RELIANCE INDUSTRIES

Data Set https://www.kaggle.com/datasets/notshrirang/reliance-stock-price-dataset (https://www.kaggle.com/datasets/notshrirang/reliance-stock-price-dataset)

In [37]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

In [69]:
```
data= pd.read_csv("reliance_data.csv")
data.tail(30)
```

Out[69]:

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover	Trades	Deliverable Volume	%Deliverb
6175	19-10-2020	RELIANCE	EQ	2175.80	2190.05	2228.70	2155.15	2175.50	2176.20	2189.75	14399062	3.153030e+15	363717.0	3772907.0	0.262
6176	20-10-2020	RELIANCE	EQ	2176.20	2179.00	2193.00	2152.25	2155.85	2155.90	2166.54	8529621	1.847970e+15	275082.0	2119328.0	0.248
6177	21-10-2020	RELIANCE	EQ	2155.90	2168.00	2192.00	2097.75	2122.65	2124.60	2143.87	15729989	3.372310e+15	399065.0	3975978.0	0.252
6178	22-10-2020	RELIANCE	EQ	2124.60	2127.40	2132.50	2091.00	2111.90	2106.95	2107.04	14215255	2.995210e+15	391498.0	5836281.0	0.410
6179	23-10-	RELIANCE	EQ	2106.95	2106.00	2135.00	2096.40	2112.00	2113.05	2118.90	10809383	2.290410e+15	265187.0	3551502.0	0.328

In [15]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6205 entries, 0 to 6204
Data columns (total 15 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Date                6205 non-null   object 
 1   Symbol              6205 non-null   object 
 2   Series              6205 non-null   object 
 3   Prev Close          6205 non-null   float64
 4   Open                6205 non-null   float64
 5   High                6205 non-null   float64
 6   Low                 6205 non-null   float64
 7   Last                5657 non-null   float64
 8   Close               6205 non-null   float64
 9   VWAP                6205 non-null   float64
 10  Volume              6205 non-null   int64  
 11  Turnover            6205 non-null   float64
 12  Trades              2356 non-null   float64
 13  Deliverable Volume  4693 non-null   float64
 14  %Deliverble         4693 non-null   float64
dtypes: float64(11), int64(1), object(3)
memory usage: 727.3+ KB
```

In [16]:
```
data.columns
```

Out[16]:
```
Index(['Date', 'Symbol', 'Series', 'Prev Close', 'Open', 'High', 'Low', 'Last',
       'Close', 'VWAP', 'Volume', 'Turnover', 'Trades', 'Deliverable Volume',
       '%Deliverble'],
      dtype='object')
```

In [17]:
```
data.isnull().sum().sort_values(ascending=False)
```

Out[17]:
```
Trades                3849
Deliverable Volume    1512
%Deliverble           1512
Last                   548
Date                     0
Symbol                   0
Series                   0
Prev Close               0
Open                     0
High                     0
Low                      0
Close                    0
VWAP                     0
Volume                   0
Turnover                 0
dtype: int64
```

In [18]:
```
data.isnull().sum().sort_values(ascending=False)/len(data)*100
```

Out[18]:
```
Trades                62.030620
Deliverable Volume    24.367446
%Deliverble           24.367446
Last                   8.831587
Date                   0.000000
Symbol                 0.000000
Series                 0.000000
Prev Close             0.000000
Open                   0.000000
High                   0.000000
Low                    0.000000
Close                  0.000000
VWAP                   0.000000
Volume                 0.000000
Turnover               0.000000
dtype: float64
```

we will discard the values having *Null Values>40%*

In [19]:
```
# number of columns before column drop

print(f"number of column= {data.shape[1]}")
```

```
number of column= 15
```

In [20]:
```
data2= data.drop(['Trades'], axis=1)
```

In [22]:
```
#number of columns after column drop

print(f"number of column= {data2.shape[1]}")
```

```
number of column= 14
```

In [25]:
```
data2.isnull().sum().sort_values(ascending=False)/len(data2)*100
```

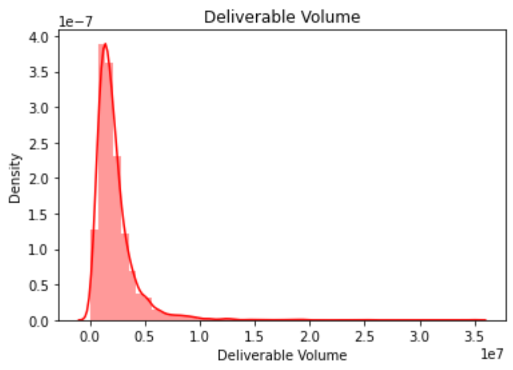
Out[25]:
```
Deliverable Volume    24.367446
%Deliverble           24.367446
Last                   8.831587
Date                   0.000000
Symbol                 0.000000
Series                 0.000000
Prev Close             0.000000
Open                   0.000000
High                   0.000000
Low                    0.000000
Close                  0.000000
VWAP                   0.000000
Volume                 0.000000
Turnover               0.000000
dtype: float64
```

In [27]:
```
data2['Last'].head()
```

Out[27]:
```
0   NaN
1   NaN
2   NaN
3   NaN
4   NaN
Name: Last, dtype: float64
```

Deliverable Volume

In [38]:
```
sns.distplot(data2['Deliverable Volume'], color='red').set(title='Deliverable Volume')
plt.show()
print(f"Skewness of the data is {data2['Deliverable Volume'].skew()}")
```



