CSE 4020 - MACHINE LEARNING

Lab 29+30

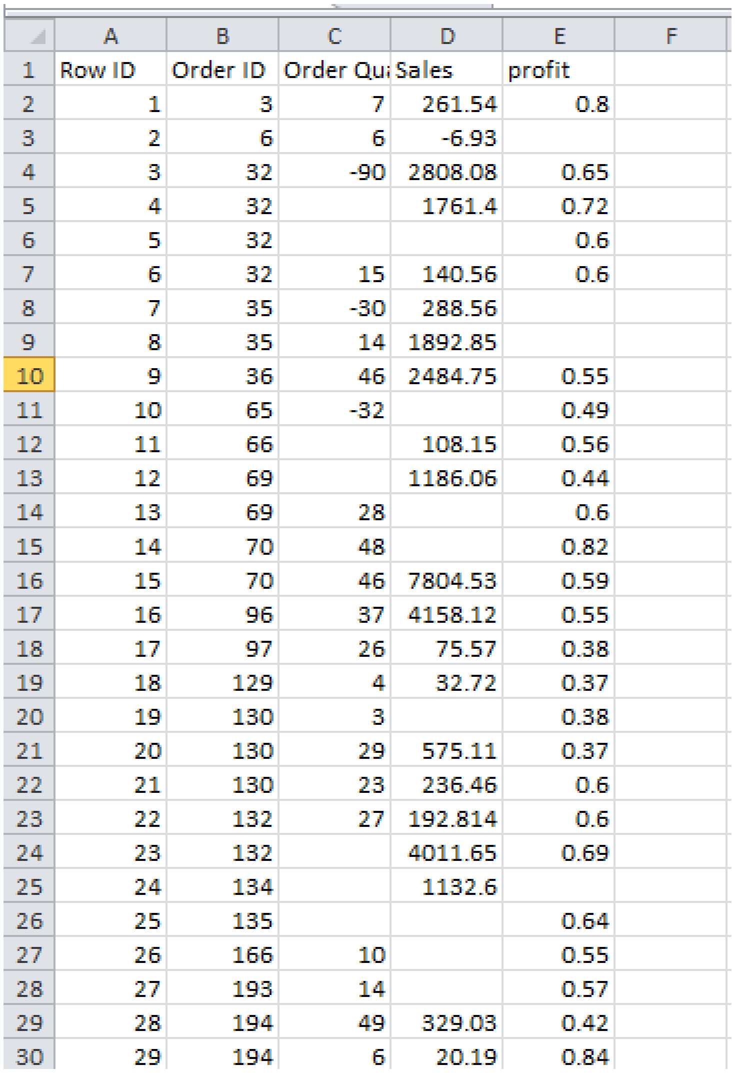
Lab Task1

Submitted by: Alokam Nikhitha(19BCE2555)

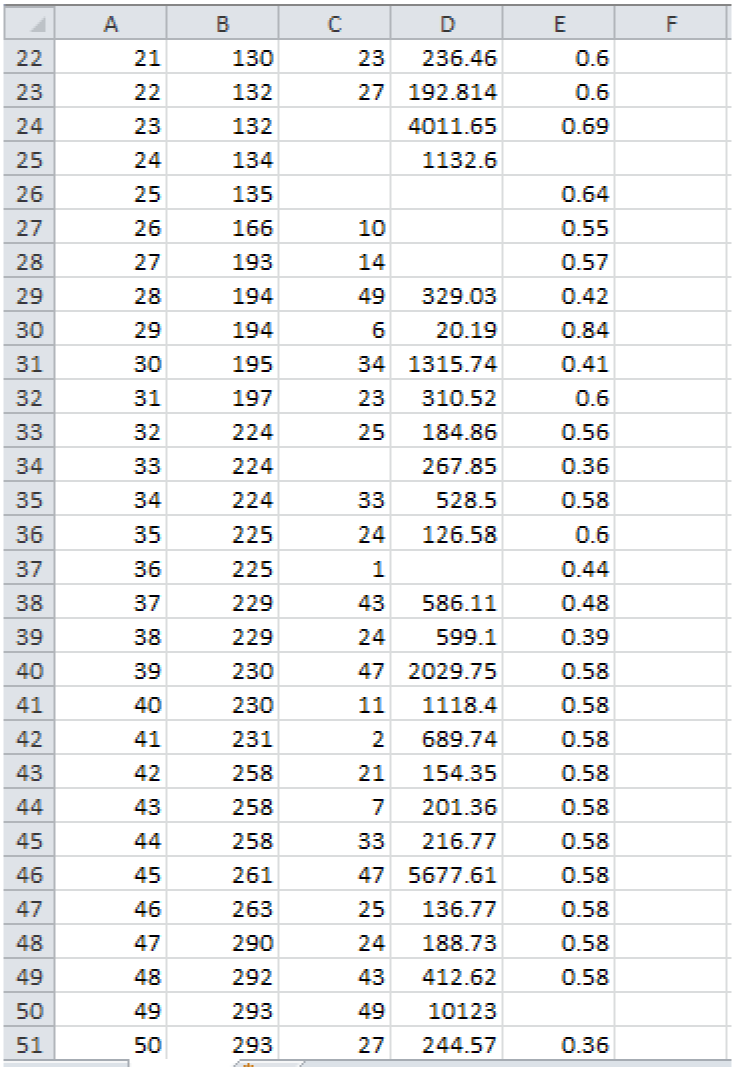
Question: Demonstrate possible missing value analysis approaches using any real world data

Dataset Used: Train dataset containing Row ID, Order ID, Order Quantity, Sales and Profit attributes.

