```
In [ ]:
```

# Time Series Sales Prediction with EDA

```
In [3]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        import statsmodels as st
        from datetime import datetime
        plt.style.use('fivethirtyeight')
        from matplotlib.pylab import rcParams
        rcParams['figure.figsize']=15,5
```

In [4]:
```python
data_original=pd.read_excel("Superstore.xls")
data_original
```

Out[4]:

| | Row ID | Order ID | Order Date | Ship Date | Ship Mode | Customer ID | Customer Name | Segment | Country | |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | CA-2016-152156 | 2016-11-08 | 2016-11-11 | Second Class | CG-12520 | Claire Gute | Consumer | United States | H |
| **1** | 2 | CA-2016-152156 | 2016-11-08 | 2016-11-11 | Second Class | CG-12520 | Claire Gute | Consumer | United States | H |
| **2** | 3 | CA-2016-138688 | 2016-06-12 | 2016-06-16 | Second Class | DV-13045 | Darrin Van Huff | Corporate | United States | Lo |
| **3** | 4 | US-2015-108966 | 2015-10-11 | 2015-10-18 | Standard Class | SO-20335 | Sean O'Donnell | Consumer | United States | L |
| **4** | 5 | US-2015-108966 | 2015-10-11 | 2015-10-18 | Standard Class | SO-20335 | Sean O'Donnell | Consumer | United States | L |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **9989** | 9990 | CA-2014-110422 | 2014-01-21 | 2014-01-23 | Second Class | TB-21400 | Tom Boeckenhauer | Consumer | United States | |
| **9990** | 9991 | CA-2017-121258 | 2017-02-26 | 2017-03-03 | Standard Class | DB-13060 | Dave Brooks | Consumer | United States | C |
| **9991** | 9992 | CA-2017-121258 | 2017-02-26 | 2017-03-03 | Standard Class | DB-13060 | Dave Brooks | Consumer | United States | C |
| **9992** | 9993 | CA-2017-121258 | 2017-02-26 | 2017-03-03 | Standard Class | DB-13060 | Dave Brooks | Consumer | United States | C |
| **9993** | 9994 | CA-2017-119914 | 2017-05-04 | 2017-05-09 | Second Class | CC-12220 | Chris Cortes | Consumer | United States | W |

9994 rows × 21 columns

In [6]:
```
data_original.dtypes
```

Out[6]:
```
Row ID                    int64
Order ID                 object
Order Date       datetime64[ns]
Ship Date        datetime64[ns]
Ship Mode                object
Customer ID              object
Customer Name            object
Segment                  object
Country                  object
City                     object
State                    object
Postal Code               int64
Region                   object
Product ID               object
Category                 object
Sub-Category             object
Product Name             object
Sales                   float64
Quantity                  int64
Discount                float64
Profit                  float64
dtype: object
```

In [7]:
```
data_original.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype         
---  ------         --------------  -----         
 0   Row ID         9994 non-null   int64         
 1   Order ID       9994 non-null   object        
 2   Order Date     9994 non-null   datetime64[ns]
 3   Ship Date      9994 non-null   datetime64[ns]
 4   Ship Mode      9994 non-null   object        
 5   Customer ID    9994 non-null   object        
 6   Customer Name  9994 non-null   object        
 7   Segment        9994 non-null   object        
 8   Country        9994 non-null   object        
 9   City           9994 non-null   object        
 10  State          9994 non-null   object        
 11  Postal Code    9994 non-null   int64         
 12  Region         9994 non-null   object        
 13  Product ID     9994 non-null   object        
 14  Category       9994 non-null   object        
 15  Sub-Category   9994 non-null   object        
 16  Product Name   9994 non-null   object        
 17  Sales          9994 non-null   float64       
 18  Quantity       9994 non-null   int64         
 19  Discount       9994 non-null   float64       
 20  Profit         9994 non-null   float64       
dtypes: datetime64[ns](2), float64(3), int64(3), object(13)
memory usage: 1.6+ MB
```

