

# Kaggle\_Ecommerce\_Sale

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
In [2]: df = pd.read_csv('C:\Data Analytics\Python Project\Kaggle Sale\sales_data.csv')
df.head()
```

```
Out[2]:
```

	Order Date	Order ID	Product	Product_ean	catégorie	Purchase Address	Quantity Ordered	Price Each
0	22-01-2019 21:25	141234	iPhone	5.638010e+12	Vêtements	944 Walnut St, Boston, MA 02215	1	700.00
1	28-01-2019 14:15	141235	Lightning Charging Cable	5.563320e+12	Alimentation	185 Maple St, Portland, OR 97035	1	14.95
2	17-01-2019 13:33	141236	Wired Headphones	2.113970e+12	Vêtements	538 Adams St, San Francisco, CA 94016	2	11.99
3	05-01-2019 20:33	141237	27in FHD Monitor	3.069160e+12	Sports	738 10th St, Los Angeles, CA 90001	1	149.99
4	25-01-2019 11:59	141238	Wired Headphones	9.692680e+12	Électronique	387 10th St, Austin, TX 73301	1	11.99

```
In [3]: df.isnull().sum()
```

```
Out[3]: Order Date      0
        Order ID       0
        Product        0
        Product_ean     0
        catégorie      0
        Purchase Address 0
        Quantity Ordered 0
        Price Each      0
        Cost price      0
        turnover        0
        margin          0
        dtype: int64
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 185950 entries, 0 to 185949
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Order Date            185950 non-null object
1   Order ID              185950 non-null int64
2   Product               185950 non-null object
3   Product_ean           185950 non-null float64
4   catégorie             185950 non-null object
5   Purchase Address      185950 non-null object
6   Quantity Ordered      185950 non-null int64
7   Price Each            185950 non-null float64
8   Cost price            185950 non-null float64
9   turnover              185950 non-null float64
10  margin                185950 non-null float64
dtypes: float64(5), int64(2), object(4)
memory usage: 15.6+ MB
```

```
In [5]: df.columns
```

```
Out[5]: Index(['Order Date', 'Order ID', 'Product', 'Product_ean', 'catégorie',
              'Purchase Address', 'Quantity Ordered', 'Price Each', 'Cost price',
              'turnover', 'margin'],
              dtype='object')
```

```
In [6]: df['Order Year'] = df['Order Date'].str.split(' ').str[0].str.split('-').str[2]
```

```
In [7]: df['Order Month'] = df['Order Date'].str.split(' ').str[0].str.split('-').str[1]
```

```
In [8]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 185950 entries, 0 to 185949
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Order Date            185950 non-null object
1   Order ID              185950 non-null int64
2   Product               185950 non-null object
3   Product_ean           185950 non-null float64
4   catégorie             185950 non-null object
5   Purchase Address      185950 non-null object
6   Quantity Ordered      185950 non-null int64
7   Price Each            185950 non-null float64
8   Cost price            185950 non-null float64
9   turnover              185950 non-null float64
10  margin                185950 non-null float64
11  Order Year            185950 non-null object
12  Order Month           185950 non-null object
dtypes: float64(5), int64(2), object(6)
memory usage: 18.4+ MB
```

```
In [9]: df['Order Year'] = df['Order Year'].astype(int)
```

```
In [10]: df['Order Month'] = df['Order Month'].astype(int)
```

```
In [11]: month_mapping = {
    1: 'Jan',
    2: 'Feb',
    3: 'Mar',
    4: 'Apr',
    5: 'May',
    6: 'Jun',
    7: 'Jul',
    8: 'Aug',
    9: 'Sep',
    10: 'Oct',
    11: 'Nov',
    12: 'Dec'
}
```

```
In [12]: df['Order Month'].replace(month_mapping,inplace = True)
```

```
In [13]: df['Time'] = df['Order Date'].str.split(' ').str[1].str.split(":").str[0]
```

```
In [14]: df['Time'] = df['Time'].astype(int)
df['Time_Zone'] = df['Time']
```