Data:



	A	B	C	D	E	F
1	Row ID	Order ID	Order Qu	Sales	profit	
2	1	3	7	261.54	0.8	
3	2	6	6	-6.93		
4	3	32	-90	2808.08	0.65	
5	4	32		1761.4	0.72	
6	5	32			0.6	
7	6	32	15	140.56	0.6	
8	7	35	-30	288.56		
9	8	35	14	1892.85		
10	9	36	46	2484.75	0.55	
11	10	65	-32		0.49	
12	11	66		108.15	0.56	
13	12	69		1186.06	0.44	
14	13	69	28		0.6	
15	14	70	48		0.82	
16	15	70	46	7804.53	0.59	
17	16	96	37	4158.12	0.55	
18	17	97	26	75.57	0.38	
19	18	129	4	32.72	0.37	
20	19	130	3		0.38	
21	20	130	29	575.11	0.37	
22	21	130	23	236.46	0.6	
23	22	132	27	192.814	0.6	
24	23	132		4011.65	0.69	
25	24	134		1132.6		
26	25	135			0.64	
27	26	166	10		0.55	
28	27	193	14		0.57	
29	28	194	49	329.03	0.42	
30	29	194	6	20.19	0.84	



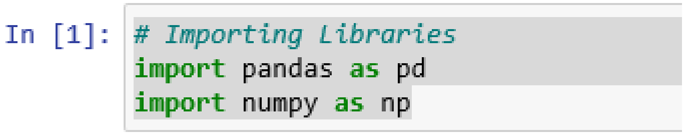
	A	B	C	D	E	F
22	21	130	23	236.46	0.6	
23	22	132	27	192.814	0.6	
24	23	132		4011.65	0.69	
25	24	134		1132.6		
26	25	135			0.64	
27	26	166	10		0.55	
28	27	193	14		0.57	
29	28	194	49	329.03	0.42	
30	29	194	6	20.19	0.84	
31	30	195	34	1315.74	0.41	
32	31	197	23	310.52	0.6	
33	32	224	25	184.86	0.56	
34	33	224		267.85	0.36	
35	34	224	33	528.5	0.58	
36	35	225	24	126.58	0.6	
37	36	225	1		0.44	
38	37	229	43	586.11	0.48	
39	38	229	24	599.1	0.39	
40	39	230	47	2029.75	0.58	
41	40	230	11	1118.4	0.58	
42	41	231	2	689.74	0.58	
43	42	258	21	154.35	0.58	
44	43	258	7	201.36	0.58	
45	44	258	33	216.77	0.58	
46	45	261	47	5677.61	0.58	
47	46	263	25	136.77	0.58	
48	47	290	24	188.73	0.58	
49	48	292	43	412.62	0.58	
50	49	293	49	10123		
51	50	293	27	244.57	0.36	

Procedure:

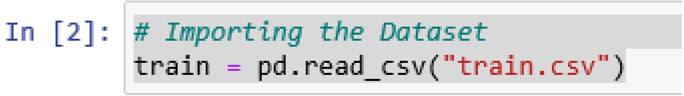
- We first import the dataset into our workspace s.
- We then find the attributes which are having null values in them.
- we check first few rows of the dataset to know what is the value used for missing data, i.e., Nan/Null/Blank/0 or -1.
- Then we check how many null values are there in each attribute.
- We replace all the missing values either with mean of nonnull values or by median of non-null values.
- We then see which has better consistency with data set, mean or median.

- At Last, we replace the missing value with the better fills…

Importing Libraries , Importing Data set(train.csv)

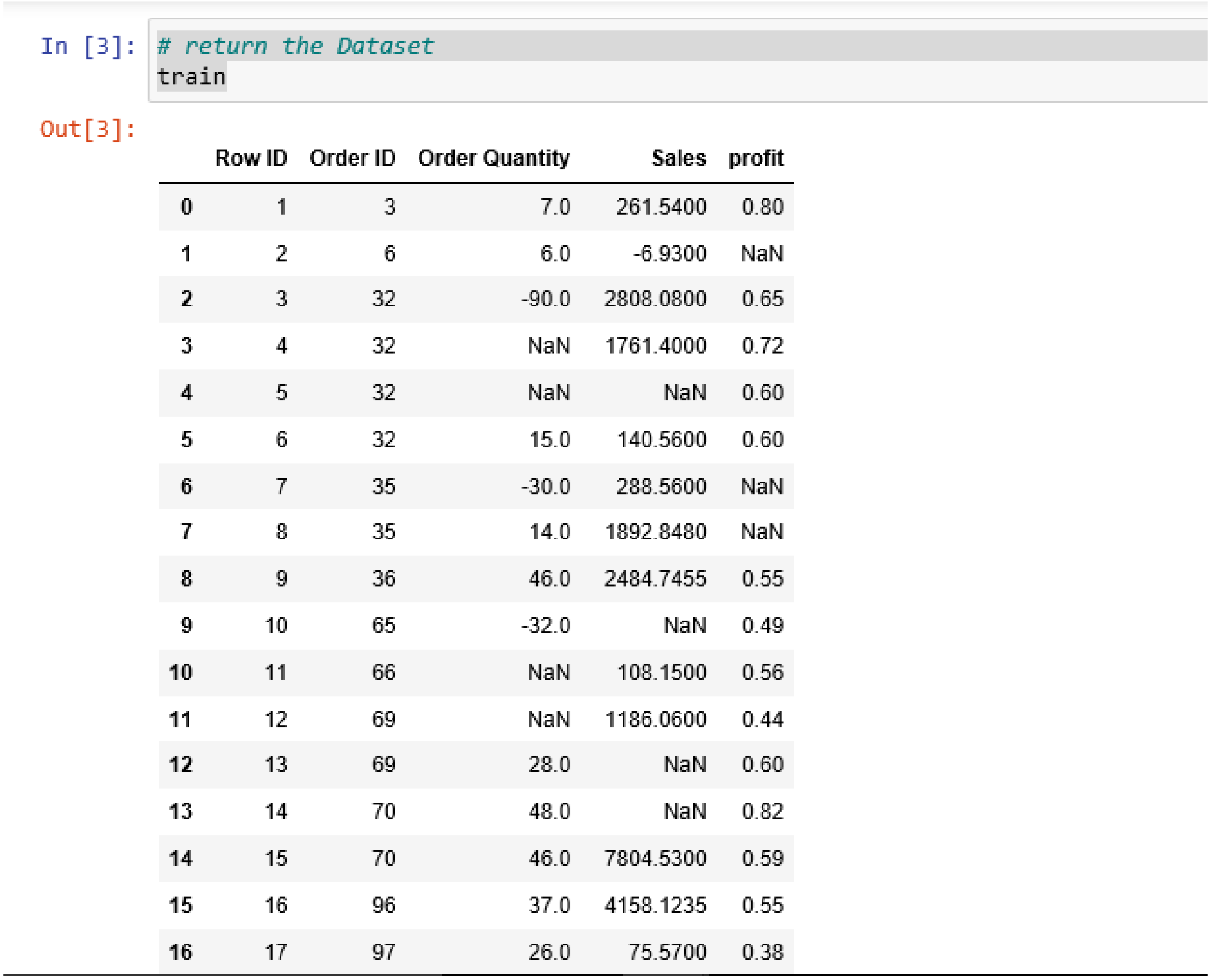


```
In [1]: # Importing Libraries
        import pandas as pd
        import numpy as np
```



```
In [2]: # Importing the Dataset
        train = pd.read_csv("train.csv")
```

