comparetime

May 7, 2025

[1]:
```
import pandas as pd
import seaborn as sns
```

[2]:
```
df=pd.read_csv("storedataset.csv")
```

[3]:
```
df.head(2)
```

[3]:
```
   Row ID        Order ID  Order Date   Ship Date     Ship Mode Customer ID  \
0       1  CA-2017-152156  08/11/2017  11/11/2017  Second Class    CG-12520
1       2  CA-2017-152156  08/11/2017  11/11/2017  Second Class    CG-12520

  Customer Name   Segment        Country       City     State  Postal Code  \
0   Claire Gute  Consumer  United States  Henderson  Kentucky      42420.0
1   Claire Gute  Consumer  United States  Henderson  Kentucky      42420.0

  Region       Product ID   Category Sub-Category  \
0  South  FUR-BO-10001798  Furniture    Bookcases
1  South  FUR-CH-10000454  Furniture       Chairs

                                        Product Name   Sales
0                  Bush Somerset Collection Bookcase  261.96
1  Hon Deluxe Fabric Upholstered Stacking Chairs,…  731.94
```

[4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9800 entries, 0 to 9799
Data columns (total 18 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9800 non-null   int64
 1   Order ID       9800 non-null   object
 2   Order Date     9800 non-null   object
 3   Ship Date      9800 non-null   object
 4   Ship Mode      9800 non-null   object
 5   Customer ID    9800 non-null   object
 6   Customer Name  9800 non-null   object
 7   Segment        9800 non-null   object
 8   Country        9800 non-null   object
 9   City           9800 non-null   object
 10  State          9800 non-null   object
 11  Postal Code    9789 non-null   float64
 12  Region         9800 non-null   object
 13  Product ID     9800 non-null   object
 14  Category       9800 non-null   object
 15  Sub-Category   9800 non-null   object
 16  Product Name   9800 non-null   object
 17  Sales          9800 non-null   float64
dtypes: float64(2), int64(1), object(15)
memory usage: 1.3+ MB
```

[5]:

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

[5]:

```
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      11
Region            0
Product ID        0
Category          0
Sub-Category      0
Product Name      0
Sales             0
dtype: int64
```

Compare order time with shipping time(What is the shipping delay?)

[6]:

```
df['Order Date']=pd.to_datetime(df['Order Date'], dayfirst=True)
df['Ship Date']=pd.to_datetime(df['Ship Date'], dayfirst=True)
```

[7]:

```
df['Delay']=df['Ship Date']-df['Order Date']
df['Delay']
```

[7]:

```
0       3 days
1       3 days
2       4 days
3       7 days
4       7 days
```

```
            …
9795   7 days
9796   5 days
9797   5 days
9798   5 days
9799   5 days
Name: Delay, Length: 9800, dtype: timedelta64[ns]
```

The Delay in the dataframe was a timedelta (i.e. 3 days 00:00:00).

We only need the number of days, not the hours and minutes.

With .dt.days we just split the days and put them in a new column called Delay_days.

[8]:

```
df['Delay_days'] = df['Delay'].dt.days
```

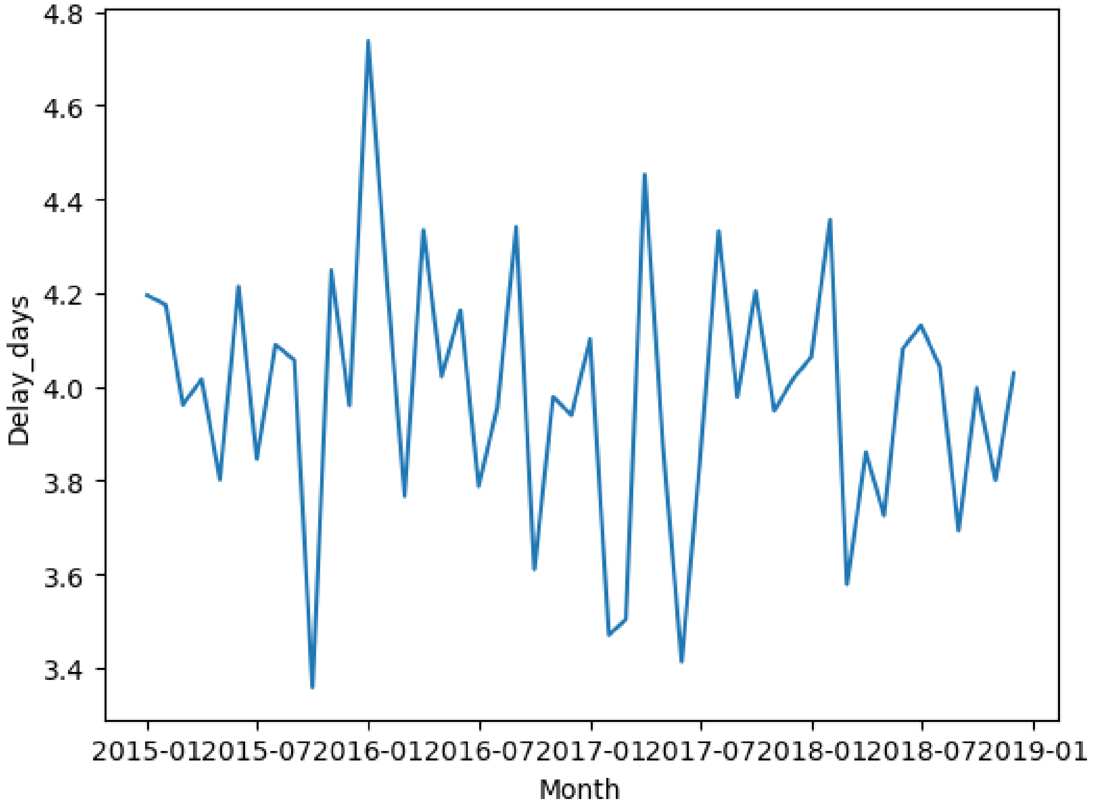
What was the average delivery delay per month and how has it changed? Because it gets too busy to chart daily, we changed the order date to "month". Here we calculated the average delay per month. Now we only have one number for each month (average delay). Since the month was a period type, we converted it to a Timestamp so we could plot it.

[9]:

```
df['Month'] = df['Order Date'].dt.to_period('M')
df_grouped = df.groupby('Month')['Delay_days'].mean().reset_index()
df_grouped['Month'] = df_grouped['Month'].dt.to_timestamp()
sns.lineplot(data=df_grouped, x='Month', y='Delay_days')
```

[9]:

```
<Axes: xlabel='Month', ylabel='Delay_days'>
```


4.8
4.6
4.4
4.2
4.0
3.8
3.6
3.4
Delay_days
2015-01
2015-07
2016-01
2016-07
2017-01
2017-07
2018-01
2018-07
2019-01
Month