

شرح پروژه

تحلیل آماری

داده های فروشگاه لوازم الکترونیکی

ترم بهمن ۱۴۰۱

استاد راهنما : جناب آقای دکتر علی غفاری برانقار

آناهیتا کلینی

شماره دانشجویی:
۲۱۴۹۸۲۰۶۴۶۴۵

برای تحلیل و پاسخ به سوالات کسب و کار درباره داده‌های فروش یک فروشگاه الکترونیکی به مدت ۱۲ ماه، از زبان برنامه نویسی پایتون ، ابزار ژوپیتر نوت بوک به عنوان محیط پیاده سازی ، کتابخانه های `pandas` و `matplotlib` استفاده کردم. داده‌ها شامل خریدهای صورت گرفته در فروشگاه الکترونیکی است که بر اساس ماه، نوع محصول، هزینه، آدرس خرید و موارد دیگر تجزیه و تحلیل شده‌اند.

من با شروع پاکسازی داده‌ها، به انجام وظایف زیر پرداختم:

- حذف مقادیر NaN از DataFrame

- حذف ردیف‌ها بر اساس یک شرط (برای مثال صرفاً قسمت ماه از یک تاریخ را لازم داشتیم نه سال و روز آن)

- تغییر نوع ستون‌ها (استفاده از روش‌هایی مانند `'astype'`، `'to_datetime'`، `'to_numeric'`)

در ادامه بعد از پاکسازی داده‌ها، به کاوش داده‌ها می‌پردازم. در این بخش، به طرح ۵ سوال مرتبط با کسب و کار داده‌ها پرداختم:

- بهترین ماه برای فروش محصولات کدام ماه است؟ درآمد آن ماه چقدر بود؟

- کدام شهر بیشترین محصولات را فروخته است؟

- در چه زمانی باید تبلیغات را نمایش دهیم تا احتمال خرید مشتریان را به حداکثر برسانیم؟

- محصولاتی که بیشترین بار با هم فروخته می‌شوند (به طور همزمان خریداری می‌شوند) چیست؟

- کدام محصول بیشترین فروش را داشت؟ چرا فکر می‌کنید بیشترین فروش را داشته است؟

برای پاسخ به این سوالات، از روش‌های مختلف کتابخانه های `Pandas` و `Matplotlib` استفاده کردم که شامل موارد زیر می‌شود:

- ترکیب چندین فایل CSV با هم برای ایجاد یک DataFrame جدید (`pd.concat`)

- افزودن ستون‌ها

- تجزیه cellها به صورت رشته برای ایجاد ستون‌های جدید (`str`).

- استفاده از روش `apply()`.

- استفاده از `groupby` برای انجام تحلیل جمعی

- ترسیم نمودارهای میله‌ای و خطی برای تجسم نتایج ما

- برچسب‌گذاری نمودارها



Sales Analysis

I use Python Pandas & Python Matplotlib to analyze and answer business questions about 12 months worth of sales data. The data contains hundreds of thousands of electronics store purchases broken down by month, product type, cost, purchase address, etc. I start by cleaning our data. Tasks during this section include:

- Drop NaN values from DataFrame
- Removing rows based on a condition
- Change the type of columns (to_numeric, to_datetime, astype)

Once I have cleaned up our data a bit, I move the data exploration section. In this section I explore 5 high level business questions related to our data:

- What was the best month for sales? How much was earned that month?
- What city sold the most product?
- What time should we display advertisements to maximize the likelihood of customer's buying product?
- What products are most often sold together?
- What product sold the most? Why do you think it sold the most?

To answer these questions I walk through many different pandas & matplotlib methods. They include:

- Concatenating multiple csvs together to create a new DataFrame (pd.concat)
- Adding columns
- Parsing cells as strings to make new columns (.str)
- Using the .apply() method
- Using groupby to perform aggregate analysis
- Plotting bar charts and lines graphs to visualize our results
- Labeling our graphs

Import necessary libraries

```
In [29]: import pandas as pd
import os
%matplotlib inline

import matplotlib.pyplot as plt
```

در این قسمت کتابخانه های مورد استفاده در برنامه را ایمپورت می کنیم.

Merge data from each month into one CSV

```
In [3]: files = [file for file in os.listdir('/Users/apple/Downloads/AnalysisProject/SalesAnalysis/Sales_Data')]

all_months_data = pd.DataFrame()
for file in files:
    df = pd.read_csv("/Users/apple/Downloads/AnalysisProject/SalesAnalysis/Sales_Data/"+file)
    all_months_data = pd.concat([all_months_data, df])

all_months_data.to_csv('all_data.csv', index=False)
```

در این قسمت داده های 12 ماه را به طور یکجا در یک فایل CSV قرار می دهیم.(حال به جای 12 فایل به ازای هر ماه یک فایل داریم به ازای 12 ماه)

Read in updated dataframe

```
In [4]: all_data = pd.read_csv("all_data.csv")
all_data.head()
```

Out[4]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	295665	Macbook Pro Laptop	1	1700	12/30/19 00:01	136 Church St, New York City, NY 10001
1	295666	LG Washing Machine	1	600.0	12/29/19 07:03	562 2nd St, New York City, NY 10001
2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301

در این بخش فایل all_data که از پیوستن 12 تا فایل به دست آوردیم را چک می کنیم که بله درست ساخته و لود شده.

پاکسازی داده‌ها در تحلیل یک فروشگاه الکترونیکی بسیار مهم است. تراکنش و پرداخت زیادی هر روز انجام می‌شود و این باعث می‌شود حجم بسیار زیادی از داده‌ها ایجاد شود. پاکسازی داده‌ها می‌تواند به شرکت‌های فروشگاه الکترونیکی کمک کند تا از داده‌های خود بهترین استفاده را برای اهداف خود بکنند. به طور خاص، اهمیت پاکسازی داده در تحلیل یک فروشگاه الکترونیکی عبارت است از:

1 - بهبود تحلیل داده‌ها: پاکسازی داده‌ها می‌تواند به شرکت‌های فروشگاه الکترونیکی کمک کند تا از داده‌های خود بهترین استفاده را بکنند و تحلیل داده‌های دقیق‌تری را ارائه دهند. این باعث می‌شود تصمیم‌گیری‌های بهتری در مورد محصولات، قیمت‌ها، تبلیغات و مدیریت موجودی اتخاذ شود.

2 - بهبود تجربه مشتری: پاکسازی داده‌ها می‌تواند به شرکت‌های فروشگاه الکترونیکی کمک کند تا تجربه مشتری را بهبود بخشند. با داشتن داده‌هایی دقیق و کامل درباره مشتریان، شرکت‌ها می‌توانند بهترین راهکارهایی را برای بهبود تجربه مشتری ارائه دهند.

3 - کاهش هزینه‌ها: پاکسازی داده‌ها می‌تواند به شرکت‌های فروشگاه الکترونیکی کمک کند تا هزینه‌های خود را کاهش دهند. حذف داده‌هایی که دیگر نیازی به آن‌ها نیست، می‌تواند حجم داده‌ها را کاهش دهد و هزینه‌های ذخیره‌سازی داده‌ها را کاهش بدهد.

4 - بهبود امنیت: پاکسازی داده‌ها می‌تواند به شرکت‌های فروشگاه الکترونیکی کمک کند تا از افشای داده‌های حساس جلوگیری کنند. با حذف داده‌هایی که دیگر نیازی به آن‌ها نیست، شرکت‌ها می‌توانند از ریسک افشای داده‌ها و نقض حریم شخصی کاربران جلوگیری کنند.

Clean up the data!

The first step in this is figuring out what we need to clean. I have found in practice, that you find things you need to clean as you perform operations and get errors. Based on the error, you decide how you should go about cleaning the data

Drop rows of NAN

In [5]: # Find NAN

```
nan_df = all_data[all_data.isna().any(axis=1)]
display(nan_df.head())

all_data = all_data.dropna(how='all')
all_data.head()
```

Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
264	NaN	NaN	NaN	NaN	NaN
648	NaN	NaN	NaN	NaN	NaN
680	NaN	NaN	NaN	NaN	NaN
1385	NaN	NaN	NaN	NaN	NaN
1495	NaN	NaN	NaN	NaN	NaN

در این قسمت ردیف‌هایی از دیتافریم که حاوی NaN بودند را حذف کردیم یا به اصطلاح دراپ کردیم.

Get rid of text in order date column

```
In [6]: all_data = all_data[all_data['Order Date'].str[0:2] != 'Or']
```

در این قسمت از تاریخ مدنظر سال و روز را نیاز نداریم و تنها شماره ی ماه کافیهست. از طرفی ممکن است خطایی تحت عنوان مقدار Or رخ بدهد. یعنی ما 2 عدد از ماه تاریخ را می خواهیم اما ممکن است در ردیف هایی 2 عدد رشته باشند و Or برگرداندن برای رفع و جلوگیری از این خطا از این تکه کد استفاده می کنیم تا آن هایی که مخالف Or هستند را در دیتافریم مورد بررسی خود نگه داریم.

Make columns correct type

```
In [7]: all_data['Quantity Ordered'] = pd.to_numeric(all_data['Quantity Ordered'])
all_data['Price Each'] = pd.to_numeric(all_data['Price Each'])
```

در این قسمت نوع هر ستون جدول را تنظیم می کنیم. مثلاً ستون قیمت هر محصول را از نوع داده عددی قرار می دهیم.

Augment data with additional columns

Add month column

```
In [8]: all_data['Month'] = all_data['Order Date'].str[0:2]
all_data['Month'] = all_data['Month'].astype('int32')
all_data.head()
```

```
Out[8]:
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
0	295665	Macbook Pro Laptop	1	1700.00	12/30/19 00:01	136 Church St, New York City, NY 10001	12
1	295666	LG Washing Machine	1	600.00	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

در این قسمت به دیتا فریم خود ستون ماه را اضافه کردیم.(خود دیتا فریم این ستون را نداشت).

Add a sales column

```
In [9]: all_data['Sales'] = all_data['Quantity Ordered'] * all_data['Price Each']
all_data.head()
```

```
Out[9]:
```

	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

در این قسمت از ضرب مقادیر ستون های قیمت هر محصول در تعداد هر محصول ستونی تحت عنوان فروش را ایجاد کردیم.

Add city column

```
In [10]: def get_city(address):  
         return address.split(",")[1].strip(" ")  
  
         def get_state(address):  
             return address.split(",")[2].split(" ")[1]  
  
         all_data['City'] = all_data['Purchase Address'].apply(lambda x: f"{get_city(x)} ({get_state(x)})")  
         all_data.head()
```

Out[10]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City
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 York City (NY)
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 York City (NY)
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 York City (NY)
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016	12	149.99	San Francisco (CA)
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301	12	11.95	Atlanta (GA)

در این قسمت ستون نام شهر را ایجاد کردیم. با استفاده از `apply()`

در این قسمت با توجه به سوالات طرح شده به دنبال پاسخ مناسب آن ها در میان دیتافریم می گردیم.

Data Exploration!

Question 1 : What was the best month for sales? How much was earned that month?

```
In [11]: all_data['Sales'] = all_data['Quantity Ordered'].astype('int') * all_data['Price Each'].astype('float')
```

```
In [12]: all_data.groupby(['Month']).sum(numeric_only=True)
```

```
Out[12]:
```

	Quantity Ordered	Price Each	Sales
Month			
1	10903	1811768.38	1822256.73
2	13449	2188884.72	2202022.42
3	17005	2791207.83	2807100.38
4	20558	3367671.02	3390670.24
5	18667	3135125.13	3152606.75
6	15253	2562025.61	2577802.26
7	16072	2632539.56	2647775.76
8	13448	2230345.42	2244467.88
9	13109	2084992.09	2097560.13
10	22703	3715554.83	3736726.88
11	19798	3180600.68	3199603.20
12	28114	4588415.41	4613443.34

با استفاده از `sum()` مجموع مقادیر فروش هر ماه را محاسبه کردیم و در آخر آن را به شکل جدول نمایش دادیم. با بررسی این 12 ردیف به این موضوع پی می بریم که ماه دوازدهم سال ، که نزدیک به سال نو نیز هست، بیشترین مقدار فروش را داشته.

در ادامه با استفاده از کتابخانه ی **matplotlib** به رسم نمودار برای نمایش و بررسی راحت تر می پردازیم.

Plt.bar() برای ایجاد نمودار میله ای استفاده می شود.

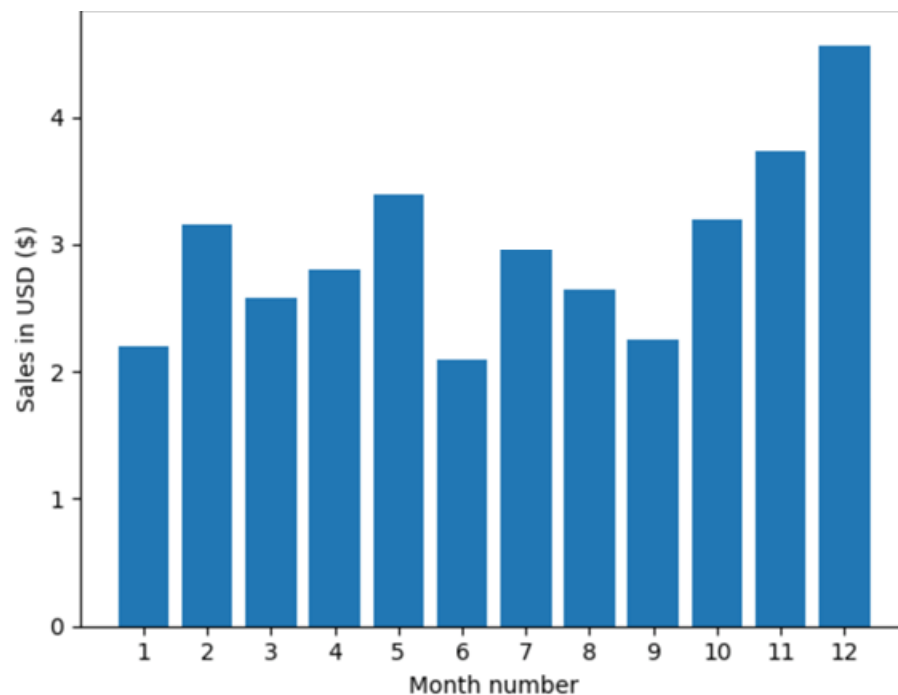
Plt.xticks() مکان های نشانگرهای روی محور **X** را تعیین می کند.

Plt.xlabel() نامی برای طول نمودار می دهیم.

Plt.ylabel() نامی برای عرض نمودار قرار می دهیم.

Plt.show() برای نمایش نمودار استفاده می کنیم.

```
In [20]: import matplotlib.pyplot as plt
months = all_data['Month'].unique()
plt.bar(months,all_data.groupby(['Month']).sum(numeric_only=True)['Sales'])
plt.xticks(months)
plt.ylabel('Sales in USD ($)')
plt.xlabel('Month number')
plt.show()
```



Question 2: What city had the highest number of sales?

```
In [17]: results = all_data.groupby('City').sum(numeric_only=True)
results
```

Out[17]:

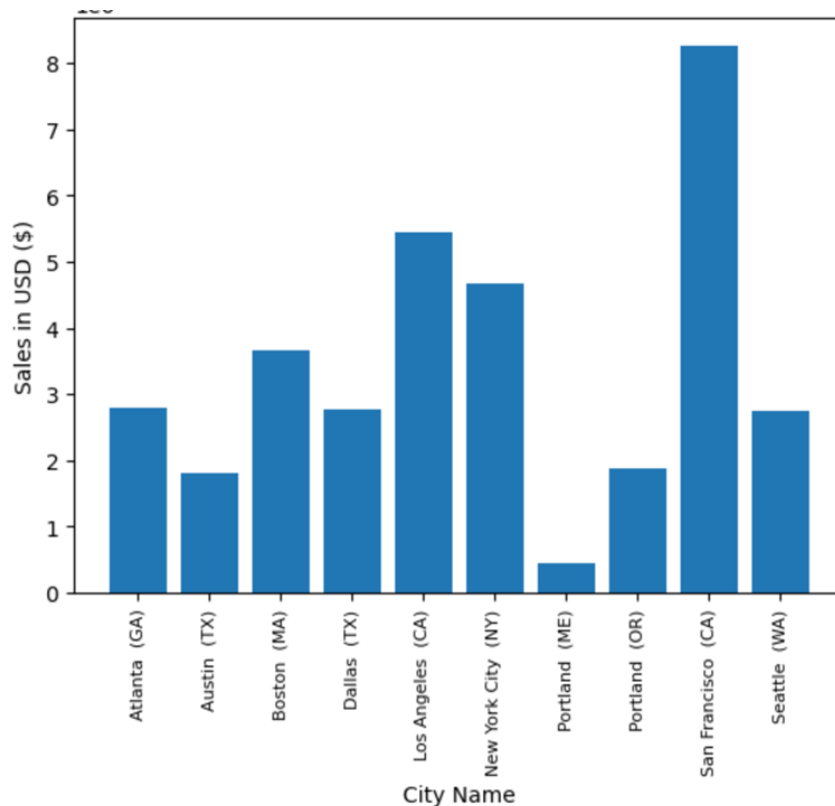
	Quantity Ordered	Price Each	Month	Sales
City				
Atlanta (GA)	16602	2779908.20	104794	2795498.58
Austin (TX)	11153	1809873.61	69829	1819581.75
Boston (MA)	22528	3637409.77	141112	3661642.01
Dallas (TX)	16730	2752627.82	104620	2767975.40
Los Angeles (CA)	33289	5421435.23	208325	5452570.80
New York City (NY)	27932	4635370.83	175741	4664317.43
Portland (ME)	2750	447189.25	17144	449758.27
Portland (OR)	11303	1860558.22	70621	1870732.34
San Francisco (CA)	50239	8211461.74	315520	8262203.91
Seattle (WA)	16553	2733296.01	104941	2747755.48

با استفاده از `sum()` میزان فروش هر شهر را محاسبه کرده و در آخر آن ها را مقایسه می کنیم.

```
In [16]: import matplotlib.pyplot as plt

keys = [city for city, df in all_data.groupby('City')]

plt.bar(keys, all_data.groupby(['City']).sum(numeric_only=True)['Sales'])
plt.ylabel('Sales in USD ($)')
plt.xlabel('City Name')
plt.xticks(keys, rotation='vertical', size=8)
plt.show()
```



سافرانسیکو بیشترین آمار فروش لوازم الکترونیکی را دارد طبق این نمودار.

سافرانسیکو به عنوان مرکز فناوری و شرکت های بزرگ تکنولوژی از جمله اپل، فیسبوک و گوگل شناخته می شود. در شهری که این شرکت های غول دنیا حضور دارند طبیعی است که مصرف وسایل الکترونیکی بالا تر باشد و خرید بیشتری در این زمینه صورت بگیرد.

Question 3: What time should we display advertisements to maximize likelihood of customer's buying product?

```
In [21]: # Add hour column
all_data['Hour'] = pd.to_datetime(all_data['Order Date']).dt.hour
all_data['Minute'] = pd.to_datetime(all_data['Order Date']).dt.minute
all_data['Count'] = 1
all_data.head()
```

```
Out[21]:
```

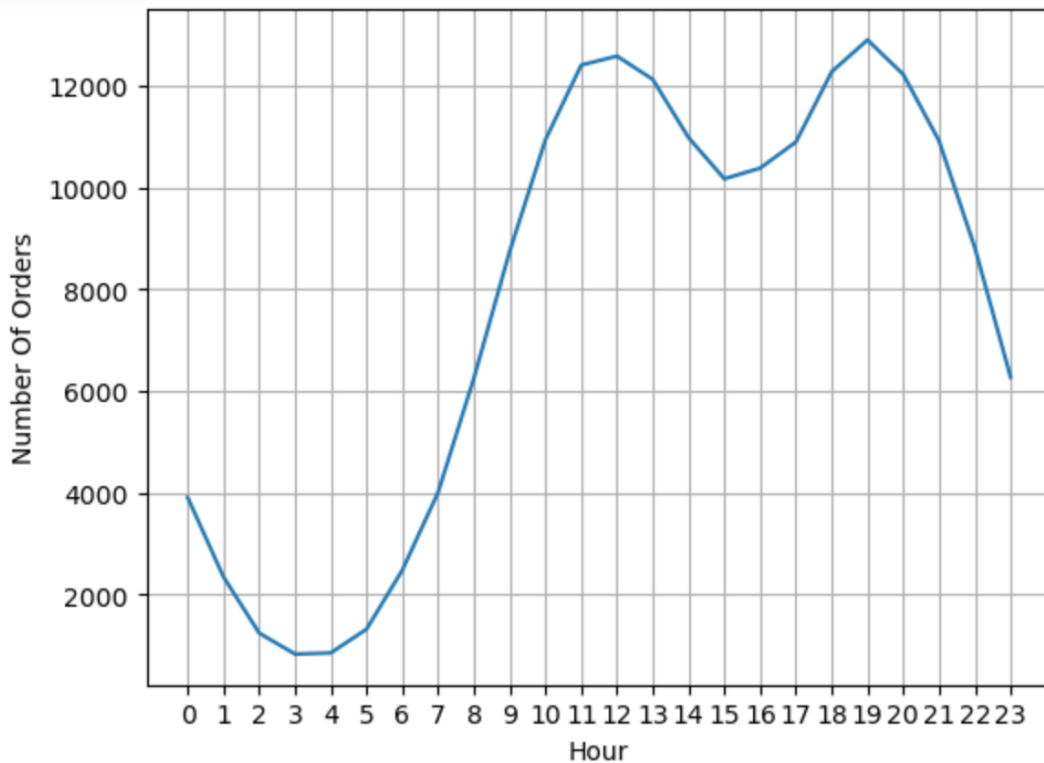
	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	Minute	Count
0	295665	Macbook Pro Laptop	1	1700.00	12/30/19 00:01	136 Church St, New York City, NY 10001	12	1700.00	New York City (NY)	0	1	1
1	295666	LG Washing Machine	1	600.00	12/29/19 07:03	562 2nd St, New York City, NY 10001	12	600.00	New York City (NY)	7	3	1
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 York City (NY)	18	21	1
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016	12	149.99	San Francisco (CA)	15	13	1
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301	12	11.95	Atlanta (GA)	12	38	1

با استفاده از `pd.to_datetime().dt.hour` و `pd.to_datetime().dt.minute` ستون مربوط به تاریخ سفارش را بر حسب ساعت و بر حسب دقیقه از نوع داده ی تاریخی جدا می کنیم و یک ستون جدول برای هر کدام اختصاص می دهیم.

```
In [23]: keys = [pair for pair, df in all_data.groupby('Hour')]

plt.plot(keys, all_data.groupby('Hour').count()['Count'])
plt.xticks(keys)
plt.xlabel('Hour')
plt.ylabel('Number Of Orders')
plt.grid()
plt.show()

# My recommendation is slightly before 11am or 7pm
```



دو قله ی نمودار یکی مربوط به ساعت 11الی 12 ظهر و دیگری مربوط به ساعت 7 شب می باشد.
پس با این اوصاف بهترین زمان برای نمایش تبلیغات و جذب مشتری به خرید در این ساعات است.

Question 4: What products are most often sold together?

```
In [69]: # https://stackoverflow.com/questions/43348194/pandas-select-rows-if-id-appear-several-time

df = all_data[all_data['Order ID'].duplicated(keep=False)]

# Referenced: https://stackoverflow.com/questions/27298178/concatenate-strings-from-several-rows-using-pandas-groupb
df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x: ','.join(x))
df = df[['Order ID', 'Grouped']].drop_duplicates()

df.head()
```

Out [69]:

	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

```
In [70]: # Referenced: https://stackoverflow.com/questions/52195887/counting-unique-pairs-of-numbers-into-a-python-dictionary

from itertools import combinations
from collections import Counter

count = Counter()

for row in df2['Grouped']:
    row_list = row.split(',')
    count.update(Counter(combinations(row_list, 2)))

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
```

در این قسمت به جدول خود ستونی تحت عنوان Grouped اضافه کردیم که نشان دهنده ی محصولات ی است که همزمان با هم توسط خریدار خریداری شده اند.

با استفاده از Counter تعداد دفعاتی که هر کدام از این محصولات با هم خریداری شده اند شمرده می شود.
با استفاده از print لیست و تعداد تکرار هر کدام نشان داده شده است.

What product sold the most? Why do you think it sold the most?

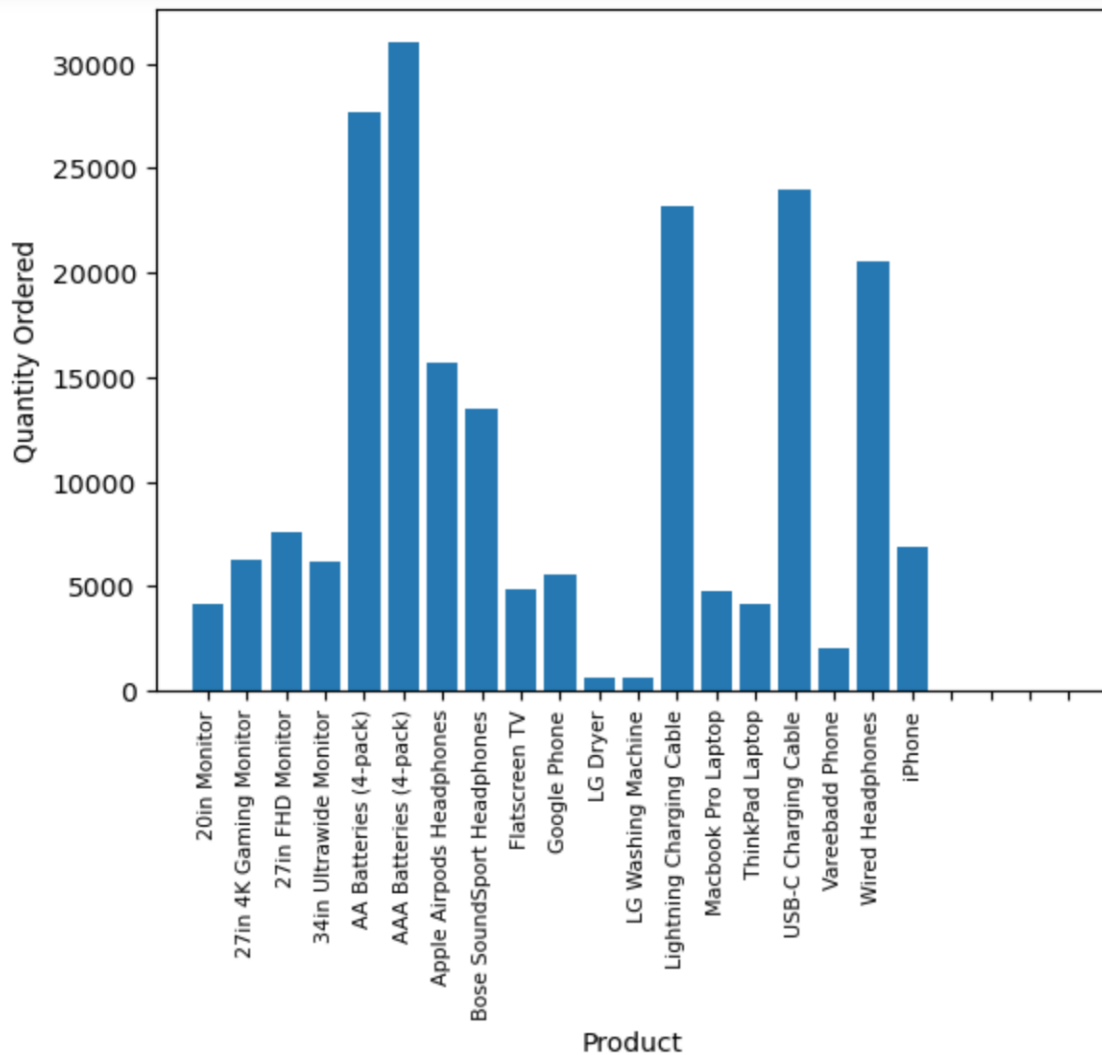
In [71]: all_data.head()

Out[71]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	Minute	Count
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 York City (NY)	0	1	1
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 York City (NY)	7	3	1
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 York City (NY)	18	21	1
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016	12	149.99	San Francisco (CA)	15	13	1
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301	12	11.95	Atlanta (GA)	12	38	1

```
In [25]: product_group = all_data.groupby('Product')
quantity_ordered = product_group.sum(numeric_only=True)['Quantity Ordered']

products = [product for product, df in product_group]
plt.bar(products, quantity_ordered)
plt.xticks(keys, rotation='vertical', size=8)
plt.xlabel('Product')
plt.ylabel('Quantity Ordered')
plt.show()
```



AAA Batteries (4-pack) بیشترین محصول الکترونیکی فروخته شده در این فروشگاه است.

دلایل قابل ذکر برای پرفروش بودن این باتری ها عبارتند از :

پراستفاده ترین اندازه ی باتری هستند و در ریموت کنترل، ساعت ها ، اسباب بازی ها ، کنترل های بازی و ... استفاده می شوند. گستردگی تنوع دستگاه های قابل استفاده از این باتری بالاست. همچنین قیمت مقرون به صرفه ای دارند.

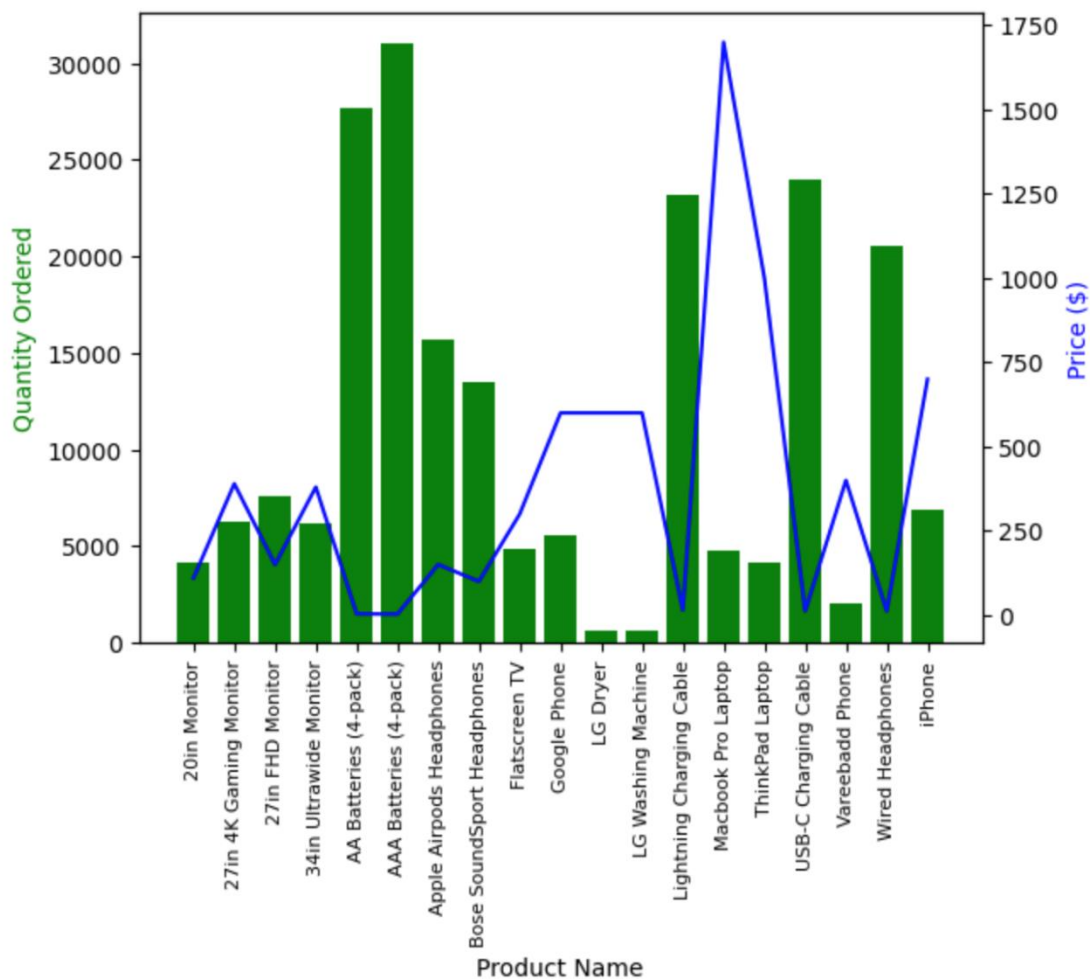
```
In [32]: # Referenced: https://stackoverflow.com/questions/14762181/adding-a-y-axis-label-to-secondary-y-axis-in-matplotlib
import matplotlib.pyplot as plt

prices = all_data.groupby('Product').mean(numeric_only=True)['Price Each']
fig, ax1 = plt.subplots()

ax2 = ax1.twinx()
ax1.bar(products, quantity_ordered, color='g')
ax2.plot(products, prices, color='b')

ax1.set_xlabel('Product Name')
ax1.set_ylabel('Quantity Ordered', color='g')
ax2.set_ylabel('Price ($)', color='b')
ax1.set_xticks(range(len(products)))
ax1.set_xticklabels(products, rotation='vertical', size=8)

fig.show()
```



از این نمودار نیز نتیجه می شود که اصولاً لوازمی که قیمت بالایی دارند تعداد دفعات کمتری به فروش می روند ولی لوازم ارزان قیمت تعداد دفعات فروش بیشتری را شامل می شوند. از طرفی یک فرد یک بار لپتاپ می خرد اما ممکن است چندین باتری تریپل ای را نیاز داشته باشد و حتی همراه باطری بخرد ولی همراه لپتاپ جدید نمی خرد.