BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM

KHOA CÔNG NGHỆ THÔNG TIN

☆☆☆☆☆

HCMUTE

MÔN HỌC: TƯƠNG TÁC DỮ LIỆU TRỰC QUAN

(INTERACTIVE DATA VISUALIZATION)

BÁO CÁO ĐỒ ÁN CUỐI KỲ

NỘI DUNG: Phân tích và trực quan hóa tập dữ liệu“ Sales Product Data”

GVHD: ThS. Lê Minh Tân

SVTH:	**MSSV**
Nguyễn Thanh Bình	20133025
Nguyễn Nhật Triều	20133102
Đoàn Quốc Trung	20133104

Mã lớp: IDVI333677_22_2_02

Tp.Hồ Chí Minh, tháng 04 năm 2023

ĐH SƯ PHẠM KỸ THUẬT TP.HCM
KHOA CNTT
****

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh Phúc
****

PHIẾU CHẤM ĐIỂM BÁO CÁO

I. Thông tin

Tên nhóm: Nhóm 06 **Số thứ tự nhóm:** 06

Thành viên nhóm:

- Nguyễn Thanh Bình 20133025
- Nguyễn Nhật Triều 20133102
- Đoàn Quốc Trung 20133104

Mã số trưởng nhóm: 20133104 **Lớp:** 201332B **Khóa:** K20

Tên đề tài cuối kỳ: Phân tích và trực quan hóa tập dữ liệu**“** ***Sales Product Data”***

II. Phân công

STT	Mã số TV	Công việc	Nội dung	Điểm cá nhân	TV ký tên
1	20133025	Viết báo cáo, tìm hiểu về tập dữ liệu		…/10	
2	20133102	Tìm hiểu về tập dữ liệu vẽ biểu đồ		…/10	
3	20133104	Tìm hiểu về tập dữ liệu, vẽ biểu đồ		…/10	

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin được bày tỏ lòng biết ơn sâu sắc đến tất cả thầy cô đại học Sư Phạm Kỹ Thuật Thành phố Hồ Chí Minh đã tạo điều kiện tốt nhất, hỗ trợ, giúp đỡ chúng em trong suốt quá trình học tập vừa qua.

Đồng thời, chúng em xin chân thành gởi lời cảm ơn sâu sắc đến Ban Chủ nhiệm và các thầy cô trong khoa Công nghệ Thông tin - trường Đại học Sư phạm Kỹ Thuật Thành Phố Hồ Chí Minh đã tạo môi trường học tập tốt nhất để chúng em có thể thuận lợi tiếp thu những kiến thức mới.

Đặc biệt, chúng em xin chân thành gửi lời cảm ơn tới thầy ThS. Lê Minh Tân -giáo viên giảng dạy, người đã trực tiếp giúp đỡ, chỉ dẫn tận tình, hướng dẫn nhóm hoàn thành bài báo cáo đồ án cuối kỳ môn "Tương tác dữ liệu trực quan" trong thời gian qua.

Trong quá trình hoàn thành, nhóm đã có những cố gắng nhất định, song do kinh nghiệm và kiến thức còn nhiều thiếu sót, dẫn đến bài báo cáo còn nhiều hạn chế nhất định. Vì vậy, nhóm chúng em kính mong nhận được những phản hồi, đóng góp ý kiến và chỉ bảo thêm từ quý thầy cô nói chung, cũng như thầy ThS. Lê Minh Tân nói riêng để nhóm có thể rút ra được nhiều bài học quý giá từ đó để áp dụng vào cuộc sống sau này.

Chúng em xin chân thành cảm ơn!

MỤC LỤC

Mục lục hình ảnh

A. PHẦN MỞ ĐẦU

1. Lý do chọn đề tài

Bán lẻ là một lĩnh vực kinh doanh rất quan trọng trong nền kinh tế của một quốc gia. Trong lĩnh vực này, các cửa hàng bán lẻ luôn đóng vai trò quan trọng trong việc cung cấp hàng hóa và dịch vụ cho khách hàng. Hiện nay, việc bán lẻ đang ngày một phát triển lớn mạnh, điều đó được thể hiện rõ khi ngày một nhiều cửa hàng bán lẻ khai trương. Tuy nhiên, không phải việc khai trương một cửa hàng bán lẻ thì sẽ được nhiều người quan tâm và mua mà phải có sự hiểu biết sâu sắc về hành vi mua sắm của khách hàng và cách hoạt động của thị trường bán lẻ.

Trong bối cảnh đó, việc phân tích dữ liệu bán hàng của một cửa hàng bán lẻ sẽ là một công cụ hữu ích để giúp các nhà quản lý cửa hàng hiểu rõ hơn về hoạt động của cửa hàng, từ đó đưa ra những quyết định kinh doanh thông minh hơn.

Chính vì vậy, nhóm chúng em quyết định chọn đề tài“ *Nghiên cứu về quản lý kinh doanh và phân tích dữ liệu bán hàng của một cửa hàng bán lẻ tại Mỹ năm 2019*” làm đề tài báo cáo đồ án cuối kỳ môn này.

2. Mục đích của đề tài

2.1. Mục đích

Áp dụng các kiến thức của môn tương tác dữ liệu đã học vào tập dữ liệu “Sales Product Data” được lấy từ hệ thống quản lý kho và bán hàng của một cửa hàng bán lẻ tại Mỹ năm 2019 nhằm phân tích doanh số bán hàng của cửa hàng đó.

2.2. Mục tiêu

- Mục tiêu 1: Nắm được các công nghệ sử dụng trong bài phân tích như: python, apache zeppelin,…

- Mục tiêu 2: Hiểu rõ được dữ liệu đồ thị, các tính chất, đặc trưng và ý nghĩa của từng biểu đồ.
- Mục tiêu 3: Nắm được các kiến thức của môn“ tương tác dữ liệu trực quan” đã học.
- Mục tiêu 4: Năm được các bước tiền xử lý dữ liệu để xử lý dữ liệu thô sang dữ liệu có thể vẽ được biểu đồ.

3. Phương pháp nghiên cứu

3.1. Về mặt lý thuyết

- Các tài liệu về môn tương tác dữ liệu trực quan trên lớp cũng như trên internet.
- Các tài liệu liên quan đến lập trình python.
- Các tài liệu liên quan đến xử lý dữ liệu đồ thị Matplotlib, Seaborn,…

3.2. Đối tượng nghiên cứu

- Ngôn ngữ python và các thư viện xử lý dữ liệu đồ thị.
- Tập dữ liệu“ Sales Product Data” được lấy từ trang mạng“ https://www.kaggle.com/”.

