

HOPE ARTIFICIAL INTELLIGENCE

INTERNSHIP

DATA SCIENCE PROJECT

DONE BY: M.GAYATHIRI ANARMEE

III YEAR IT

SRI SAI RAM INSTITUTE OF TECHNOLOGY

GUIDED BY: RAMISHA RANI K

FOUNDER: HOPE AI

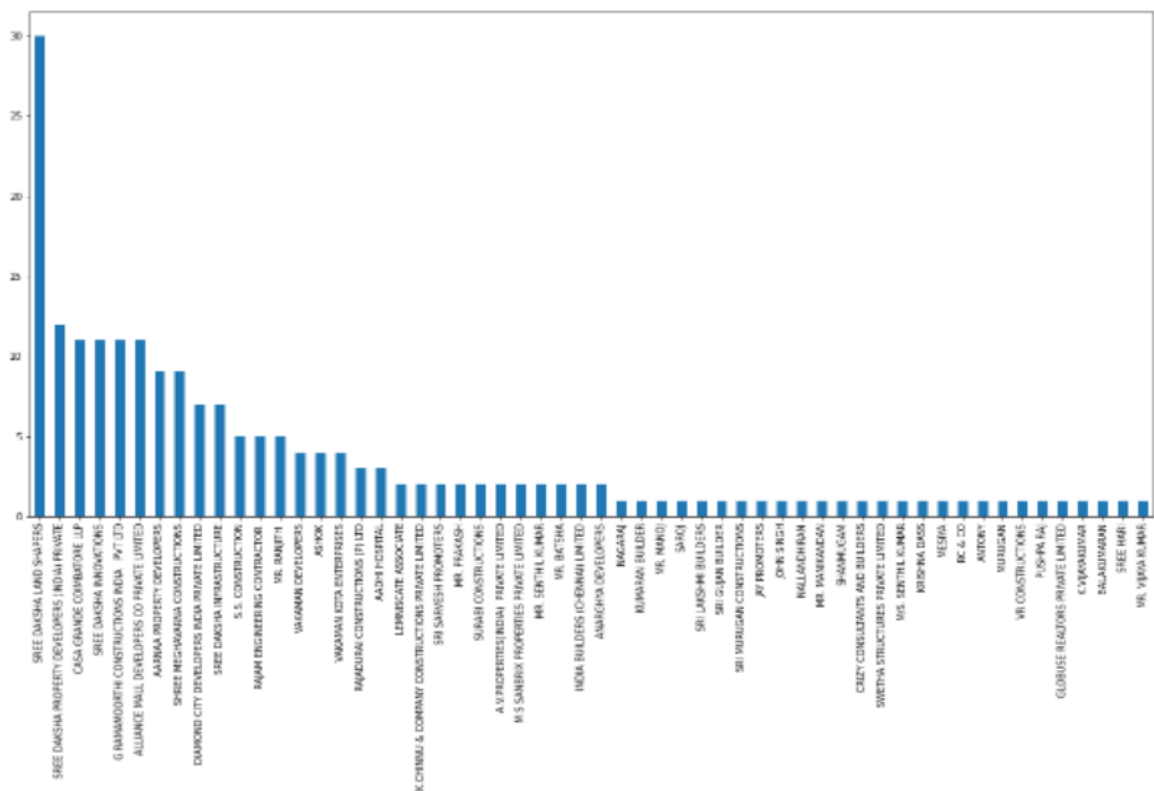
Libraries such as pandas, pyplot and collections are imported to work on datasets, plotting graphs and to work with containers respectively.

1. Create a graph which displays Customer order count. [Ex. customerA-12, customer-5]. Display this information in a graph.

