Anakha R Menon

CH.EN.U4CSE20103

CSE-B

Company

In [1]:
```
#importing required Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as mp
import seaborn as sns
```

Startup ecosystem (CompanyX_EU.csv)

Loding the Dataset

In [2]:
```
df = pd.read_csv("DS+-+Part3+-+CompanyX_EU.csv")
```

Analysing the Dataset

In [3]:
```
# cheaking for info of data set
df.info
```

Out[3]:
```
<bound method DataFrame.info of              Startup            Product Funding
Event  \
0             2600Hz          2600hz.com     NaN   Disrupt SF 2013
1               3DLT            3dlt.com   $630K  Disrupt NYC 2013
2         3DPrinterOS     3dprinteros.com     NaN   Disrupt SF 2016
3          3Dprintler      3dprintler.com     $1M   Disrupt NY 2016
4    42 Technologies  42technologies.com     NaN  Disrupt NYC 2013
..               ...                 ...     ...               ...
657           Zivity          zivity.com     $8M         TC40 2007
658           Zmorph        zmorph3d.com     $1M                -
659           Zocdoc          zocdoc.com   $223M         TC40 2007
660             Zula         zulaapp.com   $3.4M   Disrupt SF 2013
661           Zumper          zumper.com  $31.5M   Disrupt SF 2012

              Result OperatingState
0         Contestant      Operating
1         Contestant         Closed
2         Contestant      Operating
3    Audience choice      Operating
4         Contestant      Operating
..               ...            ...
657       Contestant      Operating
658  Audience choice      Operating
659       Contestant      Operating
660  Audience choice      Operating
661         Finalist      Operating

[662 rows x 6 columns]>
```

In [4]:
```
# printing first five rows of data
df.head()
```

Out[4]:

	Startup	Product	Funding	Event	Result	OperatingState
0	2600Hz	2600hz.com	NaN	Disrupt SF 2013	Contestant	Operating
1	3DLT	3dlt.com	$630K	Disrupt NYC 2013	Contestant	Closed
2	3DPrinterOS	3dprinteros.com	NaN	Disrupt SF 2016	Contestant	Operating
3	3Dprintler	3dprintler.com	$1M	Disrupt NY 2016	Audience choice	Operating
4	42 Technologies	42technologies.com	NaN	Disrupt NYC 2013	Contestant	Operating

In [5]:
```
# Checking for what data types are avilable
df.dtypes
```

Out[5]:
```
Startup           object
Product           object
Funding           object
Event             object
Result            object
OperatingState    object
dtype: object
```

In [6]:
```
#describing the Data
df.describe().T
```

Out[6]:

	count	unique	top	freq
Startup	662	662	2600Hz	1
Product	656	656	2600hz.com	1
Funding	448	240	$1M	17
Event	662	26	TC50 2008	52
Result	662	5	Contestant	488
OperatingState	662	4	Operating	465

In [7]:
```
#Total Data length
df.shape
```

Out[7]:
```
(662, 6)
```

Cleaning the DataSet

In [8]:
```
# Checking for null values
df.isnull().sum()
```

Out[8]:
```
Startup             0
Product             6
Funding           214
Event               0
Result              0
OperatingState      0
dtype: int64
```

In [9]:
```
df.isna().sum()
```

Out[9]:
```
Startup             0
Product             6
Funding           214
Event               0
Result              0
OperatingState      0
dtype: int64
```

In [10]:
```
#Percentage of null
(df.isna().sum()/len(df) ) *100
```

Out[10]:
```
Startup            0.000000
Product            0.906344
Funding           32.326284
Event              0.000000
Result             0.000000
OperatingState     0.000000
dtype: float64
```

In [11]:
```
# Repalcing the Null values of Funding with $oK
df["Funding"] = df["Funding"].fillna("$0K")
```

In [12]:
```
#Converting all Funds in to Millions
df.loc[:,'Funds_in_million'] = df['Funding'].apply(lambda x: float(x[1:-1])/1000 i
```