3.3. Phạm vi nghiên cứu

Nghiên cứu các kiến thức của môn tương tác dữ liệu, các kiến thức liên quan đến python để áp dụng cho tập dữ liệu.

4. Kết quả dự kiến đạt được

- Về mặt lý thuyết:

  + Hiểu rõ dữ liệu đồ thị, các tính chất, đặc trưng và ý nghĩa của đồ thị.

  + Áp dụng được các kiến thức môn học vào bài báo cáo.
- Về mặt ứng dụng: Phân tích, đánh giá được doanh số bán hàng của một của hàng bán lẻ tại Mỹ năm 2019.

B. PHẦN NỘI DUNG

I. Giới thiệu về tập dữ liệu

1. Nguồn gốc

Kaggle là một nền tảng để các nhà khoa học dữ liệu và những người thực hành máy học chia sẻ và làm việc trong các dự án khoa học dữ liệu. Đây là một nguồn tài nguyên tuyệt vời để tìm hiểu về khoa học dữ liệu và máy học cũng như để tìm kiếm các bộ dữ liệu để sử dụng trong các dự án của riêng chúng ta. Chính vì vậy, tập dữ liệu“ Sales Product Data” được lấy từ trang kaggle.

Link tập dữ liệu: https://www.kaggle.com/datasets/knightbearr/sales-product-data

Ngoài ra, vì để vẽ biểu đồ choropleth, nhóm còn sử dụng thêm 2 tập dữ liệu nữa là:

- Tập dữ liệu“ 2019 Census US Population Data By State” là một bảng dữ liệu về dân số, cũng như kinh độ và vĩ độ của các bang tại Hoa Kỳ trong năm 2019.[1]
- Tập dữ liệu“ USA State code” là một bảng dữ liệu bao gồm mã của 50 bang của Hoa Kỳ, bao gồm cả bang và vùng lãnh thổ khác, cùng với các thông tin khác như tên và tên ngắn của bang, vùng lãnh thổ.[2]

2. Mô tả tập dữ liệu

Tập dữ liệu“ Sales Product Data” là tập hợp dữ liệu về doanh số bán hàng từ một nhà bán lẻ. Tập dữ liệu chứa dữ liệu bán hàng trong 12 tháng, từ tháng 1 năm 2019 đến tháng 12 năm 2019. Tập dữ liệu bao gồm 12 tệp định dạng csv tương ứng với mỗi tệp là một tháng trong năm 2019 với hơn

[1] https://www.kaggle.com/datasets/peretzcohen/2019-census-us-population-data-by-state?select=2019_Census_US_Population_Data_By_State_Lat_Long.csv
[2] https://www.kaggle.com/datasets/corochann/usa-state-code

186,000 bản ghi và mỗi tệp gồm 5 cột nhưng trong đó một số cột có giá trị null. Các thông tin trong mỗi tệp bao gồm ID đặt hàng, tên sản phẩm, số lượng đặt hàng, giá bán, ngày bán hàng, địa chỉ giao hàng.Tập dữ liệu này có thể được sử dụng để phân tích xu hướng bán hàng, đánh giá hiệu quả của chiến dịch quảng cáo và phát hiện các cơ hội kinh doanh mới.

II. Giới thiệu về công nghệ sử dụng

1. Python

Python là một ngôn ngữ lập trình bậc cao, đa năng và dễ học. Python có cú pháp đơn giản, dễ hiểu, có nhiều thư viện và framework hỗ trợ mạnh mẽ cho việc phân tích dữ liệu, xử lý ảnh, và xây dựng các ứng dụng web.

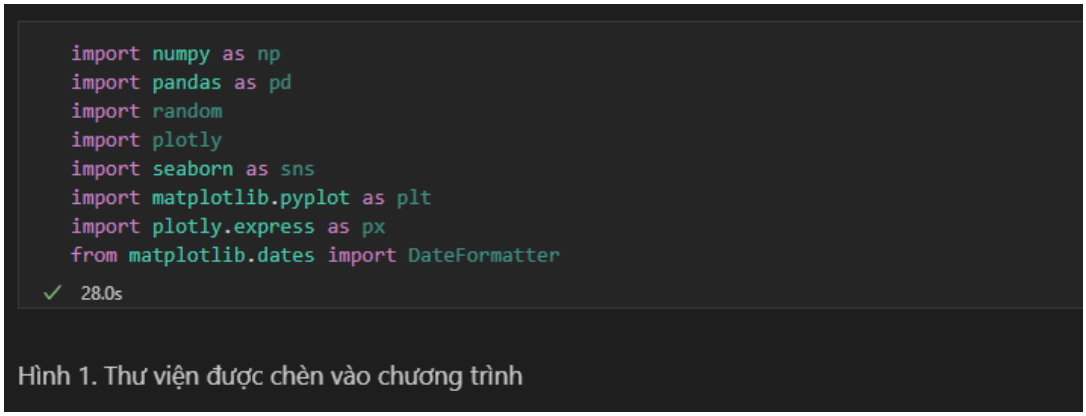
2. Apache zepplin

Apache Zeppelin là một công cụ phân tích và trình diễn dữ liệu mã nguồn mở, cho phép các nhà phân tích dữ liệu và nhà khoa học dữ liệu tương tác với dữ liệu. Nó cung cấp cho người dùng khả năng lập trình tương tác, trực quan hóa dữ liệu và chia sẻ kết quả với cộng đồng.

3. Một số thư viện

- NumPy là một thư viện Python hỗ trợ các phép toán trên ma trận, mảng và nhiều loại tính toán khoa học khác.
- Pandas là một thư viện Python được sử dụng rộng rãi cho phân tích dữ liệu và xử lý dữ liệu dạng bảng.
- Random là một thư viện Python được sử dụng để sinh số ngẫu nhiên hoặc lựa chọn các phần tử ngẫu nhiên từ một danh sách.
- Plotly là một thư viện trực quan hóa dữ liệu cho Python, cung cấp các công cụ để tạo ra các biểu đồ tương tác đẹp mắt.

- Seaborn là một thư viện trực quan hóa dữ liệu Python dựa trên Matplotlib, cung cấp các chức năng cải thiện đáng kể của trực quan hóa dữ liệu.
- Matplotlib là một thư viện trực quan hóa dữ liệu Python phổ biến, cung cấp các công cụ để tạo ra các biểu đồ tĩnh và động.

