

PRODUCT SALE ANALYSIS

DATA ANALYTICS WITH

COGNOS:GROUP2 PHASE:3

This phase involves in designing of the steps that defining in each phase of the previous documentation this involves importing necessary functions, data processing and so on in this phase we have to begin our project by loading and preprocessing the dataset.

The IBM suggests using the jupyter notebook for loading and preprocess the dataset:

Here for this project title we need to define the loading the libraries, understand the data and visualize the missing values.

For this certain inputs are defined for this project.in this phase each of the input lines of the project is given as follows:

IBM NAAN MUDHULVAN PHASE3

```
import pandas as pd
import numpy as np
```

```
df = pd.read_csv('statsfinal.csv')
```

```
df.head
```

```
<bound method NDFrame.head of      Unnamed: 0      Date  Q-P1  Q-P2
Q-P3  Q-P4      S-P1      S-P2  \
0      0      13-06-2010  5422  3725  576  907  17187.74
23616.50
1      1      14-06-2010  7047  779  3578  1574  22338.99
4938.86
2      2      15-06-2010  1572  2082  595  1145  4983.24
13199.88
3      3      16-06-2010  5657  2399  3140  1672  17932.69
15209.66
4      4      17-06-2010  3668  3207  2184  708  11627.56
20332.38
...      ...      ...      ...      ...      ...      ...
..
4595      4595  30-01-2023  2476  3419  525  1359  7848.92
21676.46
4596      4596  31-01-2023  7446  841  4825  1311  23603.82
5331.94
4597      4597  01-02-2023  6289  3143  3588  474  19936.13
19926.62
4598      4598  02-02-2023  3122  1188  5899  517  9896.74
7531.92
4599      4599  03-02-2023  1234  3854  2321  406  3911.78
24434.36
```

```
      S-P3      S-P4
0      3121.92  6466.91
1      19392.76 11222.62
2      3224.90  8163.85
3      17018.80 11921.36
4      11837.28  5048.04
...      ...      ...
4595      2845.50  9689.67
4596      26151.50  9347.43
4597      19446.96  3379.62
4598      31972.58  3686.21
4599      12579.82  2894.78
```

```
[4600 rows x 10 columns]>
```

```
df.shape
```

```
(4600, 10)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 4600 entries, 0 to 4599
```

```
Data columns (total 10 columns):
```

#	Column	Non-Null Count	Dtype
0	Unnamed: 0	4600 non-null	int64
1	Date	4600 non-null	object
2	Q-P1	4600 non-null	int64
3	Q-P2	4600 non-null	int64
4	Q-P3	4600 non-null	int64
5	Q-P4	4600 non-null	int64
6	S-P1	4600 non-null	float64
7	S-P2	4600 non-null	float64
8	S-P3	4600 non-null	float64
9	S-P4	4600 non-null	float64

```
dtypes: float64(4), int64(5), object(1)
```

```
memory usage: 359.5+ KB
```

```
df.columns.values
```

```
array(['Unnamed: 0', 'Date', 'Q-P1', 'Q-P2', 'Q-P3', 'Q-P4', 'S-P1',  
      'S-P2', 'S-P3', 'S-P4'], dtype=object)
```

```
df.dtypes
```

Unnamed: 0	int64
Date	object
Q-P1	int64
Q-P2	int64
Q-P3	int64
Q-P4	int64
S-P1	float64
S-P2	float64
S-P3	float64
S-P4	float64

```
dtype: object
```

```
df = df.drop(['Q-P4'], axis = 1)
```

```
df.head()
```