In [13]:
```
#Checking again for null
df.isnull().sum()
```

Out[13]:
```
Startup             0
Product             6
Funding             0
Event               0
Result              0
OperatingState      0
Funds_in_million    0
dtype: int64
```

In [14]:
```
#Droping all rows of null values in Product column
df.dropna(subset=["Product"],inplace=True)
```

In [15]:
```
df.describe(include="all").T
```

Out[15]:

	count	unique	top	freq	mean	std	min	25%	50%	75%
Startup	656	656	2600Hz	1	NaN	NaN	NaN	NaN	NaN	NaN
Product	656	656	2600hz.com	1	NaN	NaN	NaN	NaN	NaN	NaN
Funding	656	240	$0K	210	NaN	NaN	NaN	NaN	NaN	NaN
Event	656	26	TC50 2008	52	NaN	NaN	NaN	NaN	NaN	NaN
Result	656	5	Contestant	482	NaN	NaN	NaN	NaN	NaN	NaN
OperatingState	656	4	Operating	460	NaN	NaN	NaN	NaN	NaN	NaN
Funds_in_million	656.0	NaN	NaN	NaN	11.72211	75.014418	0.0	0.0	0.7778	4.42

In [16]:
```
df.isnull().sum()
```

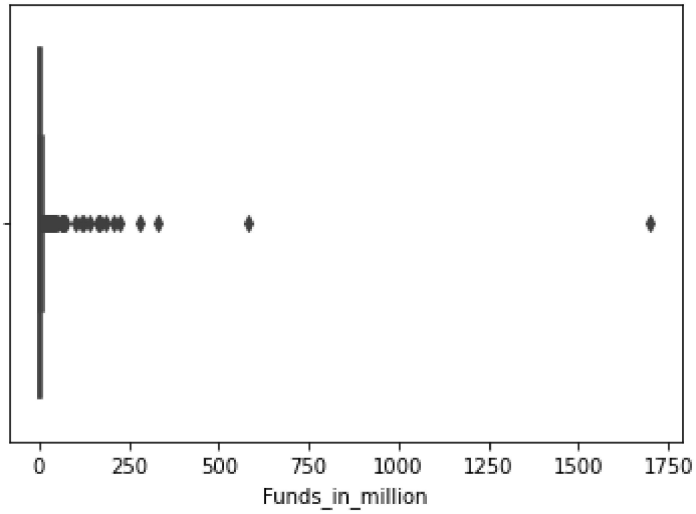
Out[16]:
```
Startup             0
Product             0
Funding             0
Event               0
Result              0
OperatingState      0
Funds_in_million    0
dtype: int64
```

Hence, Data is Cleaned we have removed null values in products and replaced null values of Funding as "$0K"

Visuvalizing the Data which is cleaned

In [17]:
```
# Plot box plot for funds in million.
sns.boxplot(x=df["Funds_in_million"])
```

Out[17]:
```
<AxesSubplot:xlabel='Funds_in_million'>
```



In [18]:
```
q3  = df["Funds_in_million"].quantile(0.75)
q1  = df["Funds_in_million"].quantile(0.25)
iqr = q3-q1
upper_fence = q3 + (1.5 * iqr)
lower_fence = q1 - (1.5 * iqr)
```

In [19]:
```
# Check the number of outliers greater than the upper fence.
len(df.loc[df["Funds_in_million"] > upper_fence])
```

Out[19]:
```
98
```

In [20]:
```
#Check frequency of the OperatingState features classes.
df["OperatingState"].value_counts()
```

Out[20]:
```
Operating    460
Closed       105
Acquired      86
Ipo            5
Name: OperatingState, dtype: int64
```