The CSV file is read as a dataframe i.e., data in the form of rows and columns. The column name could be accessed using the dataframe.

value_counts() is used to get the count of elements in the desired column and graph is plotted accordingly.

```
df['Customer Name'].value_counts().plot(kind='bar',figsize=(20,10))
```

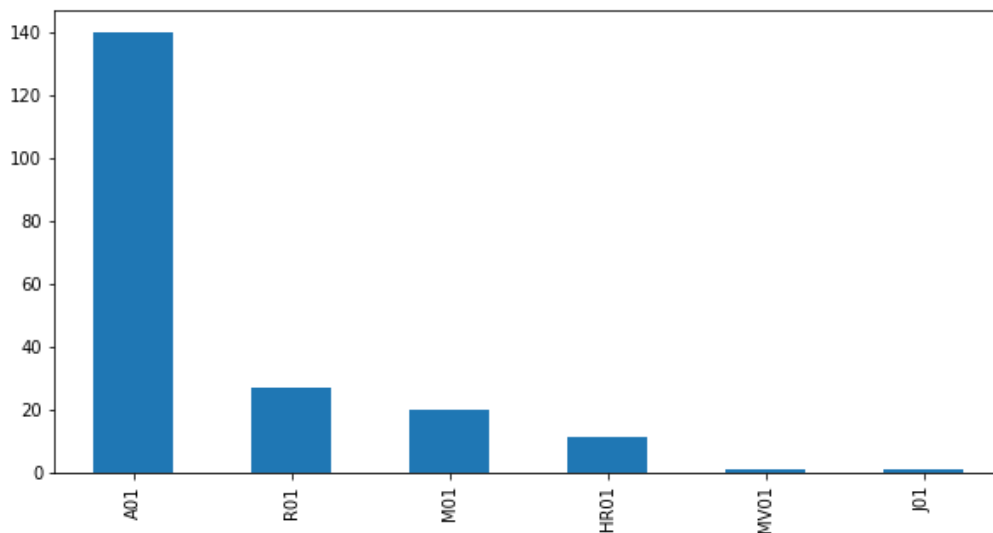


2. Create a graph which displays Marketing Staff sales count.

[Ex.A01=5,M12=15]. Display this information in a graph.

Count of marketing staff sales is also done using `value_counts()`. The count of unique values present in the column are obtained and the graph could be plotted accordingly.

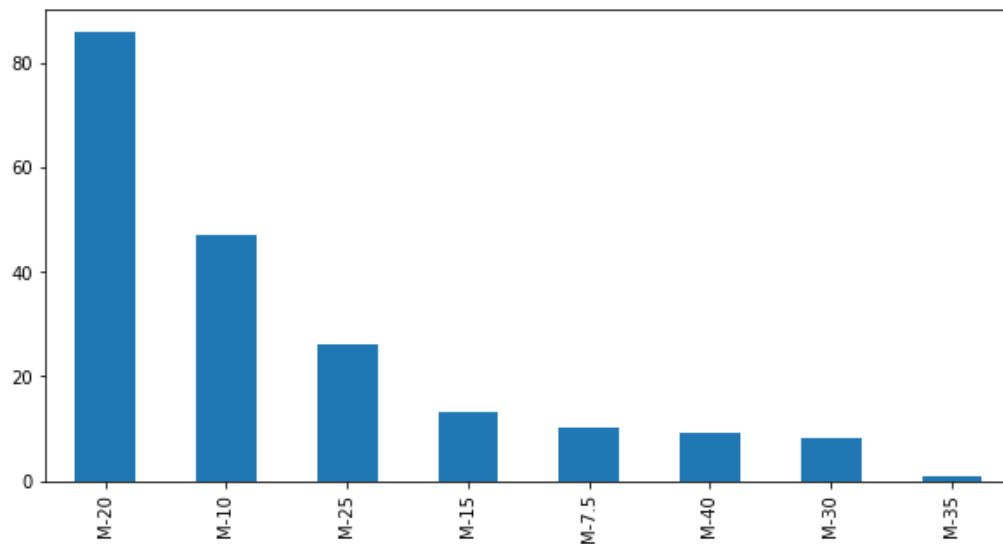
```
df['Marketing Staff'].value_counts().plot(kind='bar',figsize=(10,5))
```