In [8]:
```
data_original.shape
```

Out[8]:
```
(9994, 21)
```

In [9]:
```
data_original.describe()
```

Out[9]:

|  | Row ID | Postal Code | Sales | Quantity | Discount | Profit |
|---|---|---|---|---|---|---|
| **count** | 9994.000000 | 9994.000000 | 9994.000000 | 9994.000000 | 9994.000000 | 9994.000000 |
| **mean** | 4997.500000 | 55190.379428 | 229.858001 | 3.789574 | 0.156203 | 28.656896 |
| **std** | 2885.163629 | 32063.693350 | 623.245101 | 2.225110 | 0.206452 | 234.260108 |
| **min** | 1.000000 | 1040.000000 | 0.444000 | 1.000000 | 0.000000 | -6599.978000 |
| **25%** | 2499.250000 | 23223.000000 | 17.280000 | 2.000000 | 0.000000 | 1.728750 |
| **50%** | 4997.500000 | 56430.500000 | 54.490000 | 3.000000 | 0.200000 | 8.666500 |
| **75%** | 7495.750000 | 90008.000000 | 209.940000 | 5.000000 | 0.200000 | 29.364000 |
| **max** | 9994.000000 | 99301.000000 | 22638.480000 | 14.000000 | 0.800000 | 8399.976000 |

In [10]:
```
data_original.Category.value_counts()
```

Out[10]:
```
Office Supplies    6026
Furniture          2121
Technology         1847
Name: Category, dtype: int64
```

In [5]:
```
Tech=data_original.loc[data_original['Category']=='Technology']
Tech
```

Out[5]:

| | Row ID | Order ID | Order Date | Ship Date | Ship Mode | Customer ID | Customer Name | Segment | Country | |
|---|---|---|---|---|---|---|---|---|---|---|
| **7** | 8 | CA-2014-115812 | 2014-06-09 | 2014-06-14 | Standard Class | BH-11710 | Brosina Hoffman | Consumer | United States | An |
| **11** | 12 | CA-2014-115812 | 2014-06-09 | 2014-06-14 | Standard Class | BH-11710 | Brosina Hoffman | Consumer | United States | An |
| **19** | 20 | CA-2014-143336 | 2014-08-27 | 2014-09-01 | Second Class | ZD-21925 | Zuschuss Donatelli | Consumer | United States | Fran |
| **26** | 27 | CA-2016-121755 | 2016-01-16 | 2016-01-20 | Second Class | EH-13945 | Eric Hoffmann | Consumer | United States | An |
| **35** | 36 | CA-2016-117590 | 2016-12-08 | 2016-12-10 | First Class | GH-14485 | Gene Hale | Corporate | United States | Richar |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **9983** | 9984 | US-2016-157728 | 2016-09-22 | 2016-09-28 | Standard Class | RC-19960 | Ryan Crowe | Consumer | United States | G Ra |
| **9986** | 9987 | CA-2016-125794 | 2016-09-29 | 2016-10-03 | Standard Class | ML-17410 | Maris LaWare | Consumer | United States | An |
| **9987** | 9988 | CA-2017-163629 | 2017-11-17 | 2017-11-21 | Standard Class | RA-19885 | Ruben Ausman | Corporate | United States | At |
| **9988** | 9989 | CA-2017-163629 | 2017-11-17 | 2017-11-21 | Standard Class | RA-19885 | Ruben Ausman | Corporate | United States | At |
| **9991** | 9992 | CA-2017-121258 | 2017-02-26 | 2017-03-03 | Standard Class | DB-13060 | Dave Brooks | Consumer | United States | C I |

1847 rows × 21 columns

In [14]:
```
Tech['Order Date'].min(), Tech['Order Date'].max()
```

Out[14]:
```
(Timestamp('2014-01-06 00:00:00'), Timestamp('2017-12-30 00:00:00'))
```

In [15]:
```
data_original.columns
```

Out[15]:
```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'Stat
e',
       'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category',
       'Product Name', 'Sales', 'Quantity', 'Discount', 'Profit'],
      dtype='object')
```

In [16]:
```
data_original.isnull().sum()
```

Out[16]:
```
Row ID           0
Order ID         0
Order Date       0
Ship Date        0
Ship Mode        0
Customer ID      0
Customer Name    0
Segment          0
Country          0
City             0
State            0
Postal Code      0
Region           0
Product ID       0
Category         0
Sub-Category     0
Product Name     0
Sales            0
Quantity         0
Discount         0
Profit           0
dtype: int64
```