Statistical Analysis

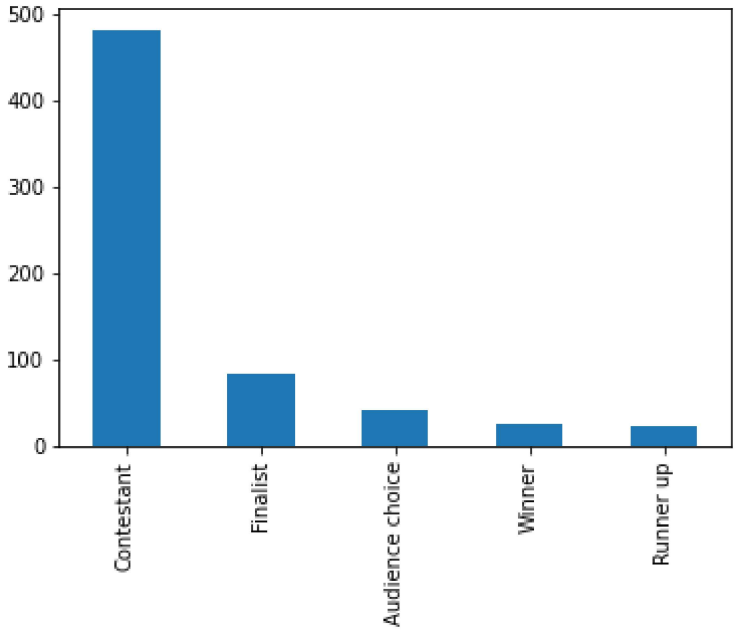
In [21]:
```
# Make a copy of the original data frame.
df1 = df.copy()
```

In [22]:
```
#Check frequency of the OperatingState Results classes.
df["Result"].value_counts()
```

Out[22]:
```
Contestant         482
Finalist            84
Audience choice     41
Winner              26
Runner up           23
Name: Result, dtype: int64
```

In [23]:
```
df["Result"].value_counts().plot(kind="bar")
```

Out[23]:
```
<AxesSubplot:>
```



In [24]:
```
df["Result"].value_counts()["Winner"]
```

Out[24]:
```
26
```

Calculate percentage of winners that are still operating and percentage of contestants that are still operating

In [25]:
```
len(df1.loc[(df1["Result"]=="Winner") & (df1["OperatingState"]=="Operating")])/len
```

Out[25]:
```
73.07692307692307
```

In [26]:
```
len(df1.loc[(df1["Result"]=="Contestant") & (df1["OperatingState"]=="Operating")])/
```

Out[26]:
```
67.84232365145229
```

In [27]:
```
df2 = df.loc[(df["Event"] != "-")]
df2.loc[(df2["Event"].str.contains("Disrupt")) & (df2["Event"].str.slice(-4).astype
```

Out[27]:

	Startup	Product	Funding	Event	Result	OperatingState	Funds_in_mill
0	2600Hz	2600hz.com	$0K	Disrupt SF 2013	Contestant	Operating	0
1	3DLT	3dlt.com	$630K	Disrupt NYC 2013	Contestant	Closed	0
2	3DPrinterOS	3dprinteros.com	$0K	Disrupt SF 2016	Contestant	Operating	0
3	3Dprintler	3dprintler.com	$1M	Disrupt NY 2016	Audience choice	Operating	1
4	42 Technologies	42technologies.com	$0K	Disrupt NYC 2013	Contestant	Operating	0
...	...	...	...	...	...	...	
646	YayPay Inc	yaypay.com	$900K	Disrupt London 2015	Contestant	Operating	0
648	YOOBIC	yoobic.com	$0K	Disrupt London 2015	Finalist	Operating	0
653	ZAP!	zapreklam.com/	$0K	Disrupt EU 2014	Audience choice	Operating	0
656	Zenefits	zenefits.com	$583.6M	Disrupt NYC 2013	Finalist	Operating	583
660	Zula	zulaapp.com	$3.4M	Disrupt SF 2013	Audience choice	Operating	3

275 rows × 7 columns