	Unnamed: 0	Date	Q-P1	Q-P2	Q-P3	S-P1	S-P2
S-P3 \							
0	0	13-06-2010	5422	3725	576	17187.74	23616.50
3121.92							
1	1	14-06-2010	7047	779	3578	22338.99	4938.86
19392.76							
2	2	15-06-2010	1572	2082	595	4983.24	13199.88
3224.90							
3	3	16-06-2010	5657	2399	3140	17932.69	15209.66

```
17018.80
4          4  17-06-2010  3668  3207  2184  11627.56  20332.38
11837.28
```

```
      S-P4
0    6466.91
1   11222.62
2    8163.85
3   11921.36
4    5048.04
```

```
df[np.isnan(df['Q-P3'])]
```

```
Empty DataFrame
```

```
Columns: [Unnamed: 0, Date, Q-P1, Q-P2, Q-P3, S-P1, S-P2, S-P3, S-P4]
Index: []
```

```
df[df['Date'] == 0].index
```

```
Int64Index([], dtype='int64')
```

```
df.isnull().sum()
```

```
Unnamed: 0    0
Date          0
Q-P1          0
Q-P2          0
Q-P3          0
S-P1          0
S-P2          0
S-P3          0
S-P4          0
```

```
dtype: int64
```

```
df.drop(labels=df[df['S-P1'] == 0].index, axis=0, inplace=True)
df[df['S-P1'] == 0].index
```

```
Int64Index([], dtype='int64')
```

```
df.fillna(df["S-P3"].mean())
```

```
      Unnamed: 0      Date  Q-P1  Q-P2  Q-P3      S-P1      S-P2
S-P3 \
0          0  13-06-2010  5422  3725   576  17187.74  23616.50
3121.92
1          1  14-06-2010  7047   779  3578  22338.99   4938.86
19392.76
2          2  15-06-2010  1572  2082   595   4983.24  13199.88
3224.90
3          3  16-06-2010  5657  2399  3140  17932.69  15209.66
17018.80
4          4  17-06-2010  3668  3207  2184  11627.56  20332.38
```

```

11837.28
...
...
...
4595      4595  30-01-2023  2476  3419  525  7848.92  21676.46
2845.50
4596      4596  31-01-2023  7446   841  4825  23603.82  5331.94
26151.50
4597      4597  01-02-2023  6289  3143  3588  19936.13  19926.62
19446.96
4598      4598  02-02-2023  3122  1188  5899   9896.74   7531.92
31972.58
4599      4599  03-02-2023  1234  3854  2321   3911.78  24434.36
12579.82

```

```

      S-P4
0      6466.91
1     11222.62
2      8163.85
3     11921.36
4      5048.04
...
4595    9689.67
4596    9347.43
4597    3379.62
4598    3686.21
4599    2894.78

```

```
[4600 rows x 9 columns]
```

```
df.fillna(df["S-P4"].mean())
```

```

      Unnamed: 0      Date  Q-P1  Q-P2  Q-P3      S-P1      S-P2
S-P3 \
0      0  13-06-2010  5422  3725  576  17187.74  23616.50
3121.92
1      1  14-06-2010  7047   779  3578  22338.99  4938.86
19392.76
2      2  15-06-2010  1572  2082   595   4983.24  13199.88
3224.90
3      3  16-06-2010  5657  2399  3140  17932.69  15209.66
17018.80
4      4  17-06-2010  3668  3207  2184  11627.56  20332.38
11837.28
...
...
...
4595      4595  30-01-2023  2476  3419  525  7848.92  21676.46
2845.50
4596      4596  31-01-2023  7446   841  4825  23603.82  5331.94
26151.50
4597      4597  01-02-2023  6289  3143  3588  19936.13  19926.62

```

```

19446.96
4598      4598  02-02-2023  3122  1188  5899   9896.74   7531.92
31972.58
4599      4599  03-02-2023  1234  3854  2321   3911.78  24434.36
12579.82

```

```

      S-P4
0      6466.91
1     11222.62
2      8163.85
3     11921.36
4      5048.04
...
4595    9689.67
4596    9347.43
4597    3379.62
4598    3686.21
4599    2894.78

```

```
[4600 rows x 9 columns]
```

```
df.fillna(df["S-P2"].mean())
```

```

      Unnamed: 0      Date  Q-P1  Q-P2  Q-P3      S-P1      S-P2
S-P3 \
0      0  13-06-2010  5422  3725   576  17187.74  23616.50
3121.92
1      1  14-06-2010  7047   779  3578  22338.99   4938.86
19392.76
2      2  15-06-2010  1572  2082   595   4983.24  13199.88
3224.90
3      3  16-06-2010  5657  2399  3140  17932.69  15209.66
17018.80
4      4  17-06-2010  3668  3207  2184  11627.56  20332.38
11837.28
...      ...      ...      ...      ...      ...      ...
...
4595    4595  30-01-2023  2476  3419   525   7848.92  21676.46
2845.50
4596    4596  31-01-2023  7446   841  4825  23603.82   5331.94
26151.50
4597    4597  01-02-2023  6289  3143  3588  19936.13  19926.62
19446.96
4598    4598  02-02-2023  3122  1188  5899   9896.74   7531.92
31972.58
4599    4599  03-02-2023  1234  3854  2321   3911.78  24434.36
12579.82

```

```

      S-P4
0      6466.91

```

```
1      11222.62
2       8163.85
3      11921.36
4       5048.04
...      ...
4595    9689.67
4596    9347.43
4597    3379.62
4598    3686.21
4599    2894.78
```

```
[4600 rows x 9 columns]
```

```
df.isnull().sum()
```

```
Unnamed: 0      0
Date            0
Q-P1            0
Q-P2            0
Q-P3            0
S-P1            0
S-P2            0
S-P3            0
S-P4            0
dtype: int64
```