In [6]:
```
cols=['Row ID',
      'Order ID',
      'Ship Date',
      'Ship Mode',
      'Customer ID',
      'Customer Name',
      'Segment',
      'Country',
      'City',
      'State',
      'Postal Code',
      'Region',
      'Product ID',
      'Category',
      'Sub-Category',
      'Product Name',
      'Quantity',
      'Discount',
      'Profit' ]
Tech.drop(cols,axis=1,inplace=True)
```

```
C:\Users\Sachin sirohi\AppData\Local\Temp\ipykernel_19552\1146970988.py:2
0: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://
pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
view-versus-a-copy)
  Tech.drop(cols,axis=1,inplace=True)
```

In [7]:
```
Tech
```

Out[7]:

| | Order Date | Sales |
|---|---|---|
| **7** | 2014-06-09 | 907.152 |
| **11** | 2014-06-09 | 911.424 |
| **19** | 2014-08-27 | 213.480 |
| **26** | 2016-01-16 | 90.570 |
| **35** | 2016-12-08 | 1097.544 |
| **...** | ... | ... |
| **9983** | 2016-09-22 | 97.980 |
| **9986** | 2016-09-29 | 36.240 |
| **9987** | 2017-11-17 | 79.990 |
| **9988** | 2017-11-17 | 206.100 |
| **9991** | 2017-02-26 | 258.576 |

1847 rows × 2 columns

In [8]:
```
Tech.sort_values('Order Date',inplace=True)
Tech
```

```
C:\Users\Sachin sirohi\AppData\Local\Temp\ipykernel_19552\1398801842.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://
pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
view-versus-a-copy)
  Tech.sort_values('Order Date',inplace=True)
```

Out[8]:

| | Order Date | Sales |
|---|---|---|
| **7478** | 2014-01-06 | 755.960 |
| **7477** | 2014-01-06 | 391.980 |
| **593** | 2014-01-09 | 31.200 |
| **765** | 2014-01-13 | 646.740 |
| **1913** | 2014-01-15 | 149.950 |
| **...** | ... | ... |
| **4924** | 2017-12-25 | 90.480 |
| **2569** | 2017-12-27 | 164.388 |
| **573** | 2017-12-28 | 14.850 |
| **1878** | 2017-12-29 | 302.376 |
| **907** | 2017-12-30 | 90.930 |

1847 rows × 2 columns

```
In [23]: Tech.columns
```

```
Out[23]: Index(['Order Date', 'Sales'], dtype='object')
```

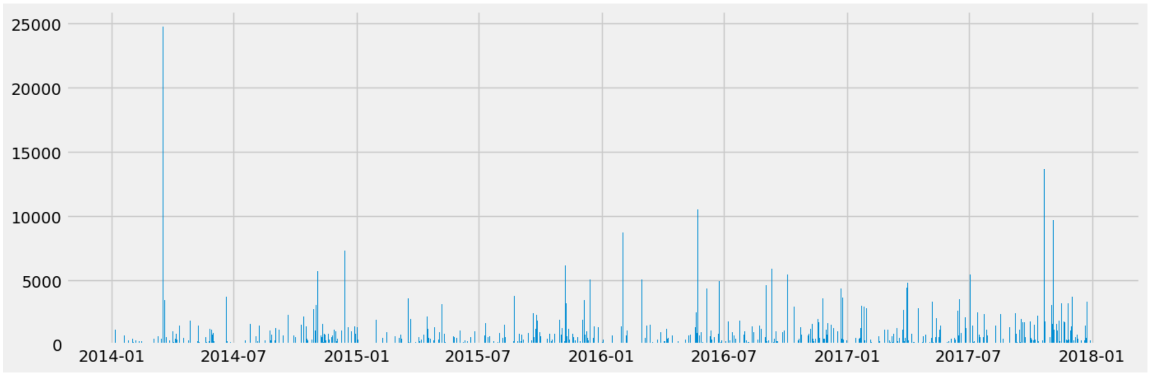
```
In [9]: Tech=Tech.groupby('Order Date').sum().reset_index()
        Tech
```