```
In [15]: df_copy = df
```

```
In [16]: def Time_Changes(x):
    if(x<=5):
        return 'Night'
    elif(x<=12):
        return "Morning"
    elif(x<=16):
        return "After_noon"
    elif(x<=21):
        return 'Evening'
```

```
else:  
    return "Night"
```

```
In [17]: df['Time_Zone'] = df['Time_Zone'].apply(Time_Changes)
```

```
In [18]: df.columns
```

```
Out[18]: Index(['Order Date', 'Order ID', 'Product', 'Product_ean', 'catégorie',  
               'Purchase Address', 'Quantity Ordered', 'Price Each', 'Cost price',  
               'turnover', 'margin', 'Order Year', 'Order Month', 'Time', 'Time_Zone'],  
              dtype='object')
```

```
In [19]: df['Category'] = df['catégorie']
```

```
In [20]: df.drop(['catégorie', 'Product_ean'], axis = 1, inplace = True)
```

```
In [21]: def category_changes(x):  
         if(x == 'Vêtements'):  
             return 'Clothes'  
         elif(x=='Alimentation'):  
             return 'Food'  
         elif(x=='Électronique'):  
             return 'Electronic'  
         else:  
             return 'Sports'
```

```
In [22]: df['Category'] = df['Category'].apply(category_changes)
```

```
In [23]: df['Purchase City'] = df['Purchase Address'].str.split(',').str[1]
```

```
In [24]: df.drop(['Purchase Address'], axis = 1, inplace = True)
```

```
In [25]: df['Order Year'].unique()
```

```
Out[25]: array([2019, 2020])
```

```
In [26]: df['Order Year'].value_counts()
```

```
Out[26]: Order Year  
2019    185916  
2020         34  
Name: count, dtype: int64
```