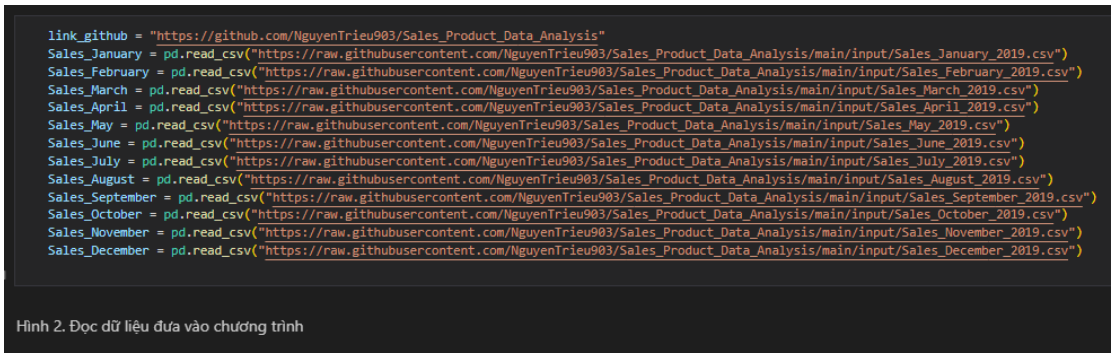


Hình 1. Thư viện được chèn vào chương trình

III. Cách xử lý dữ liệu

1. Đọc dữ liệu vào chương trình

Ở đây, sau khi tải tập dữ liệu từ kaggle về, chúng em đã tiến hành tải lên github. Sau đó, tiến hành dùng thư viện pandas để đọc dữ liệu từ github về chương trình với mỗi biến tương ứng với một tệp csv(nghĩa là mỗi biến tương ứng với một tháng trong tập dữ liệu).

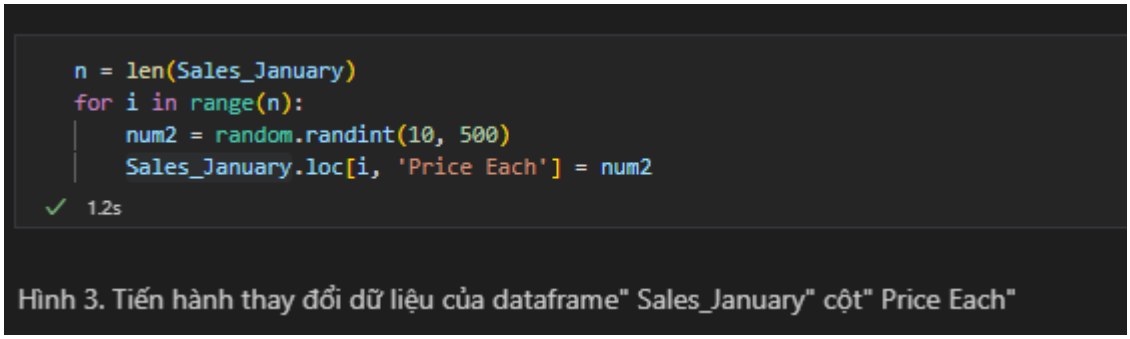


Hình 2. Đọc dữ liệu đưa vào chương trình

2. Thực hiện yêu cầu tốc độ cập nhật dữ liệu của chương trình

Ở đây, để thực hiện yêu cầu của bài báo cáo là có sự thay đổi dữ liệu và vẽ lại biểu đồ nên nhóm đã thực hiện bằng cách sử dụng hàm len() để tính

độ dài của dataframe "Sales_January" và lưu vào biến n. Tiếp đó, trong vòng lặp for từ 0 đến n-1, sử dụng hàm randint() của thư viện random để tạo ngẫu nhiên một số nguyên từ 10 đến 500, và gán giá trị đó vào cột "Price Each" của hàng thứ i trong "Sales_January".

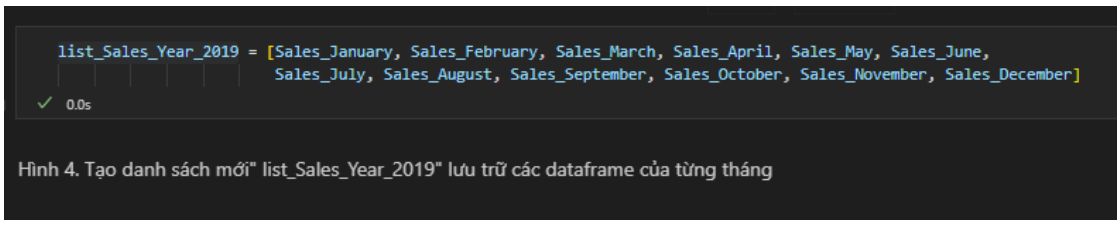


Hình 3. Tiến hành thay đổi dữ liệu của dataframe" Sales_January" cột" Price Each"

Lưu ý: Vì trong chương trình demo, có thêm đoạn mã này nên khi chạy chương trình sẽ có thể có kết quả khác so với bài báo cáo này. Còn bài báo cáo này kết quả cho ra được lấy từ dữ liệu gốc ban đầu trong trên kaggle.

3. Tạo ra một dataframe lưu trữ các dataframe của từng tháng

Ở đây, chúng em tiến hành tạo ra một danh sách gồm các bảng dữ liệu của 12 tháng năm 2019(Sales_January đến Sales_December) được lưu trữ trong các biến tương ứng, giúp cho việc xử lý và truy cập dữ liệu của từng tháng dễ dàng hơn, đồng thời cũng giúp cho việc thực hiện các phân tích và trực quan hóa dữ liệu được dễ dàng và hiệu quả hơn.



Hình 4. Tạo danh sách mới" list_Sales_Year_2019" lưu trữ các dataframe của từng tháng

Sau đó tiến hành in danh sách vừa tạo:

```
print(list_Sales_Year_2019)
✓ 0.1s
Output exceeds the size limit. Open the full output data in a text editor
[     Order ID                  Product Quantity Ordered Price Each  \
0       141234                    iPhone                1        346
1       141235  Lightning Charging Cable                1        112
2       141236          Wired Headphones                2        168
3       141237        27in FHD Monitor                  1        179
4       141238          Wired Headphones                1         99
...        ...                       ...              ...        ...
9718    150497              20in Monitor                1        119
9719    150498        27in FHD Monitor                  1         15
9720    150499           ThinkPad Laptop                1         53
9721    150500    AAA Batteries (4-pack)                2         78
9722    150501              Google Phone                1        161

          Order Date                      Purchase Address
0     01/22/19 21:25        944 Walnut St, Boston, MA 02215
1     01/28/19 14:15       185 Maple St, Portland, OR 97035
2     01/17/19 13:33  538 Adams St, San Francisco, CA 94016
3     01/05/19 20:33     738 10th St, Los Angeles, CA 90001
4     01/25/19 11:59          387 10th St, Austin, TX 73301
...             ...                                   ...
9718  01/26/19 19:09            95 8th St, Dallas, TX 75001
9719  01/10/19 22:58    403 7th St, San Francisco, CA 94016
9720  01/21/19 14:31        214 Main St, Portland, OR 97035
9721  01/15/19 14:21      810 2nd St, Los Angeles, CA 90001
9722  01/13/19 16:43         428 Cedar St, Boston, MA 02215
...
25115  12/03/19 10:39          778 River St, Dallas, TX 75001
25116  12/21/19 21:45  747 Chestnut St, Los Angeles, CA 90001

[25117 rows x 6 columns]]
```