Out[9]:

|  | Order Date | Sales |
|---|---|---|
| 0 | 2014-01-06 | 1147.940 |
| 1 | 2014-01-09 | 31.200 |
| 2 | 2014-01-13 | 646.740 |
| 3 | 2014-01-15 | 149.950 |
| 4 | 2014-01-16 | 124.200 |
| ... | ... | ... |
| 819 | 2017-12-25 | 401.208 |
| 820 | 2017-12-27 | 164.388 |
| 821 | 2017-12-28 | 14.850 |
| 822 | 2017-12-29 | 302.376 |
| 823 | 2017-12-30 | 90.930 |

824 rows × 2 columns

```
In [34]: plt.bar(Tech['Order Date'],Tech['Sales'])
         plt.show()
```



In [10]:
```
import plotly.express as px
ct=px.bar(x=Tech['Order Date'],y=Tech['Sales'],color=Tech['Sales'])
ct.show()
```

In [11]:
```
Tech=Tech.set_index('Order Date')
Tech.index
```

Out[11]:
```
DatetimeIndex(['2014-01-06', '2014-01-09', '2014-01-13', '2014-01-15',
               '2014-01-16', '2014-01-20', '2014-01-26', '2014-02-01',
               '2014-02-02', '2014-02-06',
               ...
               '2017-12-18', '2017-12-21', '2017-12-22', '2017-12-23',
               '2017-12-24', '2017-12-25', '2017-12-27', '2017-12-28',
               '2017-12-29', '2017-12-30'],
              dtype='datetime64[ns]', name='Order Date', length=824, freq=
None)
```

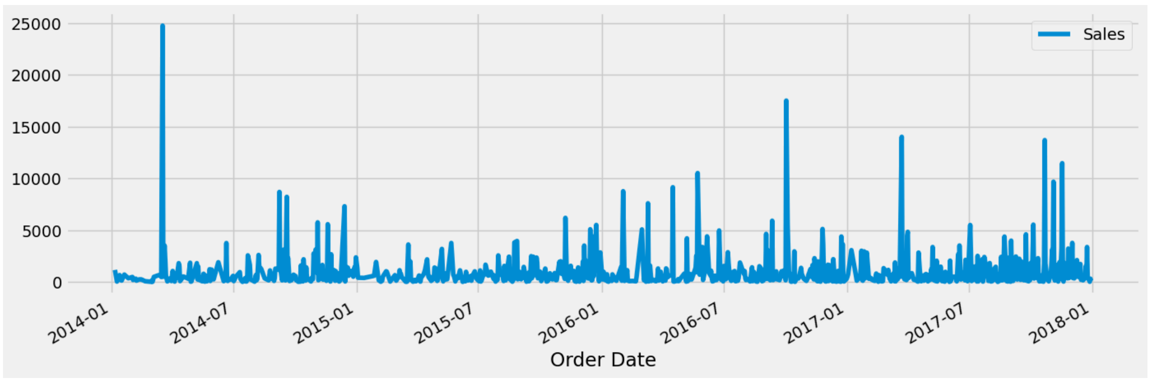
In [42]:
```
Tech.sample(10)
```

Out[42]:

| Order Date | Sales |
|---|---|
| **2016-10-25** | 783.960 |
| **2017-07-24** | 2399.960 |
| **2016-12-16** | 21.210 |
| **2016-01-05** | 250.656 |
| **2014-06-02** | 239.970 |
| **2015-07-16** | 599.900 |
| **2014-11-18** | 549.552 |
| **2014-05-21** | 617.970 |
| **2015-11-30** | 272.400 |
| **2015-09-04** | 754.326 |

In [45]:
```
plt.figure(figsize=(16,6))
Tech.plot()
plt.show()
```

```
<Figure size 1600x600 with 0 Axes>
```



In [46]:
```
pd.DatetimeIndex(Tech.index).year
```

Out[46]:
```
Int64Index([2014, 2014, 2014, 2014, 2014, 2014, 2014, 2014, 2014, 2014,
            ...
            2017, 2017, 2017, 2017, 2017, 2017, 2017, 2017, 2017, 2017],
           dtype='int64', name='Order Date', length=824)
```