Printing the data set

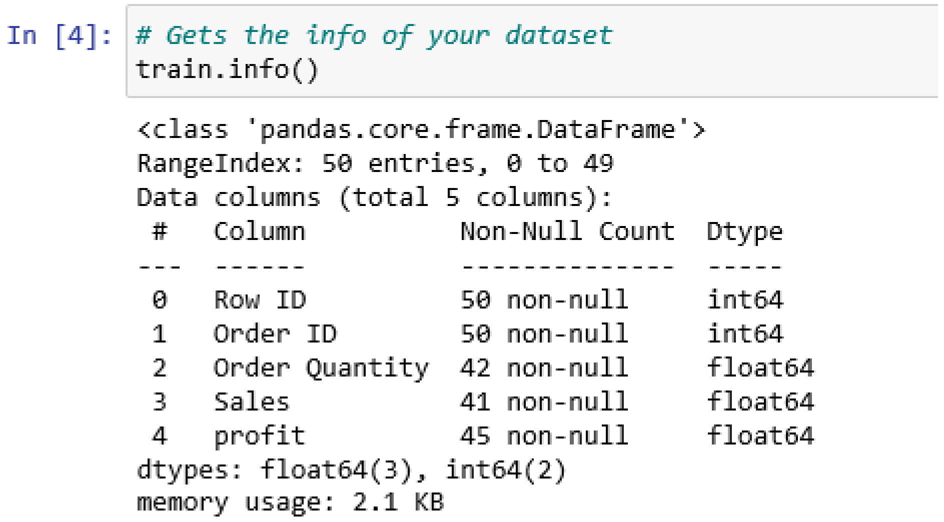


```
In [3]: # return the Dataset
        train
```

Out[3]:

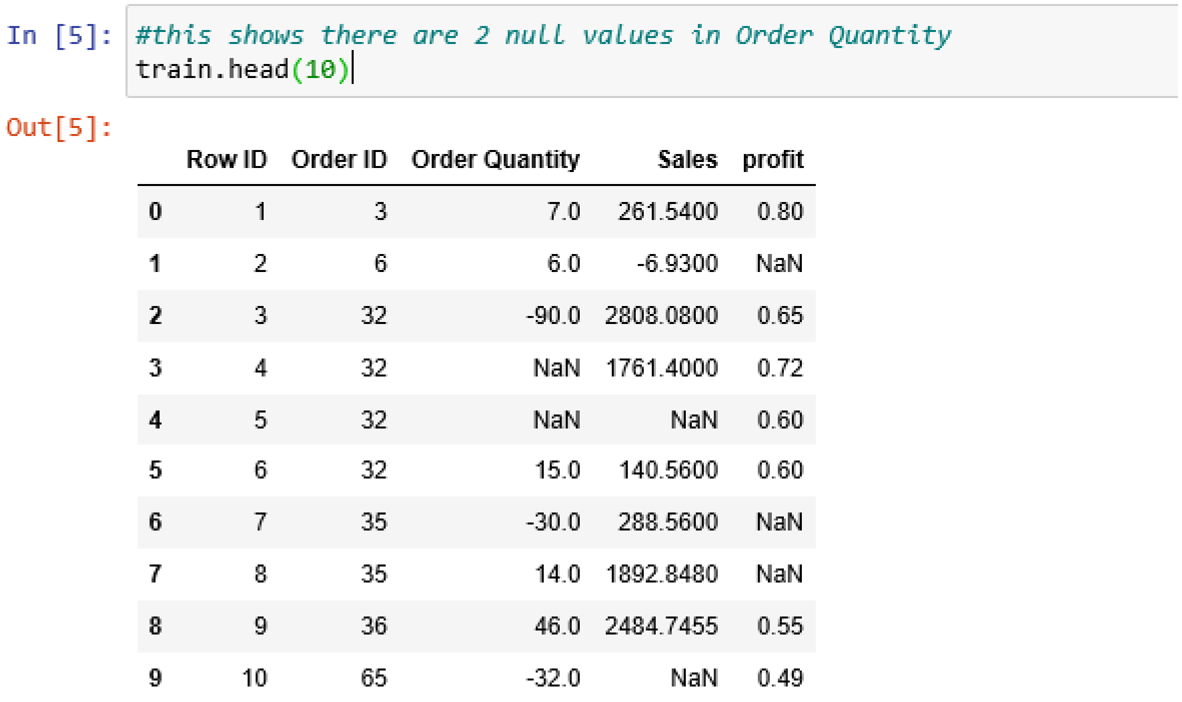
	Row ID	Order ID	Order Quantity	Sales	profit
0	1	3	7.0	261.5400	0.80
1	2	6	6.0	-6.9300	NaN
2	3	32	-90.0	2808.0800	0.65
3	4	32	NaN	1761.4000	0.72
4	5	32	NaN	NaN	0.60
5	6	32	15.0	140.5600	0.60
6	7	35	-30.0	288.5600	NaN
7	8	35	14.0	1892.8480	NaN
8	9	36	46.0	2484.7455	0.55
9	10	65	-32.0	NaN	0.49
10	11	66	NaN	108.1500	0.56
11	12	69	NaN	1186.0600	0.44
12	13	69	28.0	NaN	0.60
13	14	70	48.0	NaN	0.82
14	15	70	46.0	7804.5300	0.59
15	16	96	37.0	4158.1235	0.55
16	17	97	26.0	75.5700	0.38

Returns the info of your dataset



```
In [4]: # Gets the info of your dataset
        train.info()

        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 50 entries, 0 to 49
        Data columns (total 5 columns):
         #   Column          Non-Null Count  Dtype  
        ---  ------          --------------  -----  
         0   Row ID          50 non-null     int64  
         1   Order ID        50 non-null     int64  
         2   Order Quantity  42 non-null     float64
         3   Sales           41 non-null     float64
         4   profit          45 non-null     float64
        dtypes: float64(3), int64(2)
        memory usage: 2.1 KB
```