```
Skewness of the data is 4.230016870943885
```

As the data is highly skewed, therefore we use Mode to replace Null values in column [Deliverable Volume]

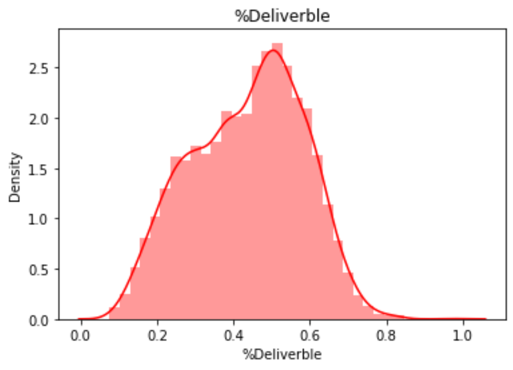
%Deliverble

In [42]:
```
data2['%Deliverble'].tail()
```

Out[42]:
```
6200    0.2461
6201    0.2251
6202    0.2019
6203    0.2857
6204    0.5717
Name: %Deliverble, dtype: float64
```

In [47]:
```
sns.distplot(data2['%Deliverble'], color='red').set(title='%Deliverble')
plt.show()

print(f"Skewness of the data is {data2['%Deliverble'].skew()}")
```



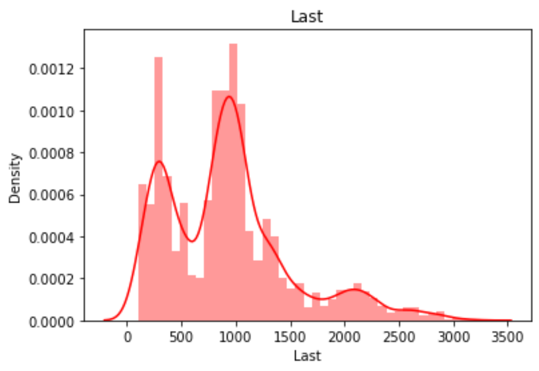
```
Skewness of the data is -0.15570560526885016
```

As the data is fairly symmetrical, therefore we use Mean to replace Null values in column [%Deliverble]

Last

In [51]:
```
sns.distplot(data2['Last'], color='red').set(title='Last')
plt.show()

print(f"Skewness of the data is {data2['Last'].skew()}")
```



```
Skewness of the data is 0.9930201110592883
```

As the data is moderately skewed, therefore we use Median to replace Null values in column [%Deliverble]

Imputing Missing values

Deliverable Volume

Imputing missing values with mode

In [58]:
```
print(f"Number of Null values are {data2['Deliverable Volume'].isnull().sum()}")
```

```
Number of Null values are 1512
```

In [53]:
```
data2['Deliverable Volume'].mode()[0]
```

Out[53]:
```
518905.0
```

In [59]:
```
data2['Deliverable Volume']= data2['Deliverable Volume'].fillna(data2['Deliverable Volume'].mode()[0])
```

In [60]:
```
print(f"Number of Null values are {data2['Deliverable Volume'].isnull().sum()}")
```

```
Number of Null values are 0
```

%Deliverble

Imputing missing values with mean

In [62]:
```
data2.columns
```

Out[62]:
```
Index(['Date', 'Symbol', 'Series', 'Prev Close', 'Open', 'High', 'Low', 'Last',
       'Close', 'VWAP', 'Volume', 'Turnover', 'Deliverable Volume',
       '%Deliverble'],
      dtype='object')
```

In [65]:
```
data2['%Deliverble'].mean()
```

Out[65]:
```
0.43633160025570006
```

In [64]:
```
print(f"Number of Null values are {data2['%Deliverble'].isnull().sum()}")
```

```
Number of Null values are 1512
```

In [66]:
```
data2['%Deliverble']= data2['%Deliverble'].fillna(data2['%Deliverble'].mean())
```

In [67]:
```
print(f"Number of Null values are {data2['%Deliverble'].isnull().sum()}")
```

```
Number of Null values are 0
```

Last

Imputing missing values with median

```
In [71]: data2['Last'].median()
```

```
Out[71]: 892.35
```

```
In [72]: print(f"Number of Null values are {data2['Last'].isnull().sum()}")
```

```
Number of Null values are 548
```

```
In [73]: data2['Last'] = data2['Last'].fillna(data2['Last'].median())
```

```
In [74]: print(f"Number of Null values are {data2['Last'].isnull().sum()}")
```

```
Number of Null values are 0
```

Export Clean data to CSV

```
In [75]: data2.to_csv(r'C:\Users\WIN8\Desktop\Reliance\reliance_data_clean.csv')
```

```
In [ ]:
```