```python
import pandas as pd
df=pd.read_csv("D:\\Data_Science_Intern\\Amazon Sales data.csv")
```

```python
df.head()
```

```
                              Region                Country
Item Type  \
0               Australia and Oceania                 Tuvalu
Baby Food
1  Central America and the Caribbean                Grenada
Cereal
2                              Europe                 Russia  Office
Supplies
3                  Sub-Saharan Africa  Sao Tome and Principe
Fruits
4                  Sub-Saharan Africa                 Rwanda  Office
Supplies

  Sales Channel Order Priority Order Date   Order ID  Ship Date  Units
Sold  \
0       Offline              H  5/28/2010  669165933  6/27/2010
9925
1        Online              C  8/22/2012  963881480  9/15/2012
2804
2       Offline              L   5/2/2014  341417157   5/8/2014
1779
3        Online              C  6/20/2014  514321792   7/5/2014
8102
4       Offline              L   2/1/2013  115456712   2/6/2013
5062

   Unit Price  Unit Cost  Total Revenue  Total Cost  Total Profit
0      255.28     159.42     2533654.00  1582243.50     951410.50
1      205.70     117.11      576782.80   328376.44     248406.36
2      651.21     524.96     1158502.59   933903.84     224598.75
3        9.33       6.92       75591.66    56065.84      19525.82
4      651.21     524.96     3296425.02  2657347.52     639077.50
```

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

```
Region            False
Country           False
Item Type         False
Sales Channel     False
Order Priority    False
Order Date        False
Order ID          False
Ship Date         False
Units Sold        False
Unit Price        False
Unit Cost         False
Total Revenue     False
Total Cost        False
Total Profit      False
dtype: bool
```

```
df['Order Date']=pd.to_datetime(df['Order Date'],errors='coerce')
```

```
#Extract month and year from Order Date
df['Year']=df['Order Date'].dt.year
df['Month']=df['Order Date'].dt.month
```

```
#Aggregate data for month-wise sales
month_wise_sales=df.groupby('Month')['Total
Revenue'].sum().reset_index()
```

```
print(month_wise_sales)
```

```
    Month  Total Revenue
0       1    10482467.12
1       2    24740517.77
2       3     2274823.87
3       4    16187186.33
4       5    13215739.99
5       6     5230325.77
6       7    15669518.50
7       8     1128164.91
8       9     5314762.56
9      10    15287576.61
10     11    20568222.76
11     12     7249462.12
```

```
#Aggregate data for year-wise sales
year_wise_sales=df.groupby('Year')['Total
Revenue'].sum().reset_index()
```

```
year_wise_sales
```

```
   Year  Total Revenue
0  2010    19186024.92
1  2011    11129166.07
2  2012    31898644.52
3  2013    20330448.66
4  2014    16630214.43
5  2015    12427982.86
6  2016    12372867.22
7  2017    13373419.63
```

```
#Aggregate data for year_month-wise sales
year_month_wise_sales=df.groupby(['Year','Month'])['Total
Revenue'].sum().reset_index()
```

```
year_month_wise_sales
```

```
    Year  Month  Total Revenue
0   2010      2     3410661.12
1   2010      5     2587973.26
2   2010      6     1082418.40
3   2010     10     6064933.75
4   2010     11     3458252.00
5   2010     12     2581786.39
6   2011      1     1042225.35
7   2011      2      387002.20
8   2011      4     2798046.49
9   2011      5      272410.45
10  2011      6       19103.44
11  2011      7       97040.64
12  2011      9      574951.92
13  2011     11     5938385.58
14  2012      1     1012884.00
15  2012      2     6707849.42
16  2012      3      994765.42
17  2012      4     4556012.38
18  2012      5     3782781.82
19  2012      6     2132075.27
20  2012      7     4445093.92
21  2012      8      576782.80
22  2012      9     4648152.72
23  2012     10     3042246.77
24  2013      2     3296425.02
25  2013      3      835759.10
26  2013      4     3262562.10
27  2013      6     1352867.40
28  2013      7     8545511.20
29  2013      8       89623.98
30  2013      9       71253.21
31  2013     10     2702770.40
32  2013     12      173676.25
33  2014      2     1819660.25
34  2014      4     4510578.10
35  2014      5     3060338.59
36  2014      6       75591.66
37  2014      7      688641.85
38  2014      8      455479.04
39  2014      9       20404.71
40  2014     10     1352370.65
41  2014     11     4647149.58
42  2015      1     5513227.50
43  2015      2     2003911.12
44  2015      4     1059987.26
45  2015      7     1292409.45
46  2015      8        6279.09
```

```
47  2015     10     1904138.04
48  2015     11      648030.40
49  2016      3      197883.40
50  2016      5      414371.10
51  2016      6      568269.60
52  2016      7      600821.44
53  2016     10      221117.00
54  2016     11     5876405.20
55  2016     12     4493999.48
56  2017      1     2914130.27
57  2017      2     7115008.64
58  2017      3      246415.95
59  2017      5     3097864.77
```

```
new_df = 'Amazon_sales_transformed_data.xlsx'
# Create a Pandas ExcelWriter using openpyxl engine
with pd.ExcelWriter(new_df, engine='openpyxl') as writer:

    df.to_excel(writer, sheet_name='Original Data', index=False)
#oroginal data


    month_wise_sales.to_excel(writer, sheet_name='Month-wise Sales',
index=False)   #month-wise sales data

    year_wise_sales.to_excel(writer, sheet_name='Year-wise Sales',
index=False)     # year-wise sales data

    year_month_wise_sales.to_excel(writer, sheet_name='Year_Month-wise
Sales', index=False)        # year-month-wise sales data

print(f"Transformed data saved to {new_df}")
```

```
Transformed data saved to Amazon_sales_transformed_data.xlsx
```