Display the first 10 rows of the data



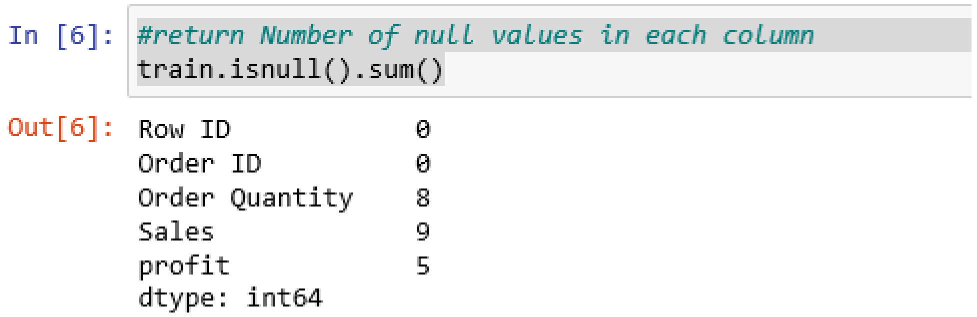
```
In [5]: #this shows there are 2 null values in Order Quantity
        train.head(10)
```

Out[5]:

	Row ID	Order ID	Order Quantity	Sales	profit
0	1	3	7.0	261.5400	0.80
1	2	6	6.0	-6.9300	NaN
2	3	32	-90.0	2808.0800	0.65
3	4	32	NaN	1761.4000	0.72
4	5	32	NaN	NaN	0.60
5	6	32	15.0	140.5600	0.60
6	7	35	-30.0	288.5600	NaN
7	8	35	14.0	1892.8480	NaN
8	9	36	46.0	2484.7455	0.55
9	10	65	-32.0	NaN	0.49

From here we can know the missing values are assigned with NaN value.

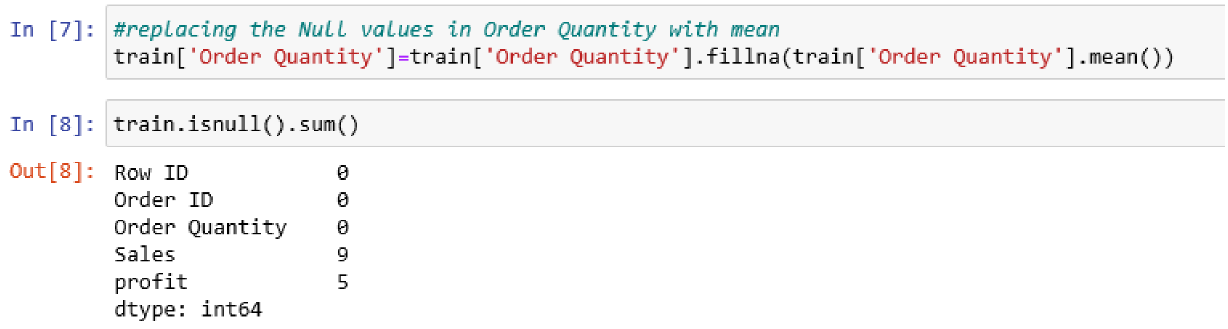
Returns Number of Null values in each column



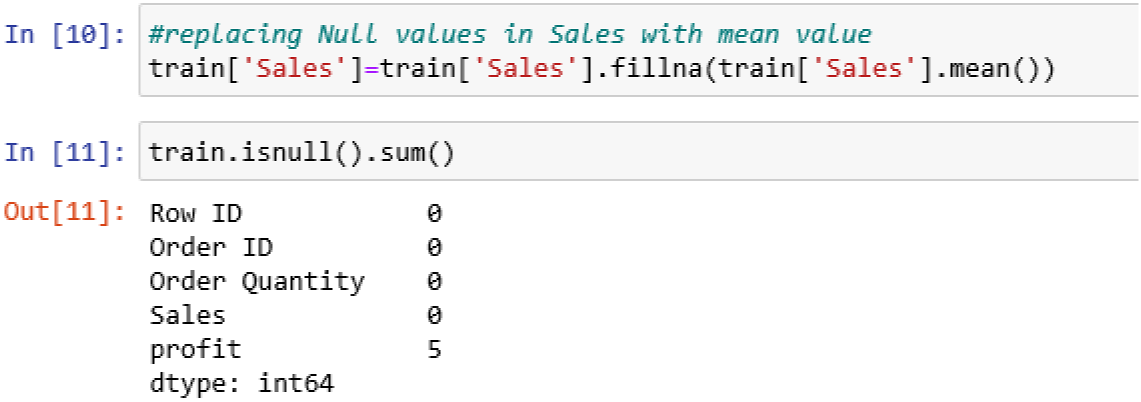
```
In [6]: #return Number of null values in each column
        train.isnull().sum()

Out[6]: Row ID            0
        Order ID          0
        Order Quantity    8
        Sales             9
        profit            5
        dtype: int64
```

This cell informs us that there are 8 null values in Order Quantity, 9 in Sales and 5 in profit.

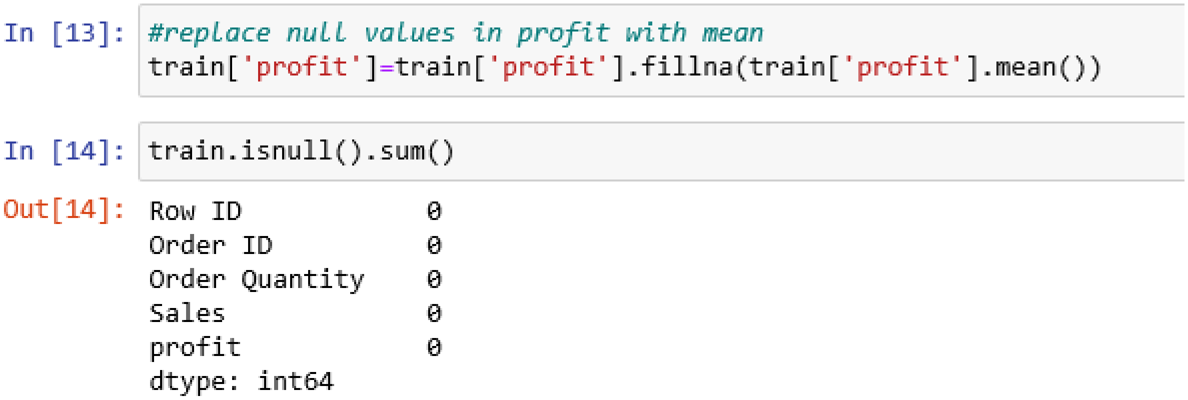
Replacing Null values with mean of remaining values



```
In [7]: #replacing the Null values in Order Quantity with mean
        train['Order Quantity']=train['Order Quantity'].fillna(train['Order Quantity'].mean())

In [8]: train.isnull().sum()

Out[8]: Row ID            0
        Order ID          0
        Order Quantity    0
        Sales             9
        profit            5
        dtype: int64
```



```
In [10]: #replacing Null values in Sales with mean value
         train['Sales']=train['Sales'].fillna(train['Sales'].mean())

In [11]: train.isnull().sum()

Out[11]: Row ID            0
         Order ID          0
         Order Quantity    0
         Sales             0
         profit            5
         dtype: int64
```



```
In [13]: #replace null values in profit with mean
         train['profit']=train['profit'].fillna(train['profit'].mean())

In [14]: train.isnull().sum()

Out[14]: Row ID            0
         Order ID          0
         Order Quantity    0
         Sales             0
         profit            0
         dtype: int64
```

