statisticalInformations

May 8, 2025

[3]:
```
import numpy as np
import pandas as pd
```

[5]:
```
df=pd.read_csv("sales_data_sample.csv",encoding="latin1")
```

[7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2823 entries, 0 to 2822
Data columns (total 25 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   ORDERNUMBER       2823 non-null   int64
 1   QUANTITYORDERED   2823 non-null   int64
 2   PRICEEACH         2823 non-null   float64
 3   ORDERLINENUMBER   2823 non-null   int64
 4   SALES             2823 non-null   float64
 5   ORDERDATE         2823 non-null   object
 6   STATUS            2823 non-null   object
 7   QTR_ID            2823 non-null   int64
 8   MONTH_ID          2823 non-null   int64
 9   YEAR_ID           2823 non-null   int64
 10  PRODUCTLINE       2823 non-null   object
 11  MSRP              2823 non-null   int64
 12  PRODUCTCODE       2823 non-null   object
 13  CUSTOMERNAME      2823 non-null   object
 14  PHONE             2823 non-null   object
 15  ADDRESSLINE1      2823 non-null   object
 16  ADDRESSLINE2      302 non-null    object
 17  CITY              2823 non-null   object
 18  STATE             1337 non-null   object
 19  POSTALCODE        2747 non-null   object
 20  COUNTRY           2823 non-null   object
 21  TERRITORY         1749 non-null   object
 22  CONTACTLASTNAME   2823 non-null   object
 23  CONTACTFIRSTNAME  2823 non-null   object
 24  DEALSIZE          2823 non-null   object
dtypes: float64(2), int64(7), object(16)
```

```
memory usage: 551.5+ KB
```

[8]:
```
df["PRODUCTLINE"]
```

[8]:
```
0       Motorcycles
1       Motorcycles
2       Motorcycles
3       Motorcycles
4       Motorcycles
           …
2818          Ships
2819          Ships
2820          Ships
2821          Ships
2822          Ships
Name: PRODUCTLINE, Length: 2823, dtype: object
```

[9]:
```
numericData=df.select_dtypes(include=np.number)
```

[10]:
```
numericData
```

[10]:
```
      ORDERNUMBER  QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  \
0           10107               30      95.70                2  2871.00
1           10121               34      81.35                5  2765.90
2           10134               41      94.74                2  3884.34
3           10145               45      83.26                6  3746.70
4           10159               49     100.00               14  5205.27
…             …                …          …                …        …
2818        10350               20     100.00               15  2244.40
2819        10373               29     100.00                1  3978.51
2820        10386               43     100.00                4  5417.57
2821        10397               34      62.24                1  2116.16
2822        10414               47      65.52                9  3079.44

      QTR_ID  MONTH_ID  YEAR_ID  MSRP
0          1         2     2003    95
1          2         5     2003    95
2          3         7     2003    95
3          3         8     2003    95
4          4        10     2003    95
…       …         …       …     …
2818       4        12     2004    54
2819       1         1     2005    54
2820       1         3     2005    54
2821       1         3     2005    54
2822       2         5     2005    54
```

```
[2823 rows x 9 columns]
```

[12]:
```
meanValues=numericData.mean()
meanValues
```

[12]:
```
ORDERNUMBER        10258.725115
QUANTITYORDERED       35.092809
PRICEEACH             83.658544
ORDERLINENUMBER        6.466171
SALES               3553.889072
QTR_ID                 2.717676
MONTH_ID               7.092455
YEAR_ID             2003.815090
MSRP                 100.715551
dtype: float64
```

[13]:
```
meadian_values=numericData.median()
meadian_values
```

[13]:
```
ORDERNUMBER        10262.0
QUANTITYORDERED       35.0
PRICEEACH             95.7
ORDERLINENUMBER        6.0
SALES               3184.8
QTR_ID                 3.0
MONTH_ID               8.0
YEAR_ID             2004.0
MSRP                  99.0
dtype: float64
```

[15]:
```
standardDeviation=numericData.std()
standardDeviation
```

[15]:
```
ORDERNUMBER          92.085478
QUANTITYORDERED       9.741443
PRICEEACH            20.174277
ORDERLINENUMBER       4.225841
SALES              1841.865106
QTR_ID                1.203878
MONTH_ID              3.656633
YEAR_ID               0.699670
MSRP                 40.187912
dtype: float64
```

[ ]:
```

```