In [47]:
```
pd.DatetimeIndex(Tech.index).month
```

Out[47]:
```
Int64Index([ 1,  1,  1,  1,  1,  1,  1,  2,  2,  2,
            ...
            12, 12, 12, 12, 12, 12, 12, 12, 12, 12],
           dtype='int64', name='Order Date', length=824)
```

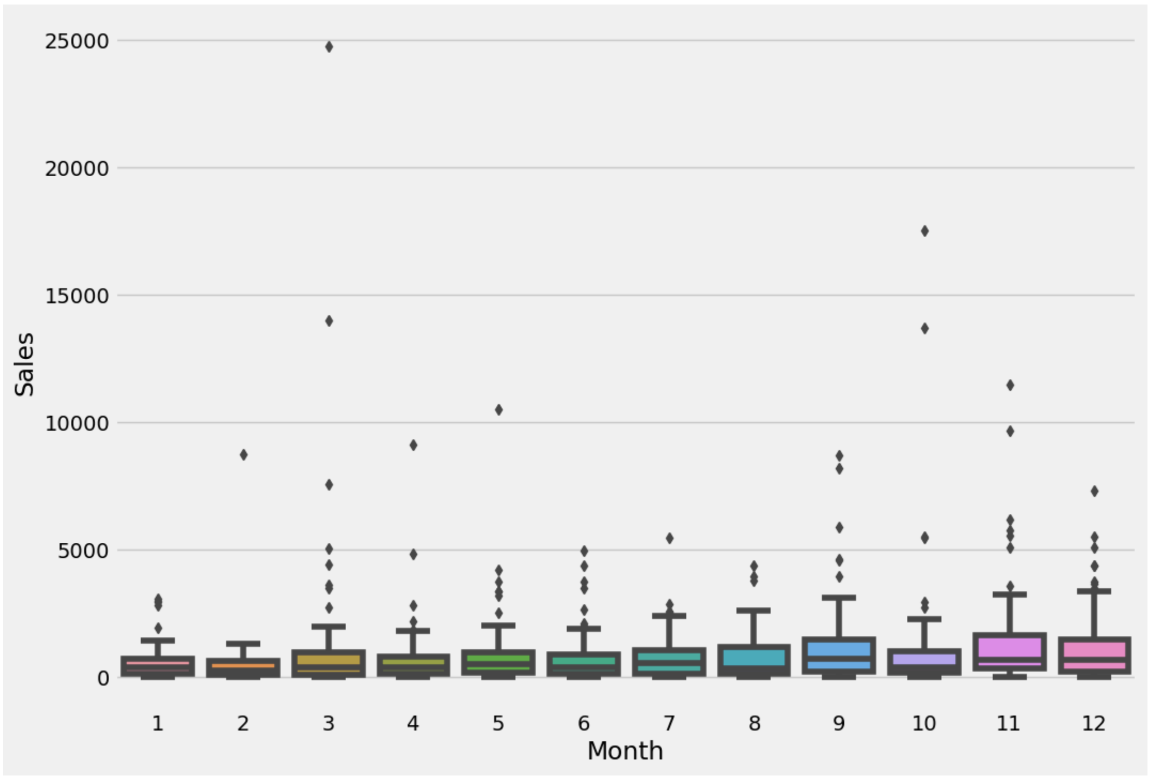
In [12]:
```
Tech['Year']=pd.DatetimeIndex(Tech.index).year
Tech['Month']=pd.DatetimeIndex(Tech.index).month
Tech['Weekday']=pd.DatetimeIndex(Tech.index).day_name()
Tech.sample(10,random_state=0)
```