```
In [27]: df = df[df['Order Year'] == 2019]
```

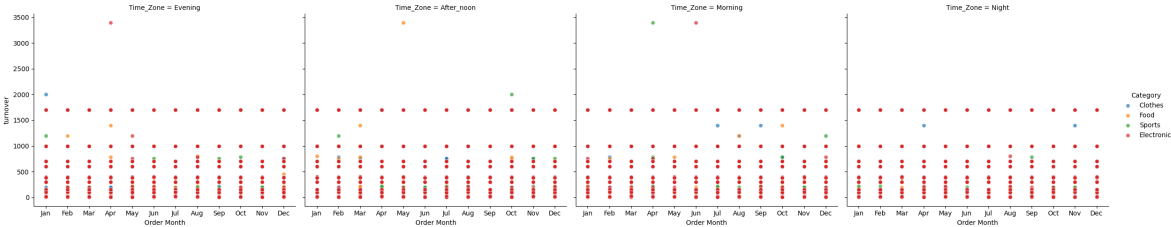
```
In [28]: df.head()
```

Out[28]:

	Order Date	Order ID	Product	Quantity Ordered	Price Each	Cost price	turnover	margin	Order Year
0	22-01-2019 21:25	141234	iPhone	1	700.00	231.0000	700.00	469.0000	2019
1	28-01-2019 14:15	141235	Lightning Charging Cable	1	14.95	7.4750	14.95	7.4750	2019
2	17-01-2019 13:33	141236	Wired Headphones	2	11.99	5.9950	23.98	11.9900	2019
3	05-01-2019 20:33	141237	27in FHD Monitor	1	149.99	97.4935	149.99	52.4965	2019
4	25-01-2019 11:59	141238	Wired Headphones	1	11.99	5.9950	11.99	5.9950	2019

In [29]:

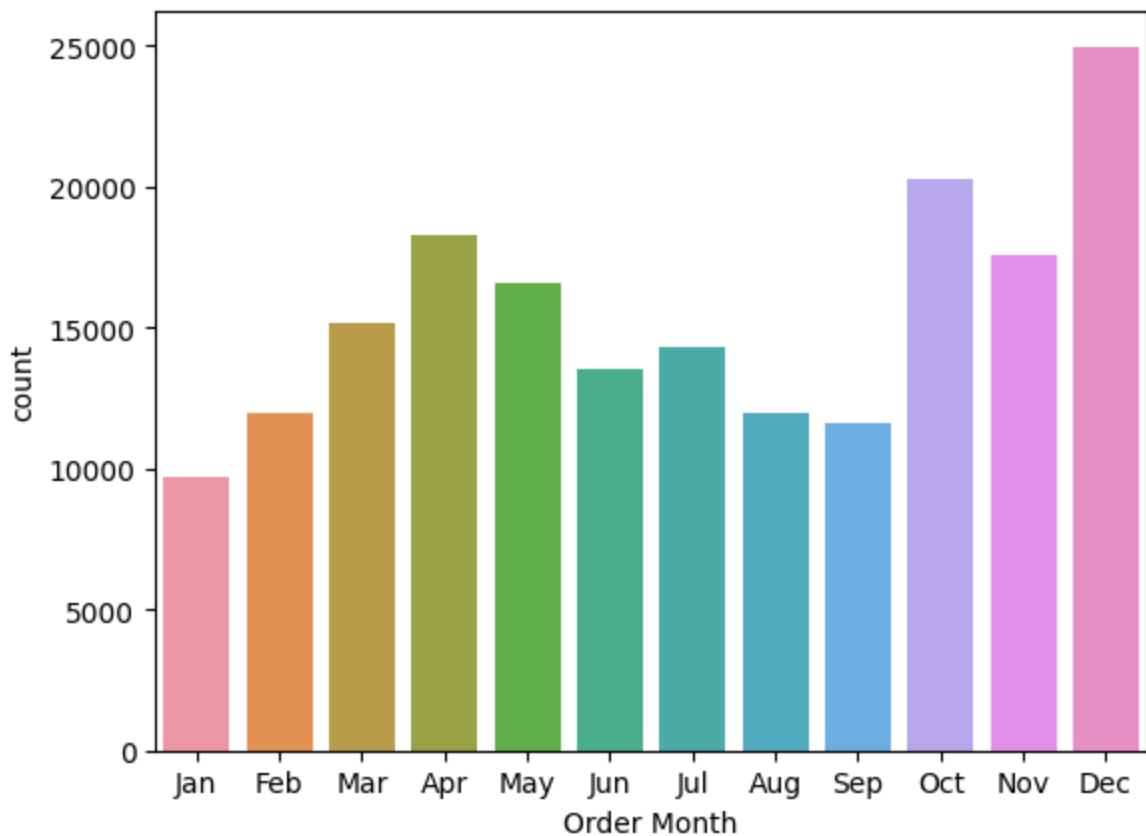
```
graph = sns.FacetGrid(df, col="Time_Zone", hue="Category", height=5, aspect=1.2)
graph.map(sns.scatterplot, "Order Month", "turnover", alpha=0.7)
graph.add_legend()
plt.show()
```



In [30]:

```
sns.countplot(x='Order Month', data=df)
```

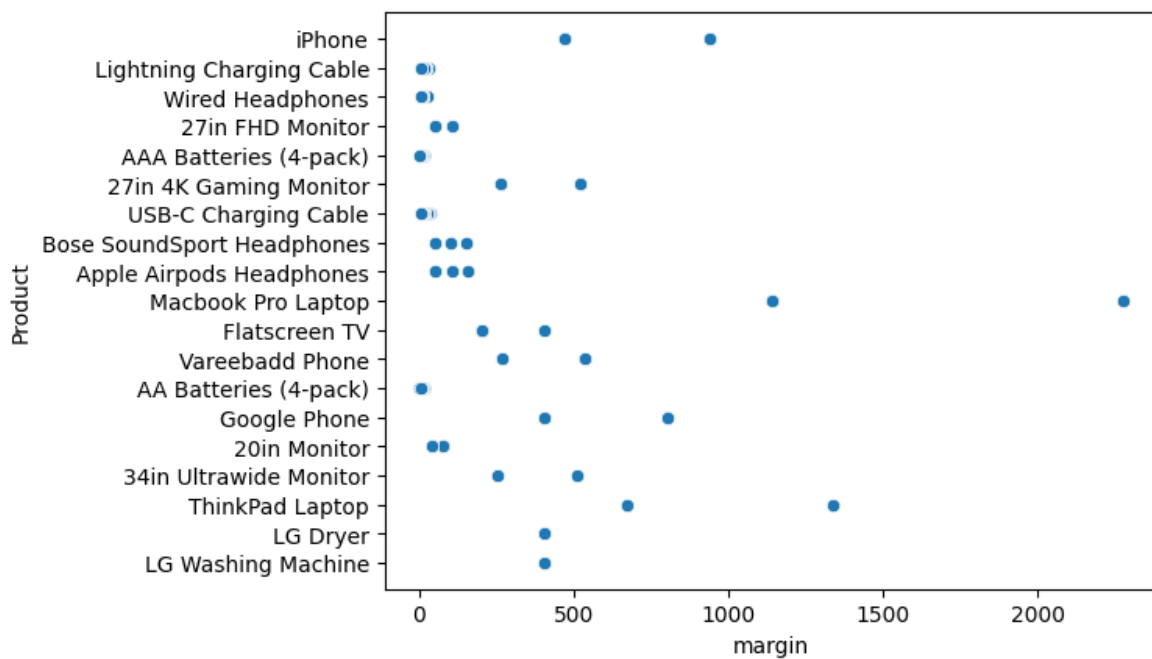
Out[30]: <Axes: xlabel='Order Month', ylabel='count'>



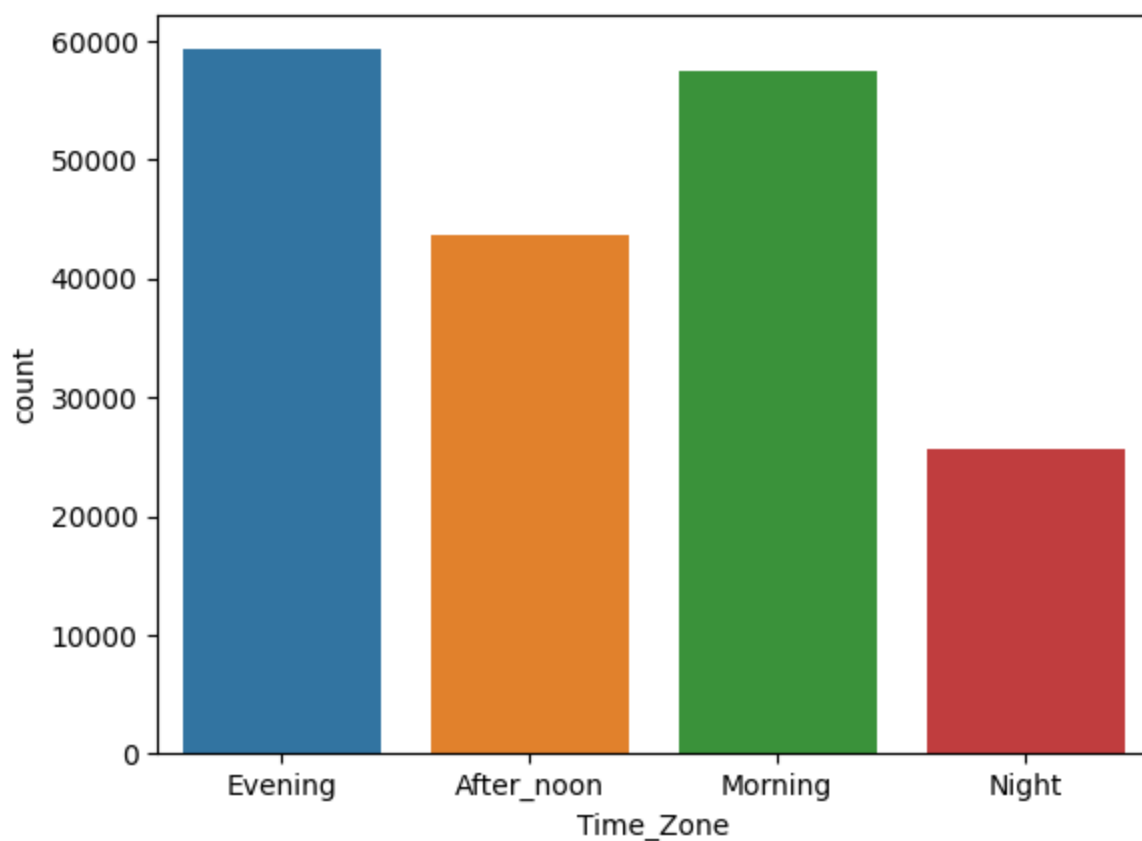
```
In [31]: #we make a def funtion to minimize our unique values to Lower
def change(x):
    if x in ['USB-C Charging Cable','Lightning Charging Cable']:
        return 'Charging Cables'
    elif x in ['AAA Batteries (4-pack)','AA Batteries (4-pack)']:
        return 'Batteries'
    elif x in ['Wired Headphones','Apple Airpods Headphones','Bose SoundSport He
        return 'Headphones'
    elif x in ['27in FHD Monitor','27in 4K Gaming Monitor','34in Ultrawide Monit
        return 'Smart Tv'
    elif x in ['iPhone','Google Phone','Vareebadd Phone']:
        return 'Smart Phones'
    elif x in ['Macbook Pro Laptop','ThinkPad Laptop']:
        return 'Laptops'
    elif x in ['LG Washing Machine','LG Dryer']:
        return 'Cleaning Machines'
    else:
        return 'Others'
```

```
In [32]: df['Sub_Product'] = df['Product'].apply(change)
```

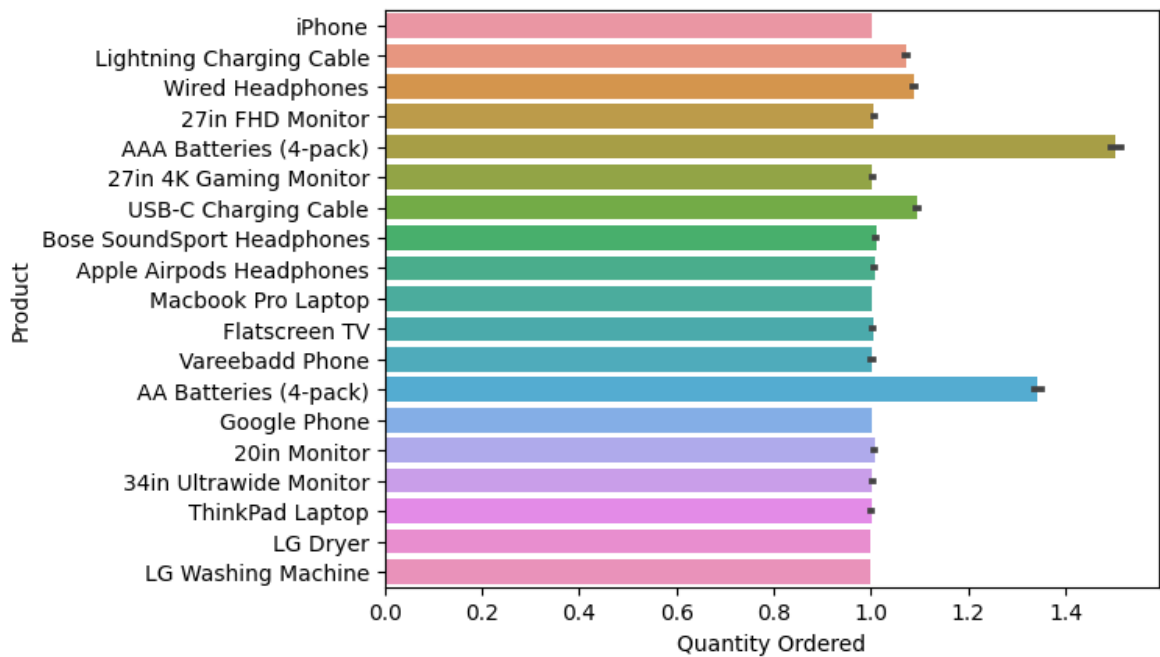
```
In [33]: sns.scatterplot(y="Product", x="margin",data=df)
plt.show()
```



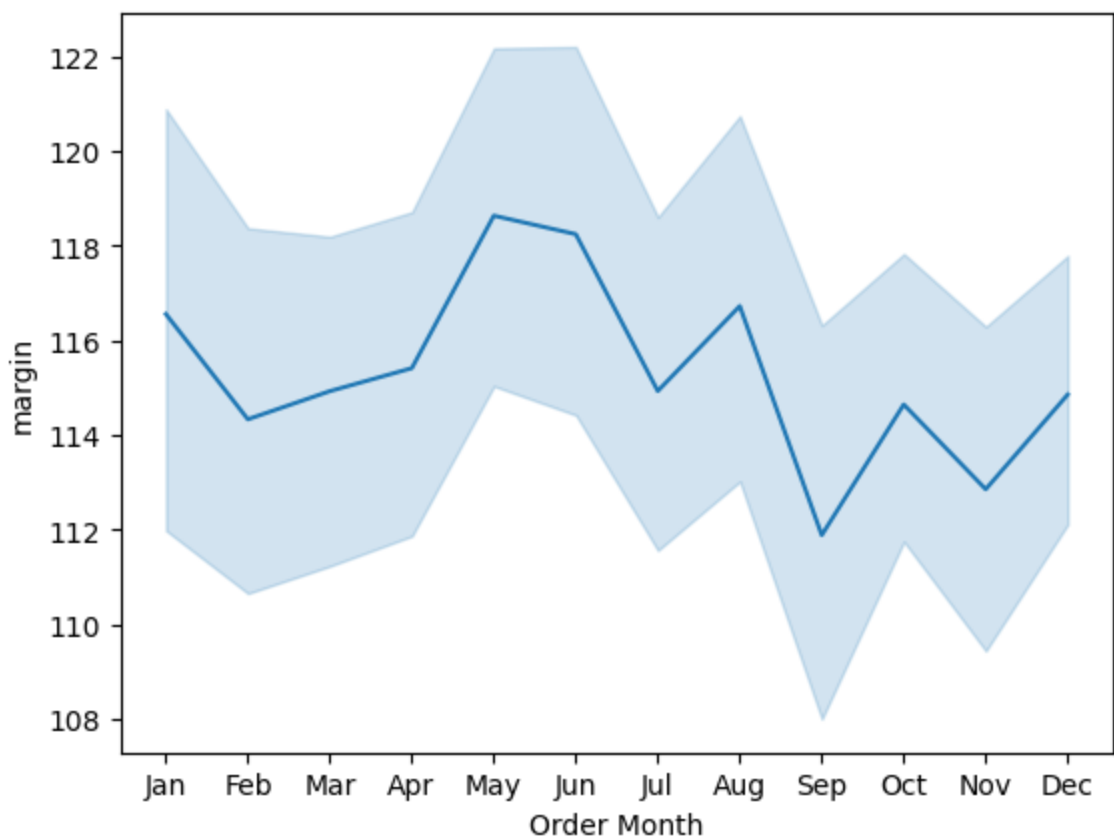
```
In [34]: sns.countplot(x = 'Time_Zone', data = df)
plt.show()
```



```
In [35]: sns.barplot(y = 'Product', x = 'Quantity Ordered', data = df )
plt.show()
```

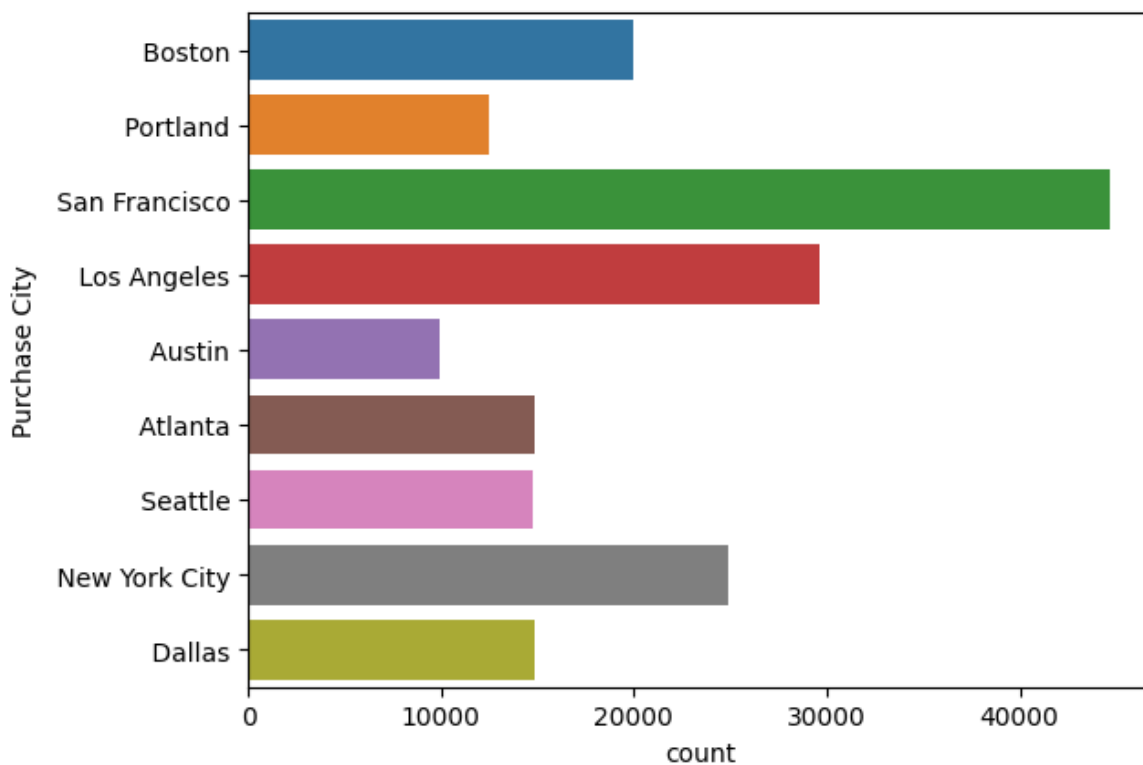


```
In [36]: sns.lineplot(x = 'Order Month', y = 'margin', data = df)
plt.show()
```



```
In [37]: sns.countplot(y= 'Purchase City', data = df)
plt.show()
```





```
In [38]: df.drop(['Time'], axis=1, inplace=True)
```

```
In [39]: # df.to_excel("output.xlsx", index=False)
```

```
In [40]: df.head()
```

```
Out[40]:
```

	Order Date	Order ID	Product	Quantity Ordered	Price Each	Cost price	turnover	margin	Order Year
0	22-01-2019 21:25	141234	iPhone	1	700.00	231.0000	700.00	469.0000	2019
1	28-01-2019 14:15	141235	Lightning Charging Cable	1	14.95	7.4750	14.95	7.4750	2019
2	17-01-2019 13:33	141236	Wired Headphones	2	11.99	5.9950	23.98	11.9900	2019
3	05-01-2019 20:33	141237	27in FHD Monitor	1	149.99	97.4935	149.99	52.4965	2019
4	25-01-2019 11:59	141238	Wired Headphones	1	11.99	5.9950	11.99	5.9950	2019

