

```
In [1]: import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objs as go
from plotly.offline import iplot
```

```
In [2]: import pandas as pd
```

```
In [3]: data = pd.read_csv("sales_data.csv")
```

```
In [4]: data.dtypes
```

```
Out[4]: Date                object
Day                int64
Month              object
Year              int64
Customer_Age       int64
Age_Group          object
Customer_Gender    object
Country            object
State              object
Product_Category   object
Sub_Category       object
Product            object
Order_Quantity     int64
Unit_Cost          int64
Unit_Price         int64
Profit             int64
Cost               int64
Revenue            int64
dtype: object
```

```
In [5]: data.Unit_Cost.astype(object)
```

```
Out[5]: 0          45
1          45
2          45
3          45
4          45
..
113031     24
113032     24
113033     24
113034     24
113035     24
Name: Unit_Cost, Length: 113036, dtype: object
```

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

```
Out[6]: Date                object
        Day                 int64
        Month               object
        Year                int64
        Customer_Age        int64
        Age_Group           object
        Customer_Gender     object
        Country             object
        State               object
        Product_Category    object
        Sub_Category        object
        Product             object
        Order_Quantity      int64
        Unit_Cost           int64
        Unit_Price          int64
        Profit              int64
        Cost                int64
        Revenue             int64
        dtype: object
```

```
In [7]: data.isnull().sum()
```

```
Out[7]: Date                0
        Day                 0
        Month               0
        Year                0
        Customer_Age        0
        Age_Group           0
        Customer_Gender     0
        Country             0
        State               0
        Product_Category    0
        Sub_Category        0
        Product             0
        Order_Quantity      0
        Unit_Cost           0
        Unit_Price          0
        Profit              0
        Cost                0
        Revenue             0
        dtype: int64
```

REVENUE AND MONTHS

```
In [8]: '2013-11-26'.split('-')[1]
```

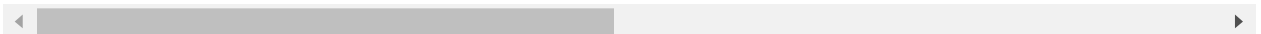
```
Out[8]: '11'
```

```
In [9]: def Month(x):
        return x.split('-')[1]
```

```
In [10]: data.head()
```

```
Out[10]:
```

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Country	State
0	2013-11-26	26	November	2013	19	Youth (<25)	M	Canada	British Columbia
1	2015-11-26	26	November	2015	19	Youth (<25)	M	Canada	British Columbia
2	2014-03-23	23	March	2014	49	Adults (35-64)	M	Australia	New South Wales
3	2016-03-23	23	March	2016	49	Adults (35-64)	M	Australia	New South Wales
4	2014-05-15	15	May	2014	47	Adults (35-64)	F	Australia	New South Wales



```
In [11]: data['Month']=data['Date'].apply(Month)
```

```
In [12]: data.dtypes
```

```
Out[12]: Date                object
Day                int64
Month              object
Year              int64
Customer_Age      int64
Age_Group         object
Customer_Gender   object
Country           object
State             object
Product_Category  object
Sub_Category      object
Product           object
Order_Quantity    int64
Unit_Cost         int64
Unit_Price        int64
Profit            int64
Cost              int64
Revenue           int64
dtype: object
```

```
In [13]: data['Month'].unique()
```

```
Out[13]: array(['11', '03', '05', '02', '07', '08', '09', '01', '12', '06', '10',
                '04'], dtype=object)
```

```
In [14]: filter=data['Month']=='Date'  
len(data[~filter])
```

Out[14]: 113036

```
In [15]: data=data[~filter]
```

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

Out[16]: (113036, 18)

```
In [17]: data['Month']=data['Month'].apply(int)
```

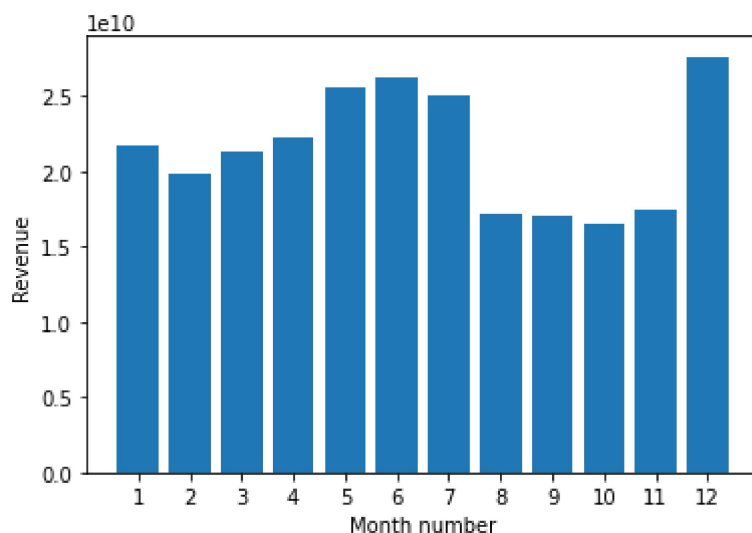
```
In [18]: data['Unit_Cost']=data['Unit_Cost'].astype(float)
```

```
In [19]: data['Revenue']=data['Revenue']*data['Revenue']
```

```
In [20]: data.groupby('Month')['Revenue'].sum()
```

Out[20]: Month
1 21658945027
2 19877856469
3 21380078086
4 22312933812
5 25580420599
6 26277586894
7 25051995307
8 17227283197
9 17082031327
10 16466588055
11 17497336298
12 27624208093
Name: Revenue, dtype: int64

```
In [21]: Months=range(1,13)
plt.bar(Months,data.groupby('Month')['Revenue'].sum())
plt.xticks(Months)
plt.ylabel('Revenue')
plt.xlabel('Month number')
plt.show()
```



Which State has max order

```
In [22]: data.head()
```

```
Out[22]:
```

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Country	State	Pr
0	2013-11-26	26	11	2013	19	Youth (<25)	M	Canada	British Columbia	
1	2015-11-26	26	11	2015	19	Youth (<25)	M	Canada	British Columbia	
2	2014-03-23	23	3	2014	49	Adults (35-64)	M	Australia	New South Wales	
3	2016-03-23	23	3	2016	49	Adults (35-64)	M	Australia	New South Wales	
4	2014-05-15	15	5	2014	47	Adults (35-64)	F	Australia	New South Wales	

```
In [23]: 'Queensland,Alberta,Bayern'.split(',')[0]
```

```
Out[23]: 'Queensland'
```

```
In [24]: def city(x):  
         return x.split(',')[0]
```

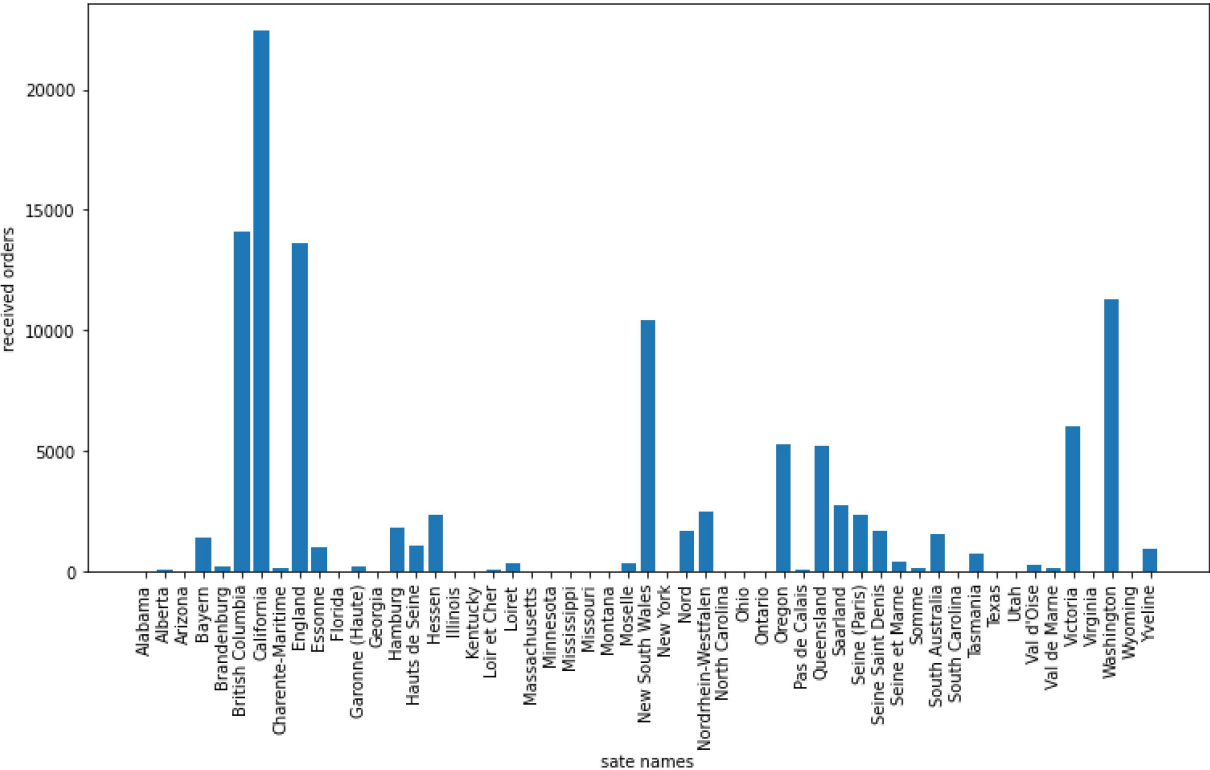
```
In [25]: data['State']=data['State'].astype(object)
```

```
In [26]: data['city']=data['State'].apply(city)
```

```
In [27]: data.groupby('city')['Order_Quantity'].count()
```

```
Out[27]: city  
Alabama          4  
Alberta          56  
Arizona          4  
Bayern           1426  
Brandenburg      198  
British Columbia 14116  
California       22450  
Charente-Maritime 148  
England          13620  
Essonne          994  
Florida          14  
Garonne (Haute)  208  
Georgia          8  
Hamburg          1836  
Hauts de Seine   1084  
Hessen          2384  
Illinois         28  
Kentucky         10  
Loire et Cher    120
```

```
In [28]: plt.rcParams["figure.figsize"] = (12.5,6.5)
plt.bar(data.groupby('city')['city'].count().index,data.groupby('city')['city'].count())
plt.xticks(rotation='vertical')
plt.ylabel('received orders')
plt.xlabel('state names')
plt.show()
```



customer age and revenue

```
In [29]: data.head()
```

Out[29]:

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Country	State	Pr
0	2013-11-26	26	11	2013	19	Youth (<25)	M	Canada	British Columbia	
1	2015-11-26	26	11	2015	19	Youth (<25)	M	Canada	British Columbia	
2	2014-03-23	23	3	2014	49	Adults (35-64)	M	Australia	New South Wales	
3	2016-03-23	23	3	2016	49	Adults (35-64)	M	Australia	New South Wales	
4	2014-05-15	15	5	2014	47	Adults (35-64)	F	Australia	New South Wales	

```
In [30]: data['Revenue'][0].dtype
```

```
Out[30]: dtype('int64')
```

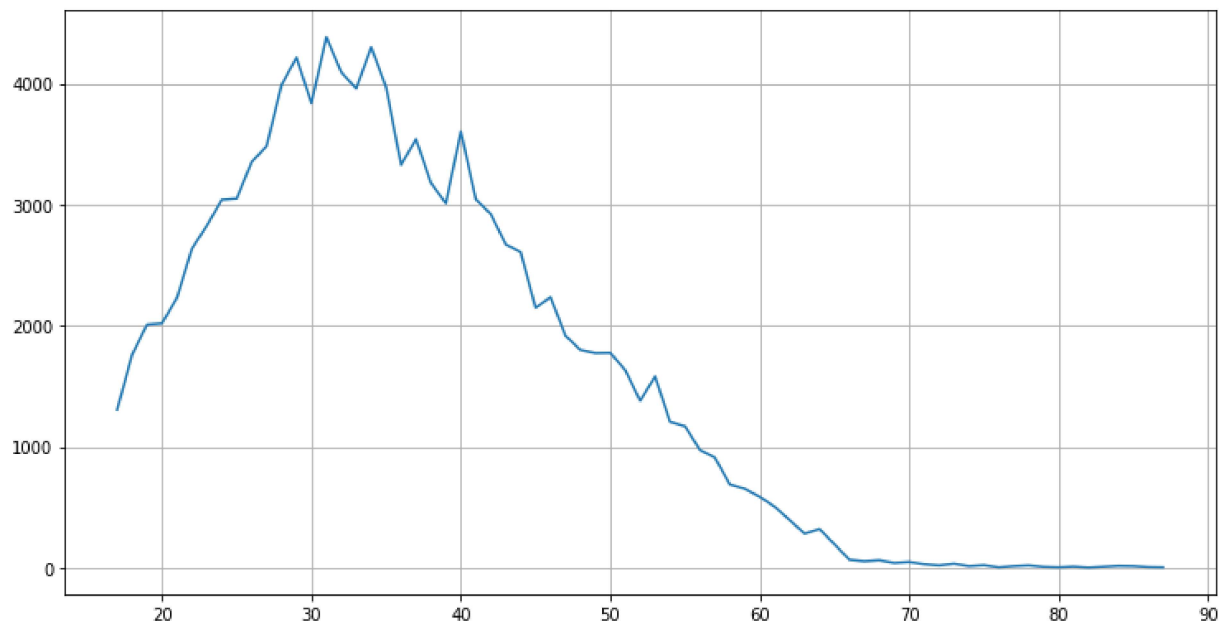
```
In [31]: data['Customer_Age'][0].dtype
```

```
Out[31]: dtype('int64')
```

```
In [32]: keys=[]  
Age=[]  
for key,age_df in data.groupby('Customer_Age'):  
    keys.append(key)  
    Age.append(len(age_df))
```

```
In [33]: plt.grid()  
plt.plot(keys, Age)
```

```
Out[33]: [<matplotlib.lines.Line2D at 0x25ef28ea320>]
```



Product with most orders

In [34]: `data.head()`

Out[34]:

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Country	State	Pr
0	2013-11-26	26	11	2013	19	Youth (<25)	M	Canada	British Columbia	
1	2015-11-26	26	11	2015	19	Youth (<25)	M	Canada	British Columbia	
2	2014-03-23	23	3	2014	49	Adults (35-64)	M	Australia	New South Wales	
3	2016-03-23	23	3	2016	49	Adults (35-64)	M	Australia	New South Wales	
4	2014-05-15	15	5	2014	47	Adults (35-64)	F	Australia	New South Wales	

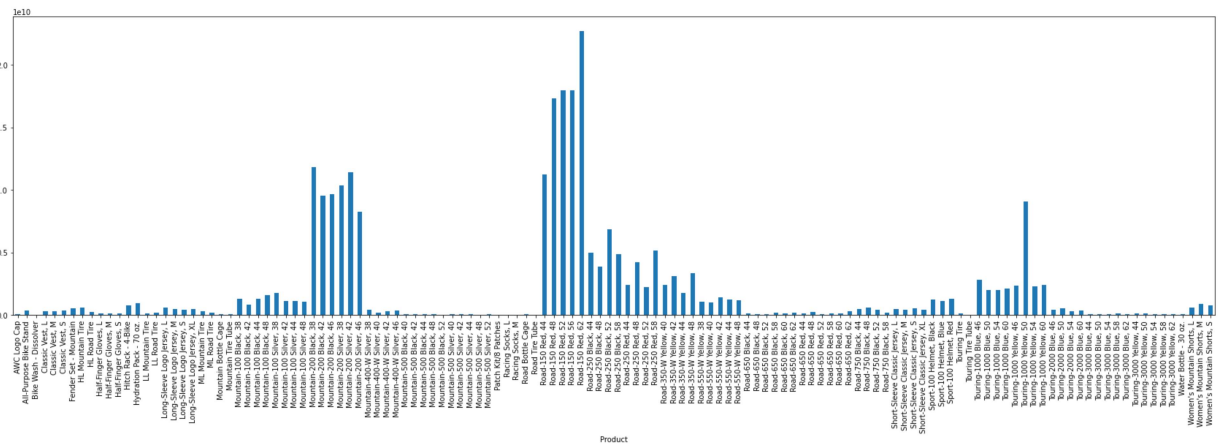
In [35]: `data.shape`

Out[35]: (113036, 19)

In [36]: `data.groupby('Product')['Order_Quantity'].sum().plot(kind='bar')`
`plt.rcParams["figure.figsize"] = (29.5,7.5)`

revenue and product

```
In [37]: data.groupby('Product')['Revenue'].sum().plot(kind='bar')
plt.rcParams["figure.figsize"] = (29.5,7.5)
```



Countries with the most orders and revenue

```
In [38]: 'Canada,Australia,Unites States'.split(',')[0]
```

```
Out[38]: 'Canada'
```

```
In [39]: def Countries(x):
         return x.split('-')[0]
```

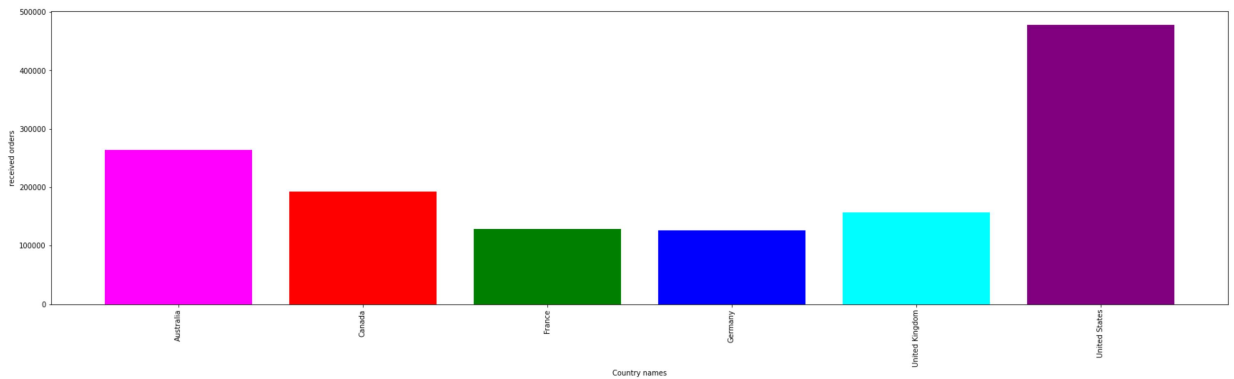
```
In [40]: data['Countries']=data['Country'].apply(Countries)
```

```
In [41]: data.groupby('Countries')['Countries'].count()
```

```
Out[41]: Countries
Australia      23936
Canada         14178
France         10998
Germany        11098
United Kingdom 13620
United States  39206
Name: Countries, dtype: int64
```

```
In [42]: Country=data.groupby('Countries')['Order_Quantity'].sum().index
Quantity=data.groupby('Countries')['Order_Quantity'].sum()
```

```
In [43]: plt.bar(data.groupby('Countries')['Countries'].count().index,data.groupby('Countries').count().values)
plt.xticks(rotation='vertical')
plt.ylabel('received orders')
plt.xlabel('Country names')
plt.grid(False)
plt.show()
```



```
In [44]: 'Accessories,Bikes,Clothing'.split(',')[0]
```

```
Out[44]: 'Accessories'
```

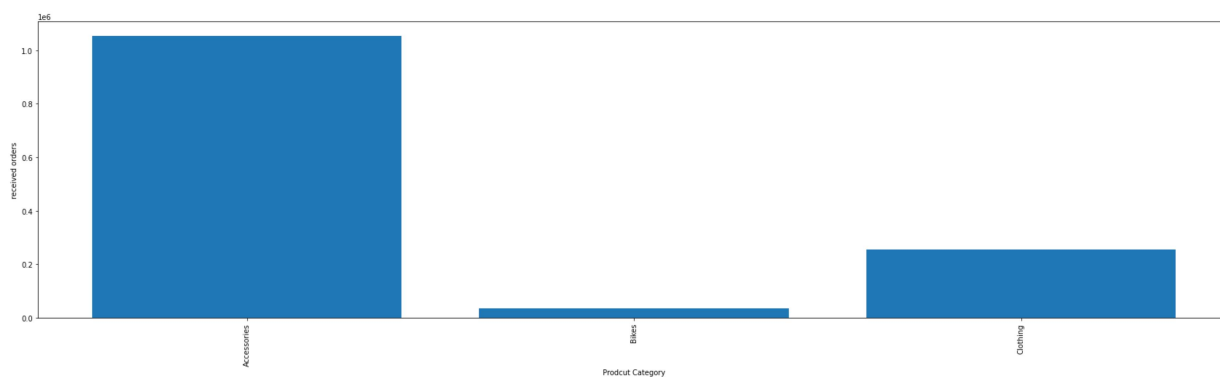
```
In [45]: def Products(x):
return x.split('-')[0]
```

```
In [46]: data['Products']=data['Product_Category'].apply(Products)
```

```
In [47]: data.groupby('Products')['Order_Quantity'].sum()
```

```
Out[47]: Products
Accessories    1054162
Bikes           36411
Clothing       254743
Name: Order_Quantity, dtype: int64
```

```
In [48]: plt.bar(data.groupby('Products')['Products'].count().index,data.groupby('Products')
plt.xticks(rotation='vertical')
plt.ylabel('received orders')
plt.xlabel('Prodcut Category')
plt.grid(False)
plt.show()
```



```
In [ ]:
```