Hình 5. In ra danh sách" list_Sales_Year_2019" vừa tạo

4. Tính tổng số hàng và cột của danh sách vừa tạo

Ở đây, nhóm đã tạo một DataFrame mới với tên là“ total_sales_2019”. Sau đó, sử dụng vòng lặp để liên kết các bảng dữ liệu (tương ứng với các tháng trong năm) trong“ list_Sales_Year_2019” vào DataFrame total_sales_2019 bằng hàm concat của Pandas. Kết quả là tất cả các bảng dữ liệu của các tháng đã được gộp lại thành một bảng dữ liệu lớn đại diện cho toàn bộ năm 2019. Cuối cùng, hàm shape được sử dụng để hiển thị kích thước của bảng dữ liệu“ total_sales_2019”, bao gồm số hàng và số cột.

```
total_sales_2019 = pd.DataFrame()
for sales_month in list_Sales_Year_2019:
    total_sales_2019 = pd.concat([total_sales_2019, sales_month])

total_sales_2019.shape
```

✓ 0.1s

```
(186849, 6)
```

Hình 6. Tính tổng số hành và cột của danh sách vừa tạo" list_Sales_Year_2019"

5. Xem thông tin từng cột

Ở đây, nhóm đã sử dụng hàm info() để tiến hành xem thông tin của từng cột trong DataFrame“ total_sales_2019” vừa mới tạo ở trên.

```
total_sales_2019.info()
```

✓ 0.8s

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 186849 entries, 0 to 25116
Data columns (total 6 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        186331 non-null  object
 4   Order Date        186305 non-null  object
 5   Purchase Address  186305 non-null  object
dtypes: object(6)
memory usage: 10.0+ MB
```

Hình 7. Xem thông tin của từng cột

6. Kiểm tra các hàng có giá trị Null

Ở đây, vì để thuận lợi khi vẽ biểu đồ và tính toán, nhóm đã tiến hành tính tổng số giá trị Null của từng cột trong DataFrame“ total_sales_2019” bằng cách kết hợp 2 hàm isnull() và hàm sum().

```
total_sales_2019.isnull().sum()
```
✓ 0.2s

```
Order ID            544
Product             544
Quantity Ordered    544
Price Each          544
Order Date          544
Purchase Address    544
dtype: int64
```

Hình 8. Tính tổng số hàng có giá trị Null của từng cột

Theo như kết quả mà chương trình trả về thì tất cả các cột đều chứa 544 hàng có giá trị Null. Chính vì vậy, nhóm tiến hành sử dụng hàm dropna() trong thư viện Pandas để loại bỏ các hàng có chứa giá trị null hoặc NaN (Not a Number). Và kết quả sau đó là:

```
total_sales_2019 = total_sales_2019.dropna()
total_sales_2019.isnull().sum()
```
✓ 0.7s

```
Order ID            0
Product             0
Quantity Ordered    0
Price Each          0
Order Date          0
Purchase Address    0
dtype: int64
```

Hình 9. Loại bỏ các hàng có chứa giá trị Null trong DataFrame" total_sales_2019"

```
total_sales_2019.shape
```
✓ 0.0s

```
(186305, 6)
```

Hình 10. Tổng số hàng sau khi loại bỏ các hàng chứa giá trị Null

7. Kiểm tra các giá trị trong cột“ Quantity Ordered”

Ở đây, nhóm tiến hành in danh sách các giá trị duy nhất trong cột“ Quantity Ordered” của DataFrame“ total_sales_2019”bằng cách dùng hàm unique().

```
total_sales_2019['Quantity Ordered'].unique()
```
✓ 0.1s
```
array(['1', '2', '3', '5', '4', '7', 'Quantity Ordered', '6', '9', '8'],
      dtype=object)
```

Hình 11. Hiển thị danh sách các giá trị duy nhất trong cột "Quantity Ordered"

Và ta thấy rằng trong cột“ Quantity Ordered” chứa giá trị không phải số là “ Quantity Ordered”. Do đó, nhóm tiến hành loại bỏ giá trị đó ra khỏi cột“ Quantity Ordered” bằng cách dùng biến filter có giá trị là một Series Boolean.

```
filter = total_sales_2019['Quantity Ordered'] == 'Quantity Ordered'
total_sales_2019 = total_sales_2019[~filter]

total_sales_2019['Quantity Ordered'].unique()
```
✓ 0.2s
```
array(['1', '2', '3', '5', '4', '7', '6', '9', '8'], dtype=object)
```

Hình 12. lọc và xử lý dữ liệu trong cột" Quantity Ordered" có chưa giá trị" Quantity Ordered"

```
total_sales_2019.shape
```
✓ 0.0s
```
(185950, 6)
```

Hình 13. Số hàng còn lại của sau khi loại bỏ các hàng có chứa giá trị" Quantity Ordered" trong cột" Quantity Ordered"

8. Chuyển đổi kiểu dữ liệu

Ở đây, vì các cột“ Quantity Ordered”,“ Price Each”,“ Order Date” cần thiết để vẽ biểu đồ, nhưng kiểu dữ liệu của các cột này đang ở kiểu chuỗi(

string) nên nhóm đã tiến hành chuyển đổi kiểu dữ liệu của các cột này như sau: chuyển đổi kiểu dữ liệu của cột“ Quantity Ordered” từ chuỗi (string) sang số nguyên (integer), chuyển đổi kiểu dữ liệu của cột“ Price Each” từ chuỗi sang số thực (float), chuyển đổi kiểu dữ liệu của cột“ Order Date” từ chuỗi sang định dạng ngày-tháng (datetime), sử dụng hàm to_datetime() từ thư viện Pandas.

```
total_sales_2019['Quantity Ordered'] = total_sales_2019['Quantity Ordered'].astype(int)
total_sales_2019['Price Each'] = total_sales_2019['Price Each'].astype(float)
total_sales_2019['Order Date'] = pd.to_datetime(total_sales_2019['Order Date'])
```

✓ 29.5s

Hình 14. Chuyển đổi kiểu dữ liệu của các cột" Quantity Ordered"," Price Each"," Order Date"

Kết quả sau khi chuyển đổi:

```
total_sales_2019.dtypes
```

✓ 0.0s

```
Order ID                    object
Product                     object
Quantity Ordered             int32
Price Each                 float64
Order Date          datetime64[ns]
Purchase Address            object
dtype: object
```

Hình 15. Kết quả sau khi chuyển đổi kiểu dữ liệu của các cột" Quantity Ordered"," Price Each"," Order Date"

9. Thêm cột State và cột Sales

Ở đây, vì cần vẽ bản đồ địa lý nên nhóm đã tiên hành thêm một cột mới“ State”, nội dung của cột“ State” được tạo ra bằng cách cắt chuỗi ở mỗi phần từ của cột“ Purchase Address” tại dấu phẩy thứ 3, sau đó lấy phần tử thứ 2 sau dấu cách để lấy mã bang (VD: "CA" cho California).

Tiếp theo, tạo một cột mới Sales bằng cách nhân số lượng đơn hàng“ Quantity Ordered” với giá của mỗi sản phẩm“ Price Each” trong bảng “total_sales_2019”.

```
def state(x):
    state = x.split(',')[2]
    return state.split(' ')[1]
total_sales_2019['State'] = total_sales_2019['Purchase Address'].apply(state)
total_sales_2019['Sales'] = total_sales_2019['Quantity Ordered'] * total_sales_2019['Price Each']
total_sales_2019.head()
```

✓ 0.3s

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	State	Sales
0	141234	iPhone	1	346.0	2019-01-22 21:25:00	944 Walnut St, Boston, MA 02215	MA	346.0
1	141235	Lightning Charging Cable	1	112.0	2019-01-28 14:15:00	185 Maple St, Portland, OR 97035	OR	112.0
2	141236	Wired Headphones	2	168.0	2019-01-17 13:33:00	538 Adams St, San Francisco, CA 94016	CA	336.0
3	141237	27in FHD Monitor	1	179.0	2019-01-05 20:33:00	738 10th St, Los Angeles, CA 90001	CA	179.0
4	141238	Wired Headphones	1	99.0	2019-01-25 11:59:00	387 10th St, Austin, TX 73301	TX	99.0

Hình 16. DataFrame" total_sales_2019" sau khi thêm 2 cột" State" và" Sales"

10. Tính tổng doanh thu từng tháng

Ở đây, nhóm đã tiến hành tính tổng doanh thu từng tháng bằng cách:

- Đầu tiên, tạo ra cột“ Month” từ cột“ Order Date" trong bảng“ total_sales_2019" bằng phương thức“ dt.month”.
- Sau đó, nhóm dữ liệu theo cột“ Month” và tính tổng cột“ Sales” sử dụng hàm“ sum()”.
- Cuối cùng, chuyển đổi các kết quả có được sang DataFrame

Trên hình 17 bên dưới, đoạn code cuối cùng xuất ra kết qua vừa tính được của 3 tháng là 4, 5, 6.

```
total_sales_2019['Month'] = total_sales_2019['Order Date'].dt.month
Sales_Month = total_sales_2019.groupby('Month')['Sales'].sum()
Sales_Month = Sales_Month.to_frame()
Sales_Month[3:6]
```

✓ 0.1s

Month	Sales
4	3390670.24
5	3152606.75
6	2577802.26

Hình 17. Tính tổng doanh thu từng tháng

IV. Vẽ biểu đồ

1. Biểu đồ cột thể hiện tổng doanh số bán hành theo từng tháng

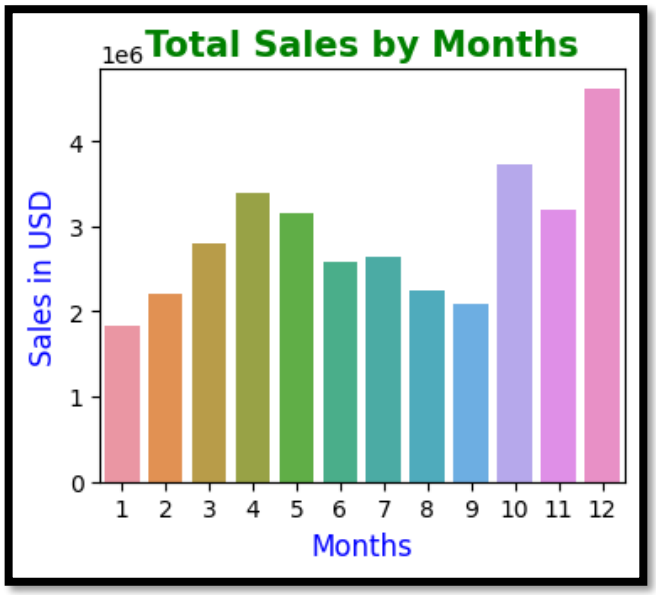
a) Cách vẽ

Ở đây, nhóm tiến hành vẽ biểu đồ bar chart(biểu đồ cột) thể hiện tổng doanh doanh số bán hàng theo từng tháng trong năm 2019 của cửa hàng bằng cách dùng hàm sns.barplot() của thư viện Seaborn với trục y là tổng doanh số bán hàng(cột“ Sales” của DataFrame“ Sales_Month” vừa tính ở trên- hình 17), trục x là tháng- dữ liệu đầu vào là Sales_Month.

```
fig, ax = plt.subplots(figsize=(4, 3.55))
sns.barplot(
    y=Sales_Month['Sales'],
    x=Sales_Month.index,
    data=Sales_Month,
    ax=ax)
plt.tight_layout(rect=[0.001, 0.01, 0.95, 0.95])
plt.title('Total Sales by Months',fontsize =15,color='green', fontweight='bold')
plt.ylabel('Sales in USD',fontsize =12,color='blue')
plt.xlabel('Months',fontsize =12,color='blue')
```

Hình 18. Đoạn mã để vẽ biểu đồ cột thể hiện tổng doanh số bán hàng theo từng tháng

b) Kết quả



Hình 19. Biểu đồ cột thể hiện tổng doanh số bán hàng theo từng tháng

c) Nhận xét

Từ biểu đồ nhận được, nhóm có các nhận xét sau về biểu đồ:

- Có thể thấy rằng, tháng 12 là tháng có số lượng sản phẩm bán ra nhiều nhất trong năm, tiếp đến là tháng 10 và tháng 4.
- Trong khi đó, tháng 1 là tháng có số lượng sản phẩm bán ra thấp nhất.
- Nhìn vào biểu đồ, từ tháng 1 đến tháng 12, số lượng sản phẩm bán ra có sự biến độ rõ rệt, cụ thể: từ tháng 1 – 4, số lượng sản phẩm bán bán ra có sự gia tăng; từ tháng 4 – 9 lại có sự giảm số lượng sản phẩm bán ra; nhưng sau đó 3 tháng cuối(10- 12) có sự tăng trương vượt bậc về số lượng sản phẩm bán ra(nằm trong top 5 tháng có số lượng sản phẩm bán ra nhiều nhất) và cuối cùng đạt đỉnh điểm vào tháng 12; sau tháng 12 thì số lượng sản phẩm bán ra lại có sự tụt giảm nghiêm trọng.
- Tóm lại, điều này có thể cho thấy sự tăng trưởng của doanh số bán hàng trong giai đoạn cuối năm, có thể do tác động của các dịp lễ và kỳ nghỉ, trong khi đó các tháng đầu năm có thể có sự giảm sút doanh số do có ít các dịp lễ và kỳ nghỉ hơn.

2. Biểu đồ cột thể hiện 10 sản phẩm bán chạy nhất

a) Cách vẽ

Để vẽ biểu đồ, ta thực hiện các bước sau:

- Đầu tiên, nhóm tạo một DataFrame mới với tên“ product_order” có nội dung bên trong bao gồm tổng số lượng đơn hàng được đặt mua cho từng sản phẩm, sắp xếp theo thứ tự giảm dần và giới hạn chỉ lấy 10 sản phẩm đứng đầu.
- Sau đó, nhóm sử dụng hàm sns.barplot() của thư viện Seaborn với trục y là tên các sản phẩm và trục x là tổng số lượng đơn hàng được đặt mua cho từng sản phẩm của DataFrame“ product_order” để vẽ biểu đồ.

```
product_order = total_sales_2019.groupby('Product').count().sort_values(by='Quantity Ordered', ascending=False)[:10]
plt.figure(figsize=(4.5,3.8))

sns.barplot(
    x=product_order['Quantity Ordered'],
    y=product_order.index,
    data=product_order,
    palette='pastel',
    linewidth=1)

plt.tight_layout(rect=[0.05, 0.05, 0.85, 0.95])
plt.title('Top 10 Best Selling Products', fontsize=15, pad=10, loc='right', fontweight='bold', color='green')
plt.xlabel('Total Quantity Ordered',fontsize =12,color='blue')
plt.ylabel('Products',fontsize =12,color='blue')

for i, v in enumerate(product_order['Quantity Ordered']):
    plt.text(v + 0.5, i + 0.15, str(v), color='black')

plt.show()
```

Hình 20. Đoạn mã để vẽ biểu đồ tìm ra 10 sản phẩm bán chạy nhất trong năm 2019

b) Kết quả

Hình 21. Biểu đồ cột thể hiện top 10 sản phẩm bán chạy nhất năm 2019

c) Nhận xét

- Dựa trên biểu đồ, ta thấy rằng sản phẩm có số lượng đơn hàng cao nhất là‘ USB-C Charging Cable’ với hơn 21903 đơn hàng được đặt hàng trong năm 2019. Các sản phẩm tiếp theo theo thứ tự giảm dần là,‘

Lightning Charging Cable’,‘ AAA Batteries (4-pack)’,‘ AA Batteries (4-pack)’,‘ Wired Headphones’,‘ Apple Airpods Headphones’,‘ Bose SoundSport Headphones’,‘ 27in FHD Monitor’,‘ iPhone’ và‘ 27in 4K Gaming Monitor’.

- Biểu đồ này có thể giúp cho cửa hàng có thể đưa ra các quyết định về các sản phẩm nên được giới thiệu hoặc tập trung bán hàng hơn để đạt được hiệu quả kinh doanh tốt hơn.

3. Biểu đồ đường thể hiện xu hướng đặt hàng các ngày trong tuần

a) Cách vẽ

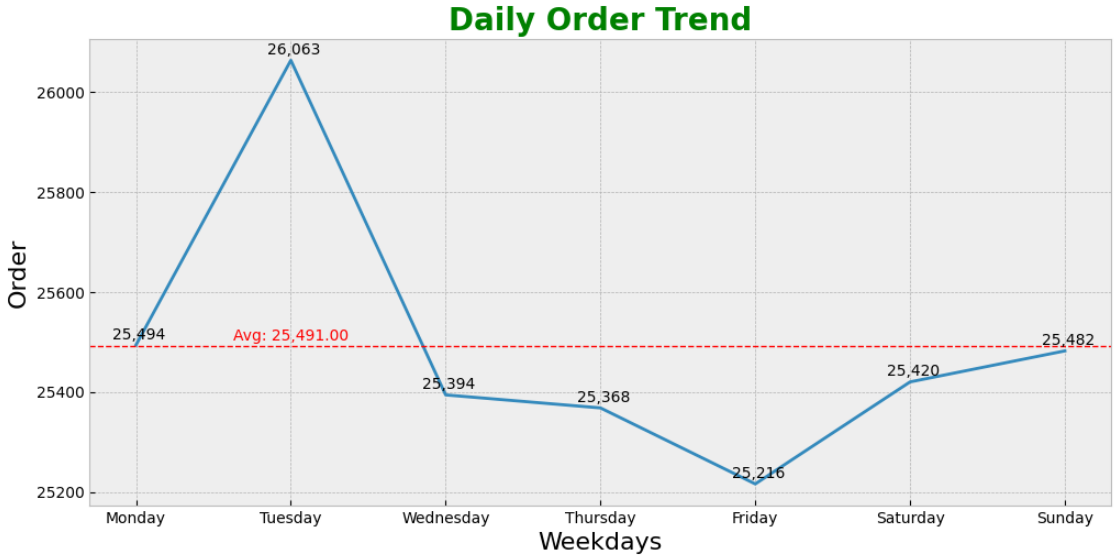
Để vẽ biểu đồ đường ta cần thực hiện các bước sau:

- Đầu tiên, nhóm đã sử dụng hàm pivot_table() để tính tổng số đơn hàng theo ngày trong tuần. Sau đó, chuyển đổi ngày trong tuần từ định dạng số sang định dạng chữ và đổi tên cột.
- Sau đó, tính giá trị trung bình của số đơn hàng trong một ngày và lưu vào DataFrame“ average_daily_order”.
- Tiếp tục, nhóm sử dụng sns.lineplot() của thư viện Seaborn để tạo biểu đồ đường, đồng thời cũng tạo các label cho trục x, trục y và đường kẻ ngang thể hiện giá trị trung bình của số đơn hàng.
- Cuối cùng, nhóm sử dụng plt.annotate() để thêm các chú thích trên biểu đồ, hiển thị giá trị của từng điểm dữ liệu.

```python
daily_order_pivot = pd.pivot_table(total_sales_2019, index=total_sales_2019['Order Date'].dt.dayofweek,
                                   values='Order ID', aggfunc='nunique')
daily_order_pivot = daily_order_pivot.rename(columns={'Order ID': 'Total Ordered'})
daily_order_pivot.index = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
daily_order_pivot.index.name = 'day'
daily_order_pivot = daily_order_pivot.reset_index()
daily_order_pivot = daily_order_pivot.rename(columns={'day': 'Order Date'})
average_daily_order = daily_order_pivot['Total Ordered'].mean()

with plt.style.context('bmh'):
    plt.figure(figsize=(12, 5.5))
    sns.lineplot(x='Order Date', y='Total Ordered', data=daily_order_pivot)
    plt.ticklabel_format(style='plain', axis='y')
    plt.title("Daily Order Trend", fontsize=20, fontweight='bold', color='green')
    plt.xlabel("Weekdays", fontsize=16)
    plt.ylabel("Order", fontsize=16)

    plt.xticks(range(7), daily_order_pivot['Order Date'])
    plt.text(daily_order_pivot['Order Date'].iloc[-6], average_daily_order+5,
             f"Avg: {average_daily_order:,.2f}",
             color='red', fontsize=10, ha='center', va='bottom')
    plt.axhline(y=average_daily_order, color='red', linestyle='--', linewidth=1)
    for x, y in zip(daily_order_pivot['Order Date'], daily_order_pivot['Total Ordered']):
        label = f"{(y):,}"
        plt.annotate(label, (x,y), textcoords='offset points', xytext=(2,4), ha='center')
plt.show()
```

Hình 22. Đoạn mã vẽ biểu đồ đường thể hiện xu hướng đặt hàng các thứ trong tuần

b) Kết quả



Hình 23. Biểu đồ đường thể hiện xu hướng đặt hàng các ngày trong tuần

c) Nhận xét

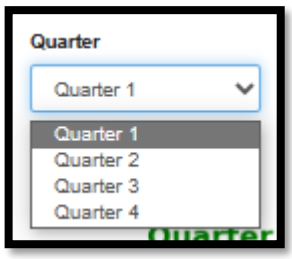
Khi nhìn vào biểu đồ, ta thấy được sự biến động của số đơn hàng được đặt của các ngày trong tuần, cụ thể như sau:

- Nhìn vào biểu đồ, chúng ta có thể thấy rằng số lượng đơn hàng được đặt trong thứ ba là cao nhất(26063 đơn hàng), xếp sau đó là thứ hai và chủ nhất; thấp nhất là vào thứ sáu(25216 đơn hàng). Điều đó cho thấy rằng, khách hàng thường đặt sản phẩm vào cuối hoặc đầu tuần.
- Trên biểu đồ cũng cho thấy giá trị trung bình số đơn hàng trong một ngày là khoảng 25491 đơn hàng.
- Tổng quan về biểu đồ, thì nó đã thấy các thông tin quan trọng về sự khác biệt giữa số đơn hàng được đặt trong các ngày khác nhau trong tuần và giá trị trung bình của số đơn hàng trong một ngày.

4. Biểu đồ tròn thể hiện tỷ lệ doanh số bán hàng của từng tháng trong một quý

a) Cách vẽ

Để vẽ được biểu đồ thể hiện tỷ lệ doanh số bán hàng của từng tháng trong một quý thì đầu tiên ta phải tạo một combobox thể hiện 4 quý trong 1 năm.



Hình 24. Hình combobox với các lựa chọn

Với mỗi lựa chọn của combobox thì ta lấy giá trị của 3 tháng tương ứng với từng quý, giá trị của mỗi tháng được lấy từ DataFrame“ Sales_Month” đã tính ở phần trước(hình 17).

Tiếp theo, nhóm chuyển giá trị lấy được thành một chuỗi rồi truyền qua hàm draw_Donut_Chart()- hình 25 để vẽ đồ thị

```
select = z.select("Quarter", [("1","Quarter 1"),
                              ("2","Quarter 2"),
                              ("3","Quarter 3"),
                              ("4","Quarter 4")])
if (select == '1'):
    y=Sales_Month[0:3]
    Months = ['January', 'February', 'March']
    y = y.values.tolist()
    y = [item for y in y for item in y]
    draw_Donut_Chart(Months, y)
elif (select == '2'):
    y=Sales_Month[3:6]
    Months = ['April', 'May', 'June']
    y = y.values.tolist()
    y = [item for y in y for item in y]
    draw_Donut_Chart(Months, y)
elif (select == '3'):
    y=Sales_Month[6:9]
    Months = ['July', 'August', 'September']
    y = y.values.tolist()
    y = [item for y in y for item in y]
    draw_Donut_Chart(Months, y)
elif (select == '4'):
    y=Sales_Month[9:12]
    Months = ['October', 'November', 'December']
    y = y.values.tolist()
    y = [item for y in y for item in y]
    draw_Donut_Chart(Months, y)
```

Hình 25. Đoạn mã tạo ra combobox và lấy giá trị để truyền đến hàm vẽ đồ thị

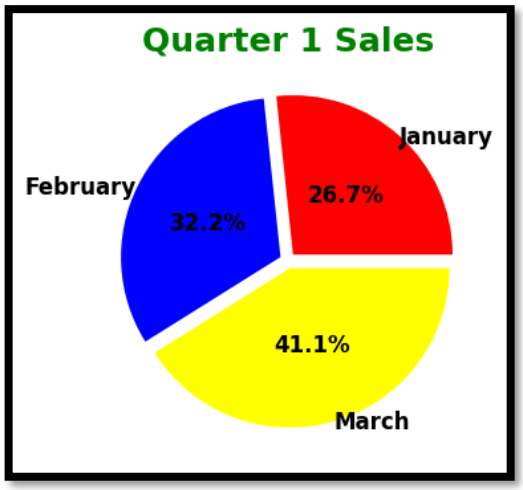
Hàm draw_Donut_Chart() sẽ nhận vào 2 tham số, đó là Months và Values. Months là một danh sách chứa tên của các tháng trong quý đó và Values là một danh sách chứa giá trị của doanh số bán hàng của từng tháng. Hàm này sẽ sử dụng thư viện matplotlib để vẽ biểu đồ Donut Chart tương ứng với dữ liệu được truyền vào.