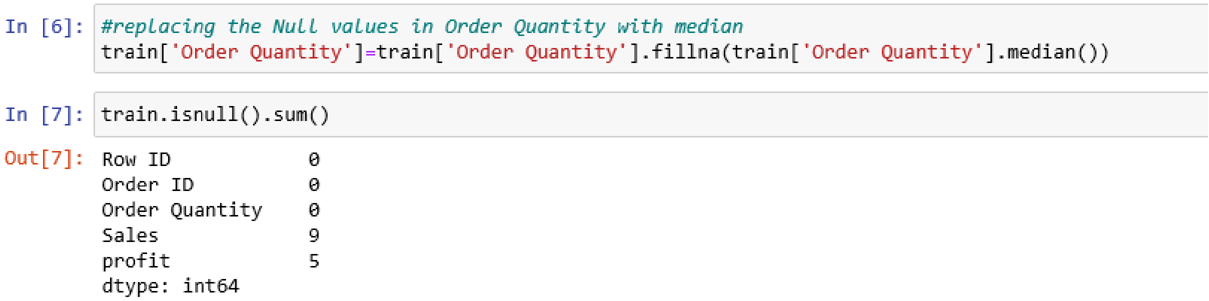
The values with which null values are replaced-

In Order Quantity ---> 19.738095

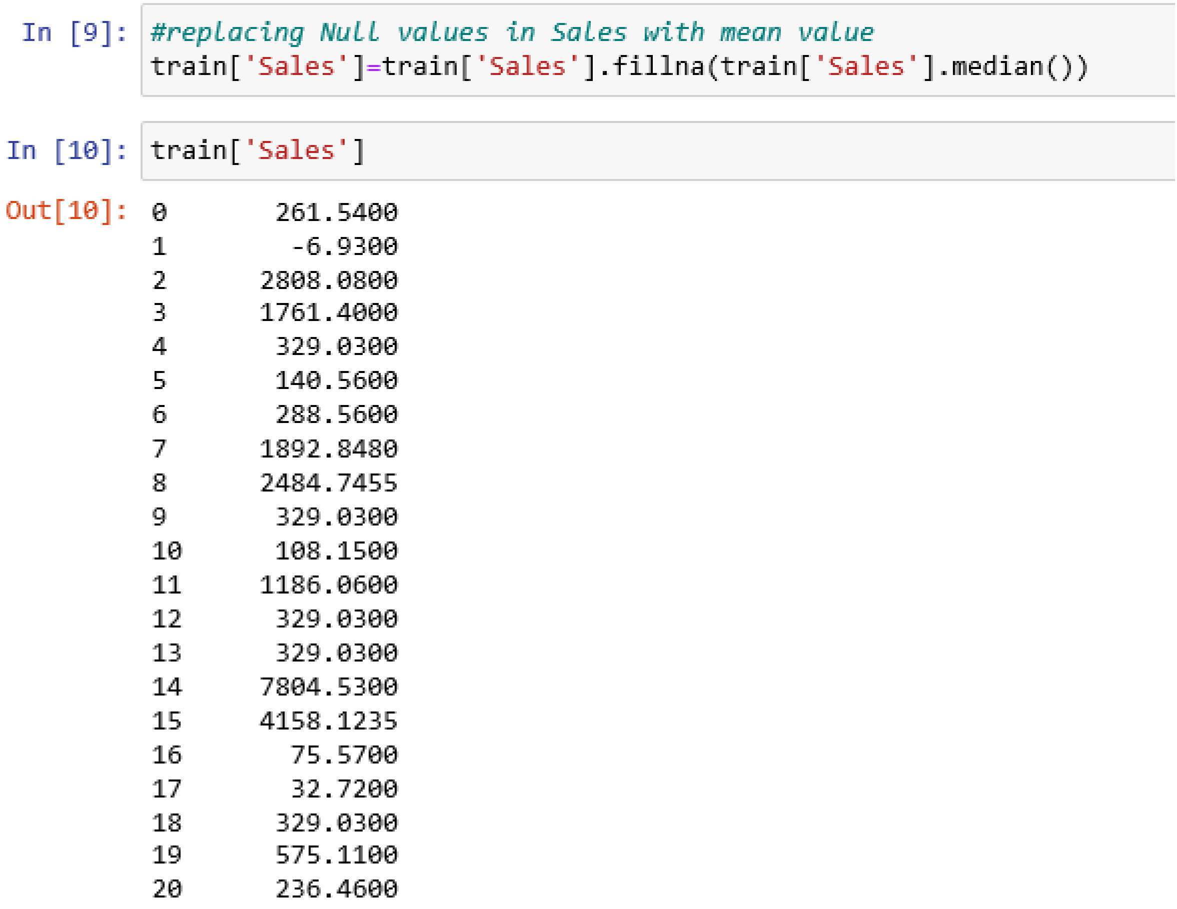
In Sales   ---> 1331.872098

In profit   ---> 0.554667

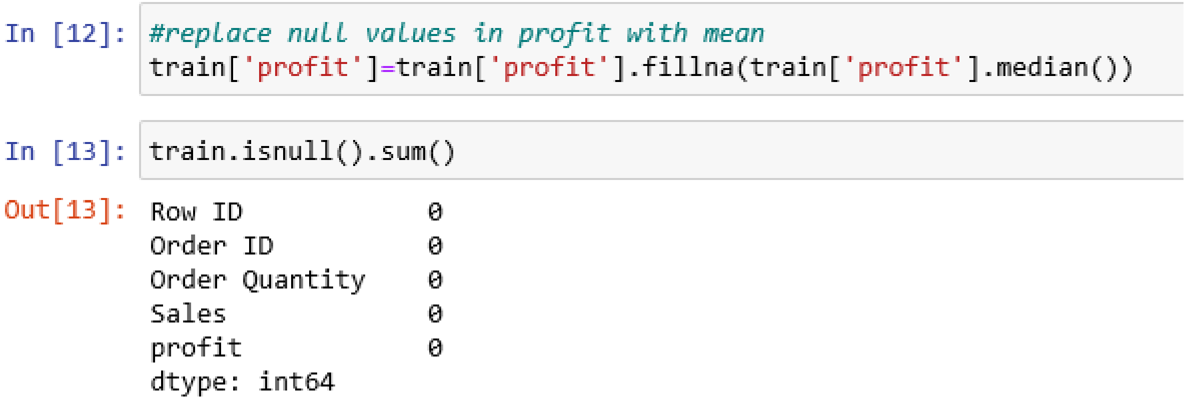
Filling with Median Values.



```
In [6]: #replacing the Null values in Order Quantity with median
        train['Order Quantity']=train['Order Quantity'].fillna(train['Order Quantity'].median())

In [7]: train.isnull().sum()

Out[7]: Row ID            0
        Order ID          0
        Order Quantity    0
        Sales             9
        profit            5
        dtype: int64
```



```
In [9]:  #replacing Null values in Sales with mean value
         train['Sales']=train['Sales'].fillna(train['Sales'].median())

In [10]: train['Sales']

Out[10]: 0       261.5400
         1        -6.9300
         2      2808.0800
         3      1761.4000
         4       329.0300
         5       140.5600
         6       288.5600
         7      1892.8480
         8      2484.7455
         9       329.0300
         10      108.1500
         11     1186.0600
         12      329.0300
         13      329.0300
         14     7804.5300
         15     4158.1235
         16       75.5700
         17       32.7200