Out[12]:

| | Sales | Year | Month | Weekday |
|---|---|---|---|---|
| **Order Date** | | | | |
| **2014-07-23** | 2555.084 | 2014 | 7 | Wednesday |
| **2014-03-28** | 302.376 | 2014 | 3 | Friday |
| **2014-02-02** | 180.960 | 2014 | 2 | Sunday |
| **2015-10-20** | 239.970 | 2015 | 10 | Tuesday |
| **2017-12-27** | 164.388 | 2017 | 12 | Wednesday |
| **2017-10-05** | 5520.506 | 2017 | 10 | Thursday |
| **2017-07-21** | 2180.644 | 2017 | 7 | Friday |
| **2015-09-03** | 247.840 | 2015 | 9 | Thursday |
| **2015-12-18** | 166.240 | 2015 | 12 | Friday |
| **2017-03-19** | 957.004 | 2017 | 3 | Sunday |

In [53]:
```
fig,ax=plt.subplots(figsize=(11,8))
sns.boxplot(data=Tech,x='Month',y='Sales',ax=ax)
```

Out[53]:
```
<AxesSubplot:xlabel='Month', ylabel='Sales'>
```



```
In [54]: Tech.loc['2014']
```

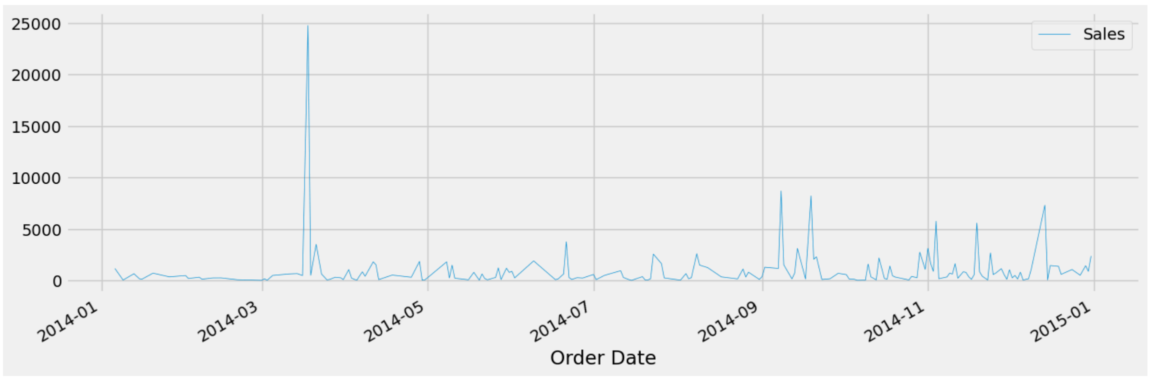
```
Out[54]:
```

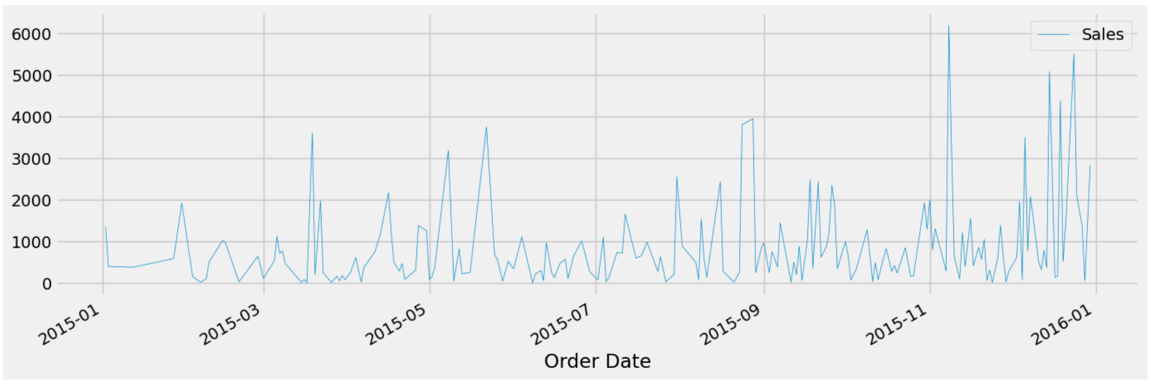
| | Sales | Year | Month | Weekday |
|---|---|---|---|---|
| **Order Date** | | | | |
| **2014-01-06** | 1147.940 | 2014 | 1 | Monday |
| **2014-01-09** | 31.200 | 2014 | 1 | Thursday |
| **2014-01-13** | 646.740 | 2014 | 1 | Monday |
| **2014-01-15** | 149.950 | 2014 | 1 | Wednesday |
| **2014-01-16** | 124.200 | 2014 | 1 | Thursday |
| **...** | ... | ... | ... | ... |
| **2014-12-24** | 1054.882 | 2014 | 12 | Wednesday |
| **2014-12-27** | 498.000 | 2014 | 12 | Saturday |
| **2014-12-29** | 1423.672 | 2014 | 12 | Monday |
| **2014-12-30** | 874.276 | 2014 | 12 | Tuesday |
| **2014-12-31** | 2360.014 | 2014 | 12 | Wednesday |

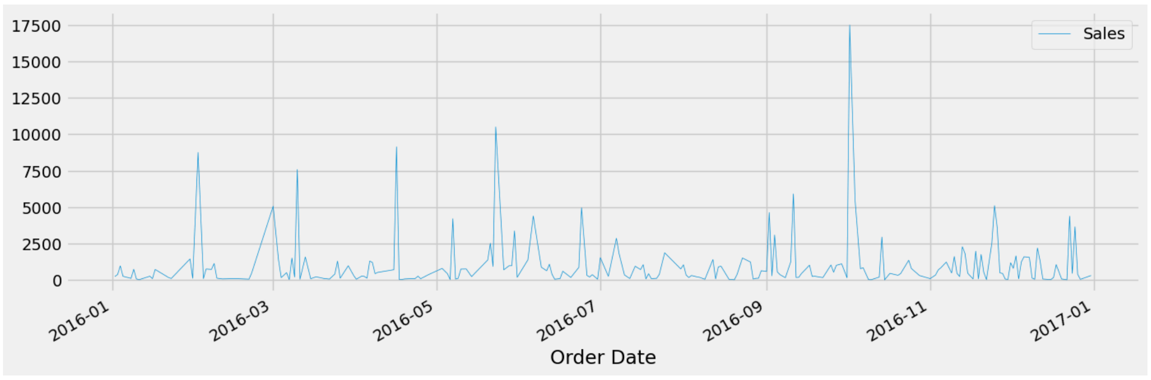
179 rows × 4 columns

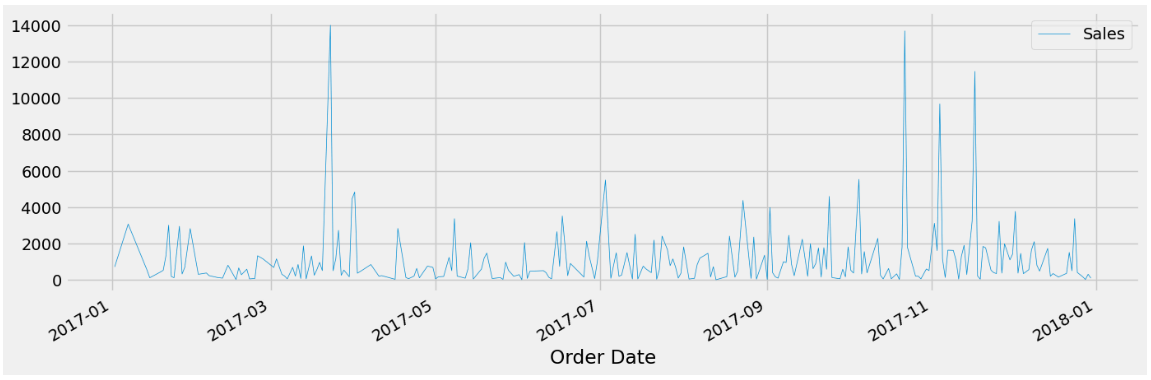
```
In [55]: col_plot=['Sales']
In [58]: Tech.loc['2014'][col_plot].plot(linewidth=0.5)
         Tech.loc['2015'][col_plot].plot(linewidth=0.5)
         Tech.loc['2016'][col_plot].plot(linewidth=0.5)
         Tech.loc['2017'][col_plot].plot(linewidth=0.5)
```

```
Out[58]: <AxesSubplot:xlabel='Order Date'>
```









```
In [15]: Tech_weekly=Tech['Sales'].resample('w').sum()
         Tech_weekly.head()
```

