Bestfer Solution MVP

Features

1. Total Units Sold x Months Graph
2. Total Sales x Months Graph

Importing Libraries

In [16]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

The data at a glance

Read .csv file from folder (BFH-LEDE)

In [17]:
```
data = pd.read_csv('/Users/rafaelcyin/Documents/GitHub/Bestfer_Solution_JN/data
```

Display .csv file as table

In [18]:
```
data.head()
```

Out[18]:

	Unnamed: 0	Codigo	Descrição - 31/07/2022	UM	Import?	Feb-22	Mar-22	Apr-22	May-22	Jun-22	Jul-22
0	1.0	'BFH0003	BF PISTOLA DE COLA 6W	PC	Sim	1781.0	2608.0	2069	2226	1362	1889
1	2.0	'BFH0014	BF JG CHAVE HEX GRANDE C/PONTA BOLA 9PCS ACZ	PC	Sim	121.0	217.0	177	100	115	49
2	3.0	'BFH0015	BF JG CH COM BITS 28PCS	PC	Sim	97.0	258.0	150	299	81	199
3	4.0	'BFH0016	BF JG CHAVE TORX GRANDE 9PCS AC	PC	Sim	129.0	268.0	138	226	240	153
4	5.0	'BFH0020	BF LAMINA P/ARCO DE SERRA 12PCS 24T	PC	Sim	2182.0	3569.0	4371	2663	2037	2623

In [19]:
```
data.describe().head()
```

Out[19]:

	Unnamed: 0	Feb-22	Mar-22	Apr-22	May-22	Jun-22
count	1692.000000	1693.000000	1693.000000	1693.000000	1693.000000	1693.000000
mean	846.500000	925.291081	1320.194826	1232.954519	1092.533963	991.559362
std	488.582644	19066.672724	27190.300530	25407.670243	22549.255084	20489.706057
min	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	423.750000	27.000000	40.000000	31.000000	28.000000	28.000000

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1693 entries, 0 to 1692
Data columns (total 20 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   Unnamed: 0                            1692 non-null   float64
 1   Codigo                                1692 non-null   object 
 2   Descrição       -       31/07/2022    1692 non-null   object 
 3   UM                                    1692 non-null   object 
 4   Import?                               1692 non-null   object 
 5   Feb-22                                1693 non-null   float64
 6   Mar-22                                1693 non-null   float64
 7   Apr-22                                1693 non-null   int64  
 8   May-22                                1693 non-null   int64  
 9   Jun-22                                1693 non-null   int64  
 10  Jul-22                                1693 non-null   int64  
 11  Total                                 1693 non-null   float64
 12  Saldo                                 1692 non-null   float64
 13  Freq.                                 1692 non-null   float64
 14  Media                                 1692 non-null   float64
 15  Tempo                                 1692 non-null   float64
 16  FOB                                   1692 non-null   float64
 17  Venda                                 1692 non-null   float64
 18  Ativo                                 1692 non-null   object 
 19  Entrada Prevista                      1692 non-null   float64
dtypes: float64(11), int64(4), object(5)
memory usage: 264.7+ KB
```

In [21]:
```
data.isna().sum()
```

```
Out[21]: Unnamed: 0                            1
         Codigo                                1
         Descrição       -       31/07/2022    1
         UM                                    1
         Import?                               1
         Feb-22                                0
         Mar-22                                0
         Apr-22                                0
         May-22                                0
         Jun-22                                0
         Jul-22                                0
         Total                                 0
         Saldo                                 1
         Freq.                                 1
         Media                                 1
         Tempo                                 1
         FOB                                   1
         Venda                                 1
         Ativo                                 1
         Entrada Prevista                      1
         dtype: int64
```

```
In [22]: data.size
```

```
Out[22]: 33860
```

Dimensions of data

```
In [23]: data.shape
```

```
Out[23]: (1693, 20)
```

Handling Missing Data

Remove `null` values from data

```
In [24]: data = data.dropna()
```

Remove `null` values

```
In [25]: data.isna().sum()
```

Out[25]:

```
Unnamed: 0                                0
Codigo                                    0
Descrição       -       31/07/2022        0
UM                                        0
Import?                                   0
Feb-22                                    0
Mar-22                                    0
Apr-22                                    0
May-22                                    0
Jun-22                                    0
Jul-22                                    0
Total                                     0
Saldo                                     0
Freq.                                     0
Media                                     0
Tempo                                     0
FOB                                       0
Venda                                     0
Ativo                                     0
Entrada Prevista                          0
dtype: int64
```

All `null` values have been removed

Cleaning Data

Check if there are any `inactive` items

In [26]:

```
data['Ativo'].unique()
```

Out[26]:

```
array(['S', 'N'], dtype=object)
```

Remove `inactive` items (Ativo = N)

In [27]:

```
data = data[data['Ativo'].str.contains('N') == False]
```

In [28]:

```
data['Ativo'].unique()
```

Out[28]:

