
colors = plt.cm.tab20.colors

#Plotting Charts of Equity, Investmetn Amount and Debt

plt.figure(figsize=(20,38))
for b,c,d in zip(sorted_Data,col_name1,col_name2):
    plt.subplot(7,3,plots+1).set_facecolor("w")
    plots=plots+1
    plt.barh(b["Startup Name"],b[c],color=colors)
    plt.grid(False)
    plt.title(c,fontsize=20,color="k")
    plt.xlabel("Amount in L")

    plt.subplot(7,3,plots+1).set_facecolor("w")
    plots=plots+1
    plt.barh(b["Startup Name"],b[d],color=colors)
    plt.grid(False)
    plt.title(d,fontsize=20,color="k")
    plt.xlabel("Equity in %")
    plt.yticks([])

    plt.subplot(7,3,plots+1)
    plots=plots+1
    b["Industry"].value_counts().plot(kind='pie',autopct='%.2f',colors=colors,shadow=True)
plt.tight_layout()
```

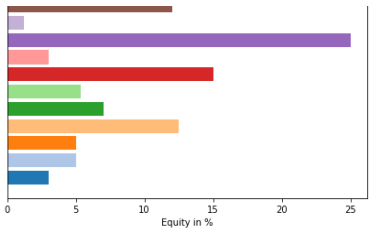


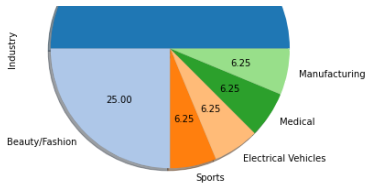


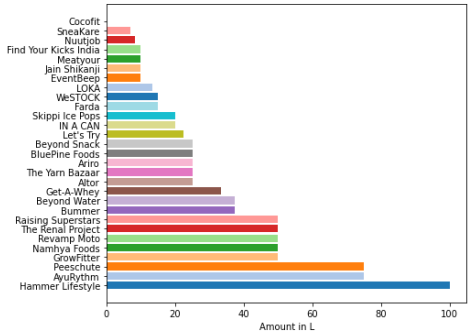


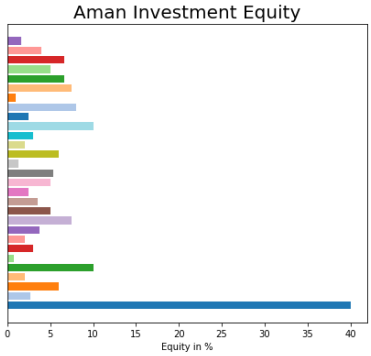


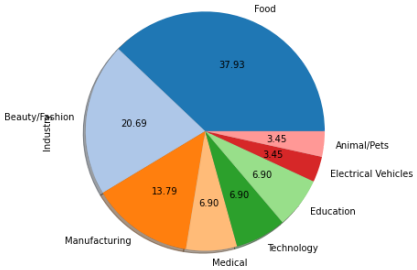


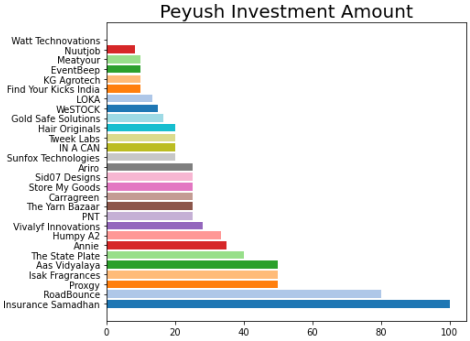


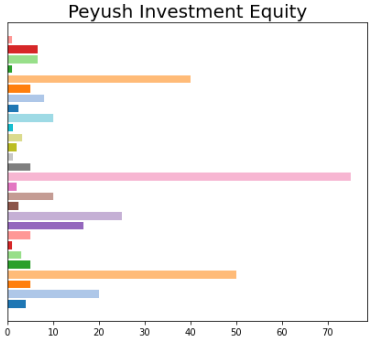


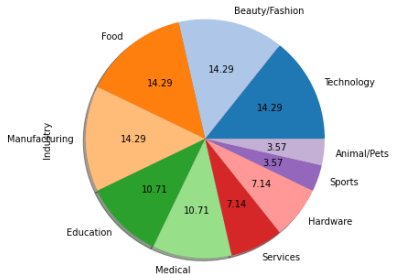


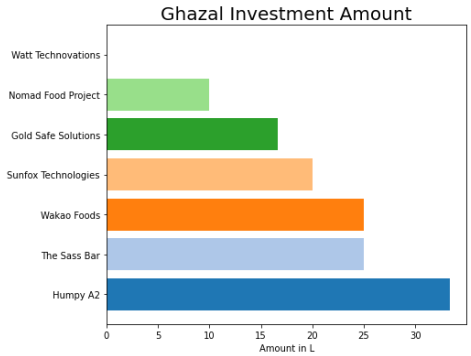


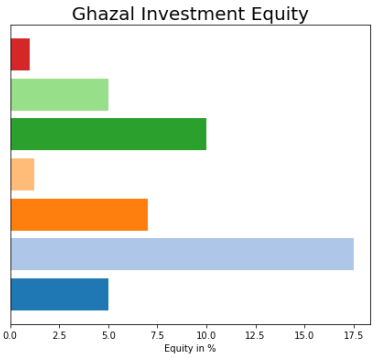


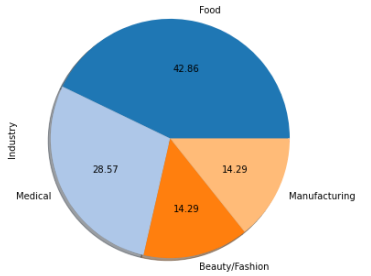












In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```