         18      329.0300
         19      575.1100
         20      236.4600
```



```
In [12]: #replace null values in profit with mean
         train['profit']=train['profit'].fillna(train['profit'].median())

In [13]: train.isnull().sum()

Out[13]: Row ID            0
         Order ID          0
         Order Quantity    0
         Sales             0
         profit            0
         dtype: int64
```

The values with which null values are replaced-

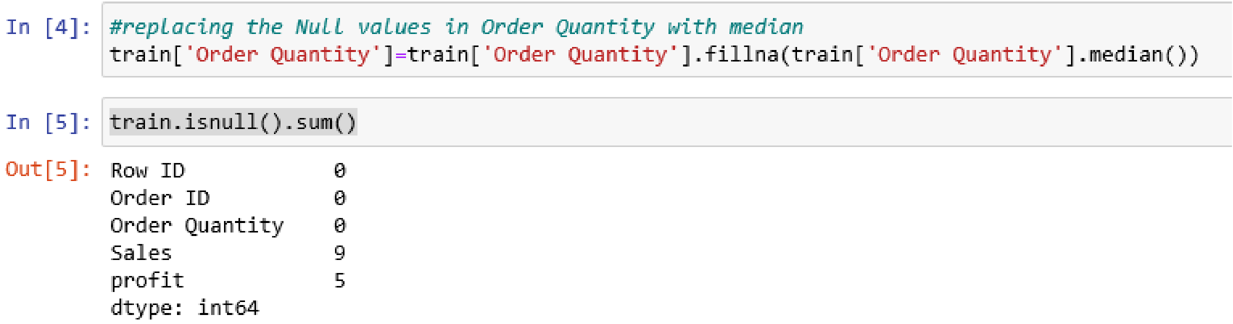
In Order Quantity ---> 24

In Sales ---> 329.0300

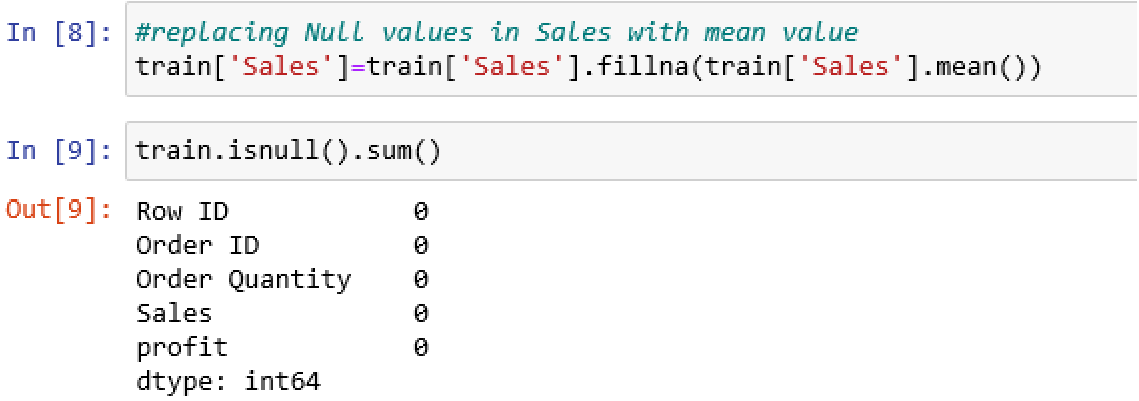
In profit ---> 0.58

Result and Conclusion: -

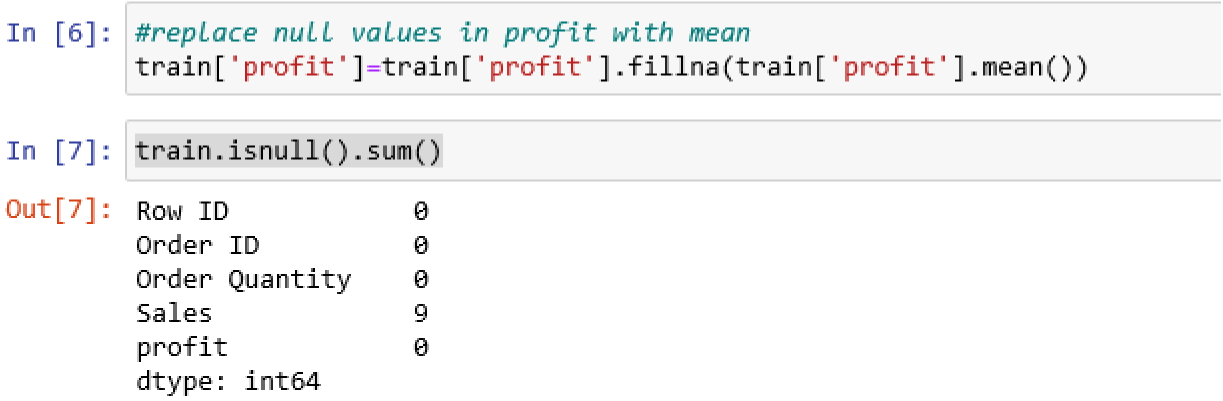
The result seems to be more consistent when we use median value to fill missing values of Order Quantity and mean value for Sales. The median and mean value for profit attribute is near about same, and thus we can use either of them.



```
In [4]: #replacing the Null values in Order Quantity with median
        train['Order Quantity']=train['Order Quantity'].fillna(train['Order Quantity'].median())

In [5]: train.isnull().sum()

Out[5]: Row ID            0
        Order ID          0
        Order Quantity    0
        Sales             9
        profit            5
        dtype: int64
```

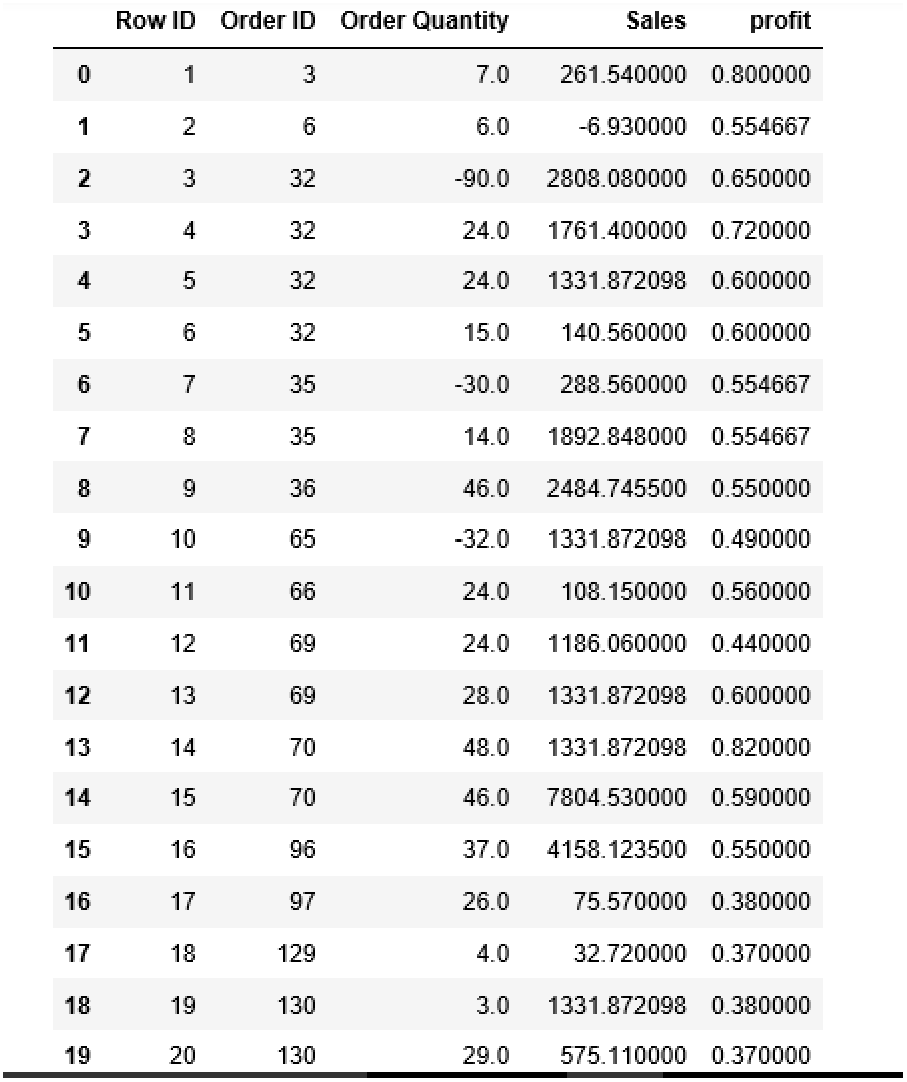


```
In [8]: #replacing Null values in Sales with mean value
        train['Sales']=train['Sales'].fillna(train['Sales'].mean())

In [9]: train.isnull().sum()

Out[9]: Row ID            0
        Order ID          0
        Order Quantity    0
        Sales             0
        profit            0
        dtype: int64
```

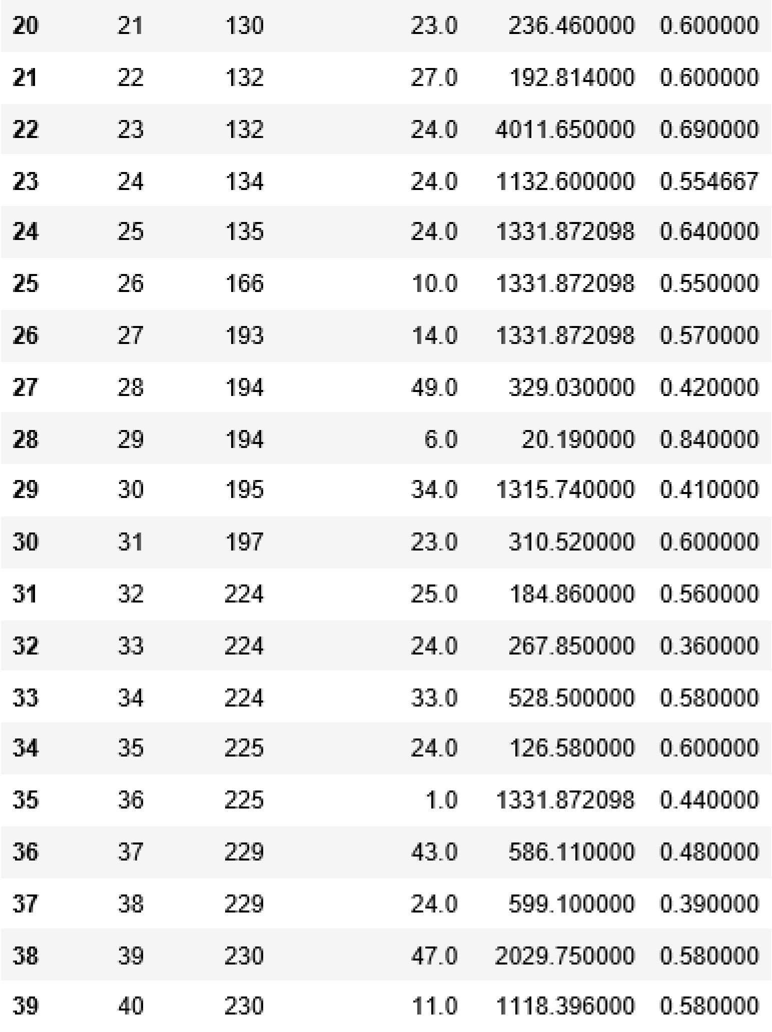


```
In [6]: #replace null values in profit with mean
        train['profit']=train['profit'].fillna(train['profit'].mean())

In [7]: train.isnull().sum()

Out[7]: Row ID            0
        Order ID          0
        Order Quantity    0
        Sales             9
        profit            0
        dtype: int64
```

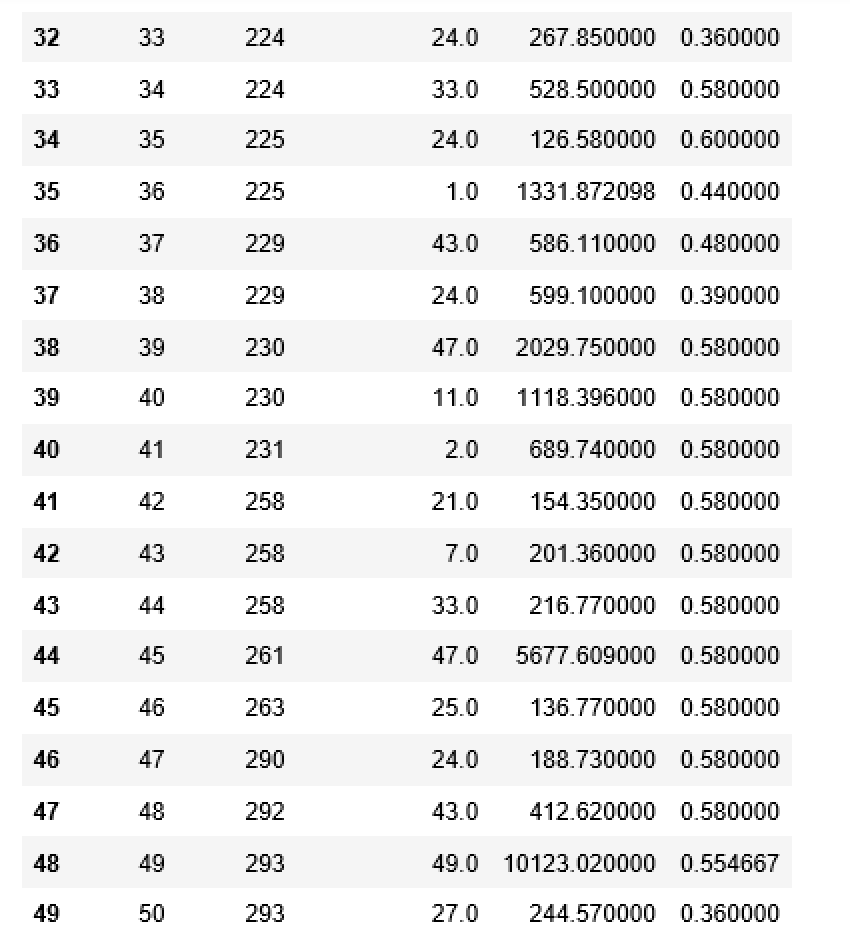
The train dataset after using median for Order Quantity, mean for Sales and median for profit is-



	Row ID	Order ID	Order Quantity	Sales	profit
0	1	3	7.0	261.540000	0.800000
1	2	6	6.0	-6.930000	0.554667
2	3	32	-90.0	2808.080000	0.650000
3	4	32	24.0	1761.400000	0.720000
4	5	32	24.0	1331.872098	0.600000
5	6	32	15.0	140.560000	0.600000
6	7	35	-30.0	288.560000	0.554667
7	8	35	14.0	1892.848000	0.554667
8	9	36	46.0	2484.745500	0.550000
9	10	65	-32.0	1331.872098	0.490000
10	11	66	24.0	108.150000	0.560000
11	12	69	24.0	1186.060000	0.440000
12	13	69	28.0	1331.872098	0.600000
13	14	70	48.0	1331.872098	0.820000
14	15	70	46.0	7804.530000	0.590000
15	16	96	37.0	4158.123500	0.550000
16	17	97	26.0	75.570000	0.380000
17	18	129	4.0	32.720000	0.370000
18	19	130	3.0	1331.872098	0.380000
19	20	130	29.0	575.110000	0.370000



20	21	130	23.0	236.460000	0.600000
21	22	132	27.0	192.814000	0.600000
22	23	132	24.0	4011.650000	0.690000
23	24	134	24.0	1132.600000	0.554667
24	25	135	24.0	1331.872098	0.640000
25	26	166	10.0	1331.872098	0.550000
26	27	193	14.0	1331.872098	0.570000
27	28	194	49.0	329.030000	0.420000
28	29	194	6.0	20.190000	0.840000
29	30	195	34.0	1315.740000	0.410000
30	31	197	23.0	310.520000	0.600000
31	32	224	25.0	184.860000	0.560000
32	33	224	24.0	267.850000	0.360000
33	34	224	33.0	528.500000	0.580000
34	35	225	24.0	126.580000	0.600000
35	36	225	1.0	1331.872098	0.440000
36	37	229	43.0	586.110000	0.480000
37	38	229	24.0	599.100000	0.390000
38	39	230	47.0	2029.750000	0.580000
39	40	230	11.0	1118.396000	0.580000



32	33	224	24.0	267.850000	0.360000
33	34	224	33.0	528.500000	0.580000
34	35	225	24.0	126.580000	0.600000
35	36	225	1.0	1331.872098	0.440000
36	37	229	43.0	586.110000	0.480000
37	38	229	24.0	599.100000	0.390000
38	39	230	47.0	2029.750000	0.580000
39	40	230	11.0	1118.396000	0.580000
40	41	231	2.0	689.740000	0.580000
41	42	258	21.0	154.350000	0.580000
42	43	258	7.0	201.360000	0.580000
43	44	258	33.0	216.770000	0.580000
44	45	261	47.0	5677.609000	0.580000
45	46	263	25.0	136.770000	0.580000
46	47	290	24.0	188.730000	0.580000
47	48	292	43.0	412.620000	0.580000
48	49	293	49.0	10123.020000	0.554667
49	50	293	27.0	244.570000	0.360000

The values with which null values are replaced after comparing median and mean values-

In Order Quantity ---> 24

In Sales ---> 1331.872098

In profit ---> 0.554667