```
array(['S'], dtype=object)
```

Successfully removed the `inactive` items from the data

Data Wrangling

Total Units Sold x Months

Store column name into a `list`

In [29]:

```
column_name = list(data.columns)
```

Create `dictionary` to store total sum of sales and their respective months

In [30]:

```
total_units = [data[column_name[5]].sum(),data[column_name[6]].sum(),data[colun
months = [column_name[5],column_name[6],column_name[7],column_name[8],column_na
index = ['Total Units']
units_data = {'Total Units Sold' : total_units, 'Months' : months}
```

Create graph to plot `total_units` against `months`

In [31]:

```
units_df = pd.DataFrame(units_data)
units_df.head()
```

Out[31]:

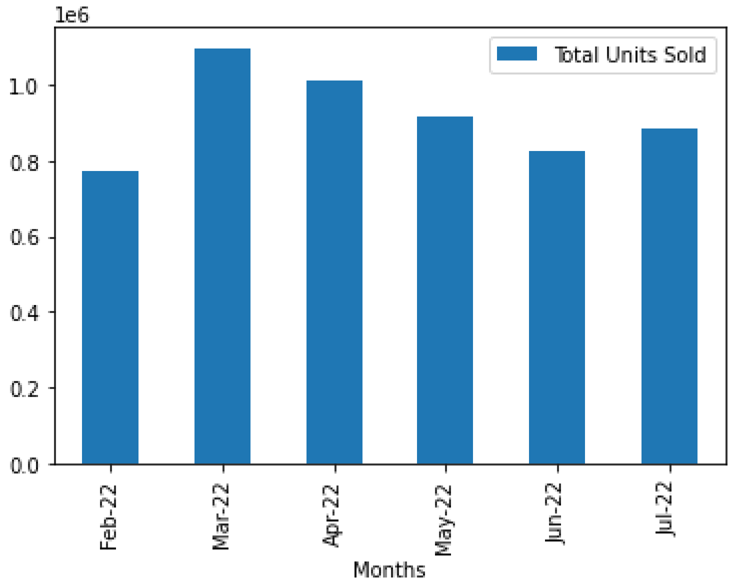
	Total Units Sold	Months
0	774161.90	Feb-22
1	1096649.92	Mar-22
2	1012810.00	Apr-22
3	915310.00	May-22
4	827942.00	Jun-22

In [32]:

```
units_df.plot(x = 'Months', y = 'Total Units Sold', kind='bar')
```

Out[32]:

```
<AxesSubplot:xlabel='Months'>
```



Total Sales x Months

Calculate `mean` price of all proudcts

In [33]:

```
avg_price = data['Venda'].mean()
avg_price
```

Out[33]:

```
26.49056
```

Multiply the `avg_price` by the entire column `Total Units Sold`

In [34]:

```
sales_df = units_df
```

In [35]:

```
sales_df["Total Units Sold"] = avg_price * sales_df["Total Units Sold"]
```

Create graph to plot `total_sales` against `months`

In [36]:

```
sales_df.head()
```

Out[36]:

	Total Units Sold	Months
0	2.050798e+07	Feb-22
1	2.905087e+07	Mar-22
2	2.682990e+07	Apr-22
3	2.424707e+07	May-22
4	2.193265e+07	Jun-22

In [37]:

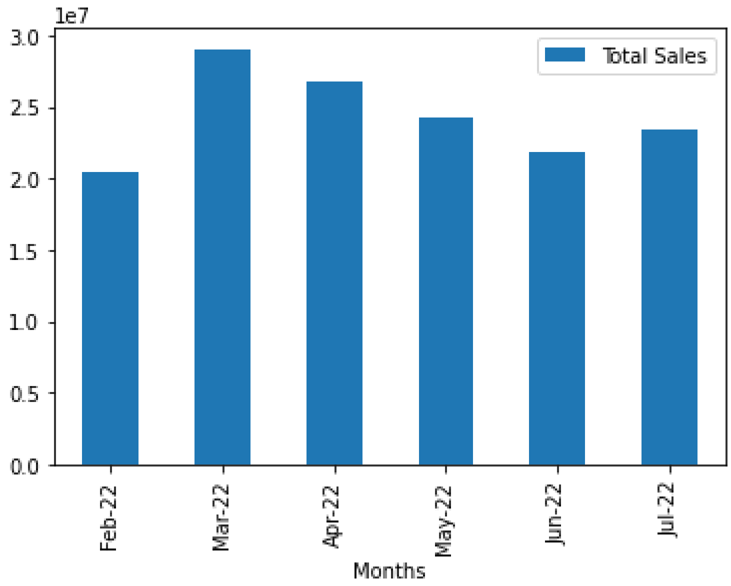
```
sales_df = sales_df.rename(columns = {'Total Units Sold' : 'Total Sales'})
```

In [38]:

```
sales_df.plot(x = 'Months', y = 'Total Sales', kind='bar')
```

Out[38]:

```
<AxesSubplot:xlabel='Months'>
```



In [ ]: