### **SALES ANALYSIS**

#### Import required libraries

```
import pandas as pd
In [1]:
        import glob
In [2]:
In [3]:
        import os
In [4]:
        import numpy as np
        Merge csv files into single file
In [5]: files = os.path.join("/Users/ilker/Pandas-Data-Science-Tasks-master/Sales
In [6]: files = glob.glob(files)
```

```
newData = salesData.to_csv("Sales.csv")
```

salesData = pd.concat(map(pd.read\_csv, files), ignore\_index=True)

> Out[9]: Order Quantity **Price** Order **Product Purchase Address Ordered** Date ID Each Macbook Pro 12/30/19 136 Church St, New 295665 1700 00:01 York City, NY 10001 Laptop LG Washing 12/29/19 562 2nd St, New 295666 1 600.0 York City, NY 10001 Machine 07:03 **USB-C Charging** 12/12/19 277 Main St, New 295667 1 11.95 York City, NY 10001 Cable 18:21 410 6th St, San 12/22/19 27in FHD Monitor Francisco, CA 295668 149.99 15:13 94016 USB-C Charging 12/18/19 43 Hill St, Atlanta, 295669 11.95 1 Cable 12:38 GA 30301 700 West St, San Bose SoundSport 12/19/19 295843 1 99.99 195 Francisco, CA Headphones 14:02 94016 663 12th St, Atlanta, AA Batteries (4-12/31/19 196 295844 3.84 pack) 17:51 GA 30301 955 Sunset St. San Lightning 12/20/19 197 295845 14.95 Francisco, CA **Charging Cable** 09:53 94016 453 Johnson St, 12/27/19 Flatscreen TV 300 198 295846 1 19:22 Atlanta, GA 30301 34in Ultrawide 12/13/19 40 Madison St, 199 295847 379.99 Monitor 15:56 Dallas, TX 75001 200 rows × 6 columns

In [10]: salesData["Month"] = salesData["Order Date"].str[0:2]

### Clean up Data

In [11]: salesData.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 186850 entries, 0 to 186849

Data columns (total 7 columns):

#	Column	Non-Null Count	Dtype
0	Order ID	186305 non-null	object
1	Product	186305 non-null	object
2	Quantity Ordered	186305 non-null	object
3	Price Each	186305 non-null	object
4	Order Date	186305 non-null	object
5	Purchase Address	186305 non-null	object
6	Month	186305 non-null	object

dtypes: object(7) memory usage: 10.0+ MB

newSalesData = salesData.dropna(how = "any") In [12]:

In [13]: newSdata = newSalesData[newSalesData["Order Date"].str[0:2] != "Or"]

In [14]: newSdata

Out[14]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
0	295665	Macbook Pro Laptop	1	1700	12/30/19 00:01	136 Church St, New York City, NY 10001	12
1	295666	LG Washing Machine	1	600.0	12/29/19 07:03	562 2nd St, New York City, NY 10001	12
2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001	12
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016	12
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301	12
•••				•••			•••
186845	222905	AAA Batteries (4-pack)	1	2.99	06/07/19 19:02	795 Pine St, Boston, MA 02215	06
186846	222906	27in FHD Monitor	1	149.99	06/01/19 19:29	495 North St, New York City, NY 10001	06
186847	222907	USB-C Charging Cable	1	11.95	06/22/19 18:57	319 Ridge St, San Francisco, CA 94016	06
186848	222908	USB-C Charging Cable	1	11.95	06/26/19 18:35	916 Main St, San Francisco, CA 94016	06
186849	222909	AAA Batteries (4-pack)	1	2.99	06/25/19 14:33	209 11th St, Atlanta, GA 30301	06

185950 rows × 7 columns

In [15]: newSdata["Month"].astype("int")

```
Out[15]: 0
                    12
         1
                    12
                    12
         3
                    12
         4
                    12
         186845
                     6
         186846
                    6
         186847
         186848
                     6
         186849
         Name: Month, Length: 185950, dtype: int64
```

## Q1: What was the best month for sales? How much was earned that month?

```
In [16]: month_group = newSdata.groupby(["Month"])
In [17]: month_group.get_group("12")
```

Out[17]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
0	295665	Macbook Pro Laptop	1	1700	12/30/19 00:01	136 Church St, New York City, NY 10001	12
1	295666	LG Washing Machine	1	600.0	12/29/19 07:03	562 2nd St, New York City, NY 10001	12
2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001	12
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016	12
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301	12
•••				•••			•••
115313	293993	AAA Batteries (4-pack)	1	2.99	12/01/19 00:41	999 10th St, San Francisco, CA 94016	12
115418	294093	USB-C Charging Cable	1	11.95	12/01/19 00:37	51 Cherry St, New York City, NY 10001	12
115783	294441	Wired Headphones	1	11.99	12/01/19 00:52	469 Sunset St, San Francisco, CA 94016	12
115980	294629	Wired Headphones	1	11.99	12/01/19 01:54	668 Ridge St, San Francisco, CA 94016	12
117040	295636	AA Batteries (4-pack)	2	3.84	12/01/19 00:20	204 Walnut St, Dallas, TX 75001	12

24984 rows × 7 columns

## Convert columns the correct type

In [18]: newSdata['Quantity Ordered'] = pd.to\_numeric(newSdata['Quantity Ordered']
 newSdata['Price Each'] = pd.to\_numeric(newSdata['Price Each'])

> /var/folders/1m/24t\_f0\_j3bb46tzcw1s7w04w0000gp/T/ipykernel\_8729/29649400 57.py:1: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame. Try using .loc[row indexer,col indexer] = value instead See the caveats in the documentation: https://pandas.pydata.org/pandas-d ocs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy newSdata['Quantity Ordered'] = pd.to\_numeric(newSdata['Quantity Ordere /var/folders/1m/24t\_f0\_j3bb46tzcw1s7w04w0000gp/T/ipykernel\_8729/29649400 57.py:2: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame. Try using .loc[row\_indexer,col\_indexer] = value instead See the caveats in the documentation: https://pandas.pydata.org/pandas-d ocs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy newSdata['Price Each'] = pd.to\_numeric(newSdata['Price Each'])

In [19]: newSdata['Price Each'].dtype

Out[19]: dtype('float64')

#### Add sales column

In [20]: newSdata["Sales"] = newSdata['Quantity Ordered'] \* newSdata["Price Each"] /var/folders/1m/24t\_f0\_j3bb46tzcw1s7w04w0000gp/T/ipykernel\_8729/15885602 07.py:1: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame. Try using .loc[row\_indexer,col\_indexer] = value instead See the caveats in the documentation: https://pandas.pydata.org/pandas-d ocs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy newSdata["Sales"] = newSdata['Quantity Ordered'] \* newSdata["Price Eac h"]

In [21]: newSdata

Out[21]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales
0	295665	Macbook Pro Laptop	1	1700.00	12/30/19 00:01	136 Church St, New York City, NY 10001	12	1700.00
1	295666	LG Washing Machine	1	600.00	12/29/19 07:03	562 2nd St, New York City, NY 10001	12	600.00
2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001	12	11.95
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016	12	149.99
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301	12	11.95
186845	222905	AAA Batteries (4-pack)	1	2.99	06/07/19 19:02	795 Pine St, Boston, MA 02215	06	2.99
186846	222906	27in FHD Monitor	1	149.99	06/01/19 19:29	495 North St, New York City, NY 10001	06	149.99
186847	222907	USB-C Charging Cable	1	11.95	06/22/19 18:57	319 Ridge St, San Francisco, CA 94016	06	11.95
186848	222908	USB-C Charging Cable	1	11.95	06/26/19 18:35	916 Main St, San Francisco, CA 94016	06	11.95
186849	222909	AAA Batteries (4-pack)	1	2.99	06/25/19 14:33	209 11th St, Atlanta, GA 30301	06	2.99

185950 rows × 8 columns

In [22]: newSdata

1 295666

295667

295668

295669

**186845** 222905

**186846** 222906

**186847** 222907

**186848** 222908

**186849** 222909

12.07 11.1					54				
Out[22]:		Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales
	0	295665	Macbook Pro Laptop	1	1700.00	12/30/19 00:01	136 Church St, New York City, NY 10001	12	1700.00

1

1

1

1

1

LG

Washing

Machine

USB-C

Cable

Charging

27in FHD

Monitor

USB-C

Cable

AAA

**Batteries** 

(4-pack)

27in FHD

Monitor

USB-C

Cable

USB-C

Cable

AAA

Charging

Batteries

(4-pack)

Charging

Charging

562 2nd

St, New

York City,

NY 10001

277 Main

St, New

York City,

NY 10001

410 6th St,

Francisco, CA 94016

43 Hill St.

Atlanta, GA

30301

795 Pine

St, Boston,

MA 02215

495 North

St, New

York City, NY 10001

319 Ridge

Francisco,

CA 94016

916 Main

Francisco,

CA 94016

209 11th

St, Atlanta,

GA 30301

St, San

St, San

San

12

12

12

12

06

06

06

06

06

600.00

11.95

149.99

11.95

2.99

149.99

11.95

11.95

2.99

12/29/19

12/12/19

12/22/19

12/18/19

06/07/19

06/01/19

06/22/19

06/26/19

06/25/19

14:33

18:35

18:57

19:29

19:02

12:38

15:13

18:21

07:03

600.00

11.95

149.99

11.95

2.99

149.99

11.95

11.95

2.99

185950 rows × 8 columns

In [23]: SalesGr = newSdata.groupby(["Month"]).sum()

/var/folders/1m/24t\_f0\_j3bb46tzcw1s7w04w0000gp/T/ipykernel\_8729/16422665 90.py:1: FutureWarning: The default value of numeric\_only in DataFrameGr oupBy.sum is deprecated. In a future version, numeric\_only will default to False. Either specify numeric\_only or select only columns which shoul d be valid for the function.

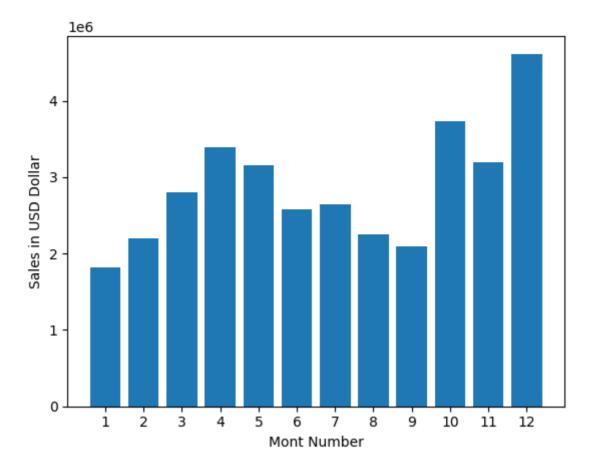
SalesGr = newSdata.groupby(["Month"]).sum()

In [24]: SalesGr

plt.xlabel("Mont Number")

plt.show()

Sales **Quantity Ordered Price Each** Out[24]: Month 01 10903 1811768.38 1822256.73 02 13449 2188884.72 2202022.42 03 17005 2791207.83 2807100.38 20558 3367671.02 3390670.24 04 05 18667 3135125.13 3152606.75 06 15253 2562025.61 2577802.26 07 16072 2632539.56 2647775.76 80 13448 2230345.42 2244467.88 09 13109 2084992.09 2097560.13 10 22703 3715554.83 3736726.88 11 19798 3180600.68 3199603.20 12 28114 4588415.41 4613443.34 In [25]: import matplotlib.pyplot as plt In [26]: month =range(1,13) In [27]: plt.bar(month, SalesGr["Sales"]) plt.xticks(month) plt.ylabel("Sales in USD Dollar")



#### Q2 What city had the highest number of sales?

#### task: add city column

```
In [28]:
         def get_city(address):
             return address.split(",")[1]
         def get_state(address):
             return address.split(",")[2].split(" ")[1]
         newSdata["city"] = newSdata["Purchase Address"].apply(lambda x : get_city
         /var/folders/1m/24t_f0_j3bb46tzcw1s7w04w0000gp/T/ipykernel_8729/30707175
         4.py:9: SettingWithCopyWarning:
         A value is trying to be set on a copy of a slice from a DataFrame.
         Try using .loc[row_indexer,col_indexer] = value instead
         See the caveats in the documentation: https://pandas.pydata.org/pandas-d
         ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
           newSdata["city"] = newSdata["Purchase Address"].apply(lambda x : get_c
         ity(x)+" "+ get_state(x))
In [29]:
         newSdata
```

_			г	$\overline{}$	$\overline{}$	п	
- 1 1	11	Τ.	-	- /	u	- 1	
$\circ$	u	L	-L	_	J	ы	

		Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	
	0	295665	Macbook Pro Laptop	1	1700.00	12/30/19 00:01	136 Church St, New York City, NY 10001	12	1700.00	New Cit
	1	295666	LG Washing Machine	1	600.00	12/29/19 07:03	562 2nd St, New York City, NY 10001	12	600.00	New Cit
	2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001	12	11.95	New Cit
	3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016	12	149.99	Franc
	4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301	12	11.95	At
1	86845	222905	AAA Batteries (4-pack)	1	2.99	06/07/19 19:02	795 Pine St, Boston, MA 02215	06	2.99	Вс
1	86846	222906	27in FHD Monitor	1	149.99	06/01/19 19:29	495 North St, New York City, NY 10001	06	149.99	New Cit
1	186847	222907	USB-C Charging Cable	1	11.95	06/22/19 18:57	319 Ridge St, San Francisco, CA 94016	06	11.95	Franc
1	86848	222908	USB-C Charging Cable	1	11.95	06/26/19 18:35	916 Main St, San Francisco, CA 94016	06	11.95	Franc
1	86849	222909	AAA Batteries (4-pack)	1	2.99	06/25/19 14:33	209 11th St, Atlanta, GA 30301	06	2.99	At

185950 rows × 9 columns

In [30]: results = newSdata.groupby(["city"]).sum()

/var/folders/1m/24t\_f0\_j3bb46tzcw1s7w04w0000gp/T/ipykernel\_8729/41448368 96.py:1: FutureWarning: The default value of numeric\_only in DataFrameGr oupBy.sum is deprecated. In a future version, numeric\_only will default to False. Either specify numeric\_only or select only columns which shoul d be valid for the function.

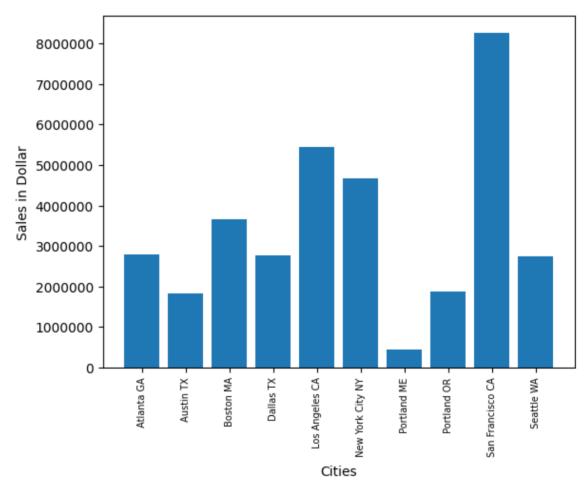
results = newSdata.groupby(["city"]).sum()

results In [31]: Out[31]: **Quantity Ordered Price Each Sales** city 16602 2779908.20 2795498.58 Atlanta GA **Austin TX** 11153 1809873.61 1819581.75 **Boston MA** 22528 3637409.77 3661642.01 **Dallas TX** 16730 2752627.82 2767975.40 Los Angeles CA 33289 5421435.23 5452570.80 **New York City NY** 27932 4635370.83 4664317.43 **Portland ME** 2750 447189.25 449758.27 **Portland OR** 11303 1860558.22 1870732.34 San Francisco CA 50239 8211461.74 8262203.91 **Seattle WA** 16553 2733296.01 2747755.48 In [32]: results.index d OR', ' San Francisco CA', ' Seattle WA'], dtype='object', name='city')

In [33]: newSdata

Out[33]:		Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	
	0	295665	Macbook Pro Laptop	1	1700.00	12/30/19 00:01	136 Church St, New York City, NY 10001	12	1700.00	New Cit
	1	295666	LG Washing Machine	1	600.00	12/29/19 07:03	562 2nd St, New York City, NY 10001	12	600.00	New Cit
	2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001	12	11.95	New Cit
	3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016	12	149.99	Franc
	4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301	12	11.95	At
	•••			•••		•••	•••			
	186845	222905	AAA Batteries (4-pack)	1	2.99	06/07/19 19:02	795 Pine St, Boston, MA 02215	06	2.99	Вс
	186846	222906	27in FHD Monitor	1	149.99	06/01/19 19:29	495 North St, New York City, NY 10001	06	149.99	New Cit
	186847	222907	USB-C Charging Cable	1	11.95	06/22/19 18:57	319 Ridge St, San Francisco, CA 94016	06	11.95	Franc
	186848	222908	USB-C Charging Cable	1	11.95	06/26/19 18:35	916 Main St, San Francisco, CA 94016	06	11.95	Franc
	186849	222909	AAA Batteries (4-pack)	1	2.99	06/25/19 14:33	209 11th St, Atlanta, GA 30301	06	2.99	At
	185950 r	ows × 9 d	columns							
In [34]:	cities	= [city	for city	/, df in	newSdat	a.groupb	y("city")	1		
In [35]:	cities									

Out[36]: Text(0, 0.5, 'Sales in Dollar')



Q3:What time should we display advertisement to maximize likelihood of customers buying product?

```
In [37]: newSdata
```

Quantity **Price** Order Out [37]: Order **Purchase Product** Month Sales **Ordered Address** ID Each Date 136 Macbook Church 12/30/19 New 0 295665 Pro 1 1700.00 St, New 12 1700.00 00:01 Cit Laptop York City, NY 10001 562 2nd LG 12/29/19 St, New New 1 295666 Washing 1 600.00 12 600.00 07:03 York City, Cit Machine NY 10001 277 Main USB-C 12/12/19 St, New New 295667 2 Charging 1 11.95 12 11.95 18:21 York City, Cit Cable NY 10001 410 6th 27in 12/22/19 St, San 295668 FHD 149.99 12 149.99 Franc 15:13 Francisco, Monitor CA 94016 USB-C 43 Hill St, 12/18/19 Αt 295669 Charging 1 11.95 Atlanta, 12 11.95 12:38 GA 30301 Cable 795 Pine AAA 06/07/19 St, Вc **186845** 222905 1 2.99 2.99 **Batteries** 06 19:02 Boston, (4-pack) MA 02215 495 North 27in 06/01/19 St, New New **186846** 222906 FHD 1 149.99 06 149.99 19:29 York City, Cit Monitor NY 10001 319 Ridge USB-C 06/22/19 St, San 186847 222907 Charging 1 11.95 06 11.95 Franc 18:57 Francisco, Cable CA 94016 916 Main USB-C 06/26/19 St, San **186848** 222908 1 11.95 06 Charging 11.95 France Francisco, 18:35 Cable CA 94016 209 11th AAA06/25/19 St, Αt **186849** 222909 2.99 06 2.99 **Batteries** 1 14:33 Atlanta,

185950 rows × 9 columns

(4-pack)

newSdata["Order Date"] = pd.to\_datetime(newSdata["Order Date"]) In [38]:

GA 30301

> /var/folders/1m/24t\_f0\_j3bb46tzcw1s7w04w0000gp/T/ipykernel\_8729/33258907 02.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame. Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-d ocs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy newSdata["Order Date"] = pd.to\_datetime(newSdata["Order Date"])

In [39]: newSdata

Out[39]:		Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	
	0	295665	Macbook Pro Laptop	1	1700.00	2019- 12-30 00:01:00	136 Church St, New York City, NY 10001	12	1700.00	New Cit
	1	295666	LG Washing Machine	1	600.00	2019- 12-29 07:03:00	562 2nd St, New York City, NY 10001	12	600.00	New Cit
	2	295667	USB-C Charging Cable	1	11.95	2019- 12-12 18:21:00	277 Main St, New York City, NY 10001	12	11.95	New Cit
	3	295668	27in FHD Monitor	1	149.99	2019- 12-22 15:13:00	410 6th St, San Francisco, CA 94016	12	149.99	Fran
	4	295669	USB-C Charging Cable	1	11.95	2019- 12-18 12:38:00	43 Hill St, Atlanta, GA 30301	12	11.95	At
	•••								•••	
	186845	222905	AAA Batteries (4-pack)	1	2.99	2019- 06-07 19:02:00	795 Pine St, Boston, MA 02215	06	2.99	Вс
	186846	222906	27in FHD Monitor	1	149.99	2019- 06-01 19:29:00	495 North St, New York City, NY 10001	06	149.99	New Cit
	186847	222907	USB-C Charging Cable	1	11.95	2019- 06-22 18:57:00	319 Ridge St, San Francisco, CA 94016	06	11.95	Fran
	186848	222908	USB-C Charging Cable	1	11.95	2019- 06-26 18:35:00	916 Main St, San Francisco, CA 94016	06	11.95	Fran
	186849	222909	AAA Batteries (4-pack)	1	2.99	2019- 06-25 14:33:00	209 11th St, Atlanta, GA 30301	06	2.99	At
	185950 r	ows × 9 o	columns							

In [40]: newSdata["Hour"] = newSdata["Order Date"].dt.minute

/var/folders/1m/24t\_f0\_j3bb46tzcw1s7w04w0000gp/T/ipykernel\_8729/18762708
93.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame. Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copynewSdata["Hour"] = newSdata["Order Date"].dt.hour

In [44]: newSdata["Minute"] = newSdata["Order Date"].dt.hour

/var/folders/1m/24t\_f0\_j3bb46tzcw1s7w04w0000gp/T/ipykernel\_8729/80444347 3.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copynewSdata["Minute"] = newSdata["Order Date"].dt.hour

In [45]: newSdata

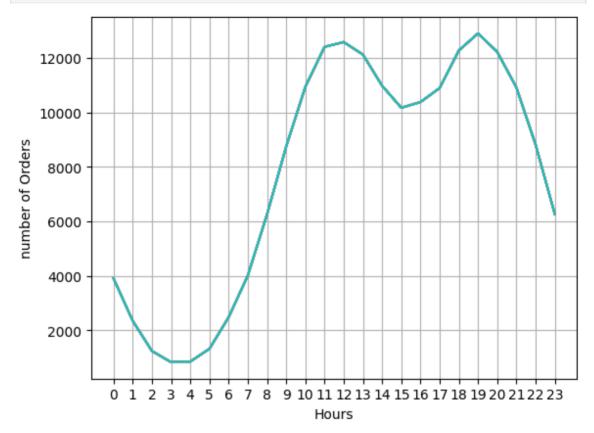
Out[45]: **Price** Order Order Quantity **Purchase Product** Month Sales **Ordered Address** ID Each Date 136 Macbook 2019-Church New 0 295665 Pro 1 1700.00 12-30 St, New 12 1700.00 Cit Laptop 00:01:00 York City, NY 10001 562 2nd LG 2019-St, New New 295666 Washing 600.00 12-29 12 600.00 York City, Cit 07:03:00 Machine NY 10001 277 Main USB-C 2019-St, New New 2 295667 Charging 12 11.95 1 11.95 12-12 York City, Cit Cable 18:21:00 NY 10001 410 6th 27in 2019-St, San 295668 FHD 149.99 12-22 12 149.99 Fran Francisco, Monitor 15:13:00 CA 94016 USB-C 2019-43 Hill St, Αt 295669 Charging 1 11.95 12-18 Atlanta, 12 11.95 Cable 12:38:00 GA 30301 795 Pine AAA 2019-St, Вс **186845** 222905 **Batteries** 1 2.99 06-07 06 2.99 Boston, (4-pack) 19:02:00 MA 02215 495 North 27in 2019-St, New New **186846** 222906 FHD 149.99 06-01 06 149.99 1 York City, Cit Monitor 19:29:00 NY 10001 319 Ridge USB-C 2019-St, San 186847 222907 Charging 1 11.95 06-22 06 11.95 Fran Francisco, Cable 18:57:00 CA 94016 916 Main USB-C 2019-St, San 1 06 **186848** 222908 Charging 11.95 06-26 11.95 Fran Francisco, Cable 18:35:00 CA 94016 209 11th AAA2019-St, Αt **186849** 222909 2.99 2.99 **Batteries** 1 06-25 06 Atlanta, (4-pack) 14:33:00 GA 30301 185950 rows × 11 columns

```
In [50]: hours = [hour for hour, df in newSdata.groupby("Hour")]

filter = newSdata.groupby(["Hour"]).count()

plt.xticks(hours)
plt.xlabel("Hours")
plt.ylabel("number of Orders")
```

```
plt.grid()
plt.plot(hours, filter)
plt.show()
# the answer we can say 11 am and 7 pm.
```



# What products are most often sold together? (Order ID si aynı olan birlikte satılmıştır.)

Out[62]

:		Order ID	Grouped
	16	295681	Google Phone, USB-C Charging Cable, Bose SoundSp
	36	295698	Vareebadd Phone, USB-C Charging Cable
	42	295703	AA Batteries (4-pack),Bose SoundSport Headphones
	66	295726	iPhone,Lightning Charging Cable
	76	295735	iPhone,Apple Airpods Headphones,Wired Headphones
	•••		
	186798	222863	27in FHD Monitor, Bose SoundSport Headphones
	186804	222868	iPhone,Apple Airpods Headphones
	186819	222882	Apple Airpods Headphones, AA Batteries (4-pack)
	186822	222884	Google Phone, USB-C Charging Cable
	186842	222903	iPhone,Lightning Charging Cable

7136 rows × 2 columns

```
In [70]: from collections import Counter
         from itertools import combinations
         count = Counter()
         for row in df["Grouped"]:
             row_list = row.split(",")
             count.update(Counter(combinations(row_list,2)))
         #count.most_common(10)
         for key,value in count.most common(10):
             print(key, value)
         ('iPhone', 'Lightning Charging Cable') 1005
         ('Google Phone', 'USB-C Charging Cable') 987
         ('iPhone', 'Wired Headphones') 447
         ('Google Phone', 'Wired Headphones') 414
         ('Vareebadd Phone', 'USB-C Charging Cable') 361
         ('iPhone', 'Apple Airpods Headphones') 360
         ('Google Phone', 'Bose SoundSport Headphones') 220
         ('USB-C Charging Cable', 'Wired Headphones') 160
         ('Vareebadd Phone', 'Wired Headphones') 143
         ('Lightning Charging Cable', 'Wired Headphones') 92
```

#### What product sold the most?

```
In [71]: newSdata
```

Out[71]:		Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	
	0	295665	Macbook Pro Laptop	1	1700.00	2019- 12-30 00:01:00	136 Church St, New York City, NY 10001	12	1700.00	New Cit
	1	295666	LG Washing Machine	1	600.00	2019- 12-29 07:03:00	562 2nd St, New York City, NY 10001	12	600.00	New Cit
	2	295667	USB-C Charging Cable	1	11.95	2019- 12-12 18:21:00	277 Main St, New York City, NY 10001	12	11.95	New Cit
	3	295668	27in FHD Monitor	1	149.99	2019- 12-22 15:13:00	410 6th St, San Francisco, CA 94016	12	149.99	Fran
	4	295669	USB-C Charging Cable	1	11.95	2019- 12-18 12:38:00	43 Hill St, Atlanta, GA 30301	12	11.95	At
	•••									
	186845	222905	AAA Batteries (4-pack)	1	2.99	2019- 06-07 19:02:00	795 Pine St, Boston, MA 02215	06	2.99	Вс
	186846	222906	27in FHD Monitor	1	149.99	2019- 06-01 19:29:00	495 North St, New York City, NY 10001	06	149.99	New Cit
	186847	222907	USB-C Charging Cable	1	11.95	2019- 06-22 18:57:00	319 Ridge St, San Francisco, CA 94016	06	11.95	Fran
	186848	222908	USB-C Charging Cable	1	11.95	2019- 06-26 18:35:00	916 Main St, San Francisco, CA 94016	06	11.95	Fran
	186849	222909	AAA Batteries (4-pack)	1	2.99	2019- 06-25 14:33:00	209 11th St, Atlanta, GA 30301	06	2.99	At
	185950 r	ows × 11	columns							
In [73]:										
In [77]:										
	product	s = [pro	oduct <b>fo</b>	r product	, df <b>i</b> n	product	_group]			
In [1]:	prices	= newSda	ata["Prod	duct"].me	ean()["P	rice Eac	h"]			

```
fig, ax1 = plt.subplots()
        ax2 = ax1.twinx()
        ax1.bar(products,quantity_ordered)
        ax2.plot(products,prices,"b-")
        ax1.set_xlabel("Product Name")
        ax1.set_ylabel("Quantity ordered")
        ax2.set_xlabel("price", color = "b")
        ax1.set_xticklabels(products, rotation = "vertical", size = 8)
        plt.bar(products, quantity_ordered)
        plt.show()
                                                   Traceback (most recent call la
        NameError
        st)
        Cell In[1], line 1
        ----> 1 prices = newSdata["Product"].mean()["Price Each"]
              3 fig, ax1 = plt.subplots()
              4 ax2 = ax1.twinx()
        NameError: name 'newSdata' is not defined
In []:
```