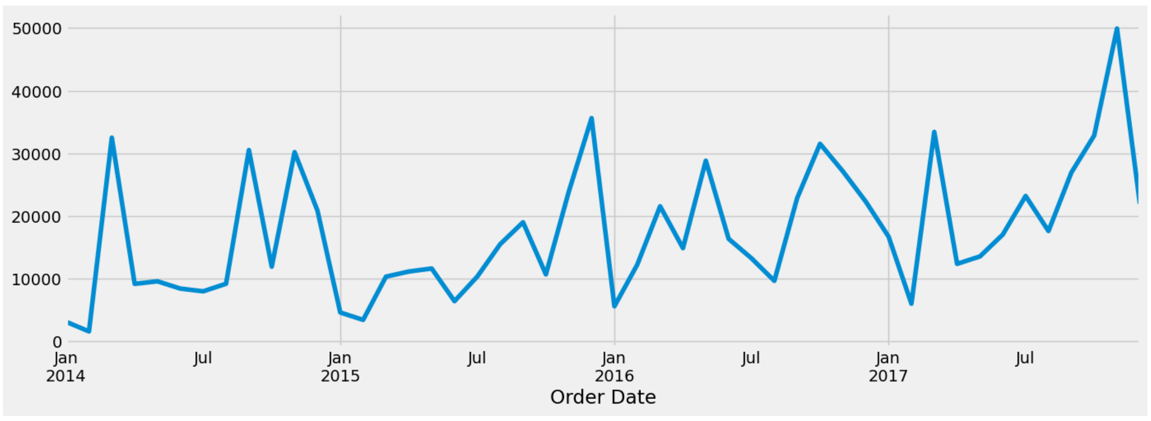
```
Out[15]: Order Date
         2014-01-12    1179.14
         2014-01-19     920.89
         2014-01-26    1043.26
         2014-02-02     649.86
         2014-02-09     421.92
         Freq: W-SUN, Name: Sales, dtype: float64
```

```
In [16]: Tech_Monthly=Tech['Sales'].resample('M').sum()
         Tech_Monthly.head()
```

```
Out[16]: Order Date
         2014-01-31     3143.290
         2014-02-28     1608.510
         2014-03-31    32511.174
         2014-04-30     9195.434
         2014-05-31     9599.876
         Freq: M, Name: Sales, dtype: float64
```

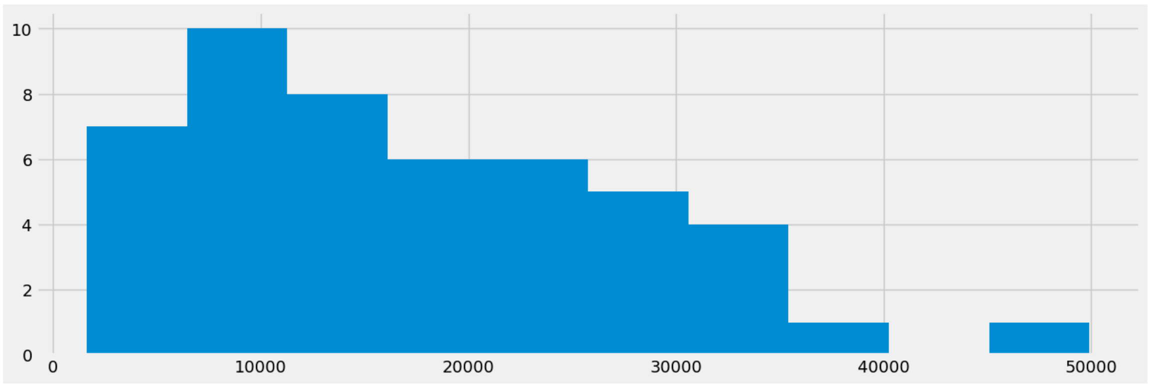
```
In [18]: Tech_Monthly.plot()
```

```
Out[18]: <AxesSubplot:xlabel='Order Date'>
```



```
In [19]: Tech_Monthly.hist()
```

```
Out[19]: <AxesSubplot:>
```


10
8
6
4
2
0
0
10000
20000
30000
40000
50000