3. Display the graph which displays Gradewise sales count[Ex:M10=90]. Display this information in a graph.

Count of grade wise sales is displayed in a similar fashion using `value_counts()`. Once we have obtained the count of unique values from the desired column the graph is plotted using `plot()` where we can specify the kind of graph required(i.e., bar, pie, etc.) and the figure size.

```
df['Grade Of Concrete'].value_counts().plot(kind='bar',figsize=(10,5))
```



4. Display the graph which shows the total quantity of each grades.[Ex.M10=30]

Two lists are used. One is used to store the unique grade values and the other one is used to keep track of the quantity of each grade. Index value of the grade corresponds to that present in the list storing its quantity. Once the lists are consolidated they are plotted using `plt.plot()` where the list names are passed as arguments.

```
d=[]  
for i in range(len(df)):  
    if(df['Grade Of Concrete'][i] not in d):  
        d.append(df['Grade Of Concrete'][i])  
print(d)
```

```
['M-25', 'M-10', 'M-20', 'M-7.5', 'M-30', 'M-40', 'M-15', 'M-35', nan]
```

#quantity for each grade is calculated and added to a list q[] which corresponds

#to the index values of grade of concrete in list d[]

q=[]

for j in range(len(d)):

a=0

for i in range(len(df)):

if(d[j]==df['Grade Of Concrete'][i]):

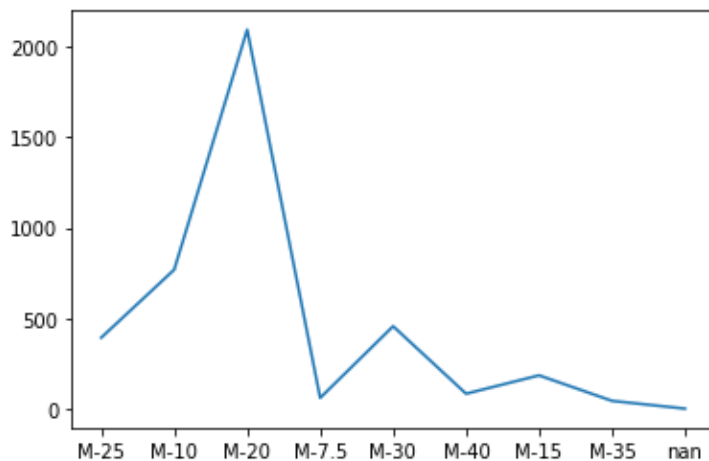
a+=df['QTY'][i]

q.append(a)

print(q)

[392.5, 768.5, 2095.5, 59.5, 455.5, 82.5, 184.0, 43.0, 0]

plt.plot(d,q)



Descriptive Analysis:

It is a type of analysis of data that helps describe, show or summarize data points in constructive way such that patterns might emerge that fulfill every condition of the data.

```
df[['PRICE PER CUM','QTY']].mean()
```

```
PRICE PER CUM    3895.800995
QTY              20.405000
dtype: float64
```

```
df[['PRICE PER CUM','QTY']].median()
```

```
PRICE PER CUM    4100.0
QTY              8.0
dtype: float64
```

```
df[['Customer Name','PRICE PER CUM','QTY','Grade Of Concrete','Marketing Staff']].describe(include='all')
```

Out[]:	Customer Name	PRICE PER CUM	QTY	Grade Of Concrete	Marketing Staff
count	200	201.000000	200.000000	200	200
unique	56	NaN	NaN	8	6
top	SREE DAKSHA LAND SHAPERS	NaN	NaN	M-20	A01
freq	30	NaN	NaN	86	140
mean	NaN	3895.800995	20.405000	NaN	NaN
std	NaN	824.907068	32.283854	NaN	NaN
min	NaN	33.000000	1.500000	NaN	NaN
25%	NaN	3250.000000	5.000000	NaN	NaN
50%	NaN	4100.000000	8.000000	NaN	NaN
75%	NaN	4300.000000	18.625000	NaN	NaN
max	NaN	5705.000000	227.000000	NaN	NaN

Github Project Link: <https://github.com/Gayathiri10/Hope-learning-Data-Science>