```python
def draw_Donut_Chart(Months, Values):
    colors = ['#FF0000', '#0000FF', '#FFFF00']
    explode = (0.05, 0.05, 0.05)
    plt.figure(figsize=(4,4.75))

    plt.pie(Values, colors=colors, labels=Months,
        autopct='%1.1f%%', pctdistance=0.5, labeldistance=1,
        explode=explode, textprops={'color':"black", 'fontweight':'bold', 'fontsize':12})

    centre_circle = plt.Circle((0, 0), 0.00, fc='white')
    fig = plt.gcf()

    fig.gca().add_artist(centre_circle)
    if 'January' in Months:
        title_donut = 'Quarter {value} Sales'.format(value = 1)
    elif 'April' in Months:
        title_donut = 'Quarter {value} Sales'.format(value = 2)
    elif 'August' in Months:
        title_donut = 'Quarter {value} Sales'.format(value = 3)
    elif 'October' in Months:
        title_donut = 'Quarter {value} Sales'.format(value = 4)

    plt.title(title_donut,fontsize=18, fontweight='bold', color='green')
    plt.tight_layout(rect=[0.0001, 0.0001, 0.95, 0.95])

    plt.show()
```

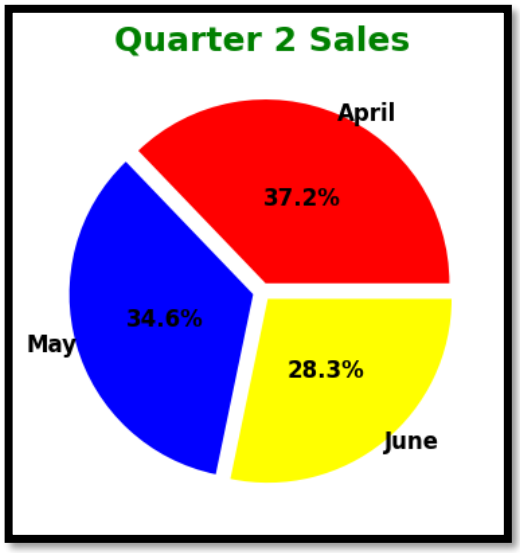
✓ 0.0s

Hình 26. Đoạn mã vẽ biểu đồ hình tròn thể hiện tỷ lệ doanh số bán hàng từng tháng trong quý

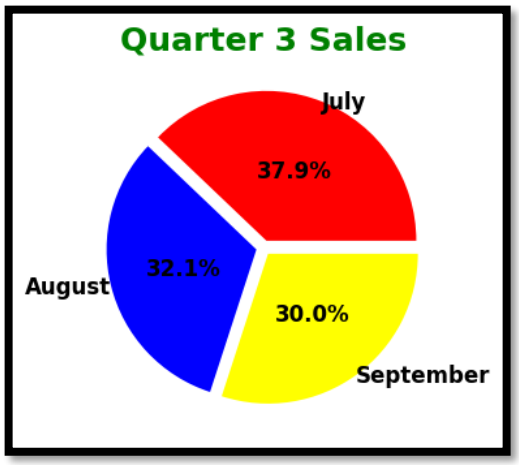
b) Kết quả



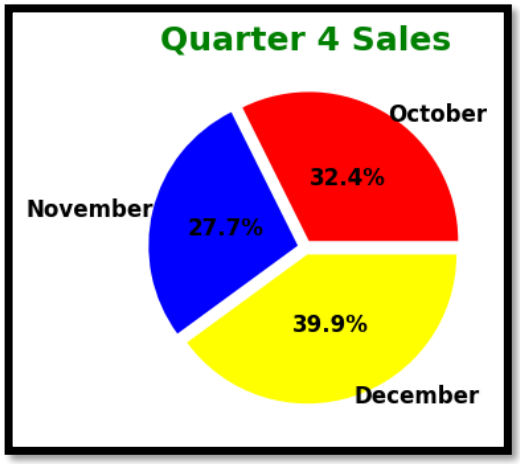
Hình 27. Biểu đồ hình tròn thể hiện tỷ lệ doanh số bán hàng các tháng trong quý 1



Hình 28. Biểu đồ hình tròn thể hiện tỷ lệ doanh số bán hàng các tháng trong quý 2



Hình 29. Biểu đồ hình tròn thể hiện tỷ lệ doanh số bán hàng các tháng trong quý 3



Hình 30. Biểu đồ hình tròn thể hiện tỷ lệ doanh số bán hàng các tháng trong quý 4

c) Nhận xét

Với quý một: Dựa vào biểu đồ, ta thấy được tỷ lệ doanh số bán hàng của 3 tháng đầu năm trong quý 1. Theo đó, tháng 3 chiếm tỷ lệ cao nhất với 41.1%, tiếp theo đó là tháng 2 chiếm 32.2% và cuối cùng là tháng 1 chiếm 26.7%. Điều đó cho thấy, đầu năm người dân ít mua hàng.

Với quý hai: Dựa vào biểu đồ, ta thấy được tỷ lệ doanh số bán hàng của 3 tháng tiếp theo trong quý 2. Theo đó, tháng 6 chiếm tỷ lệ cao nhất với 37.2%, tiếp theo đó là tháng 5 chiếm 34.6% và cuối cùng là tháng 1 chiếm 28.3%.

Với quý 3: Dựa vào biểu đồ, ta thấy được tỷ lệ doanh số bán hàng của 3 tháng tiếp theo trong quý 3. Theo đó, tháng 7 chiếm tỷ lệ cao nhất với 37.9%, tiếp theo đó là tháng 8 chiếm 32.1% và cuối cùng là tháng 9 chiếm 30.0%.

Với quý 4: Dựa vào biểu đồ, ta thấy được tỷ lệ doanh số bán hàng của 3 tháng cuối năm trong quý 4. Theo đó, tháng 12 chiếm tỷ lệ cao nhất với 39.9%, tiếp theo đó là tháng 10 chiếm 32.4% và cuối cùng là tháng 11 chiếm 27.7%.

5. Biểu đồ choropleth thể hiện tổng doanh số bán hàng của của hàng theo từng bang của Mỹ

a) Cách vẽ

Để vẽ được biểu đồ, nhóm đã thực hiện các bước sau:

- Đầu tiên, nhóm tạo DataFrame *sales_by_state_test* từ *total_sales_2019*, chỉ chọn cột “ Order Date”,“ State”,“ Sales”. Sau đó, reset index và đặt cột“ Order Date” làm index.

- Sau đó, tạo cột 'Month/Year' từ index của DataFrame với định dạng '%m/%Y'.
- Tiếp theo, chỉ lấy cột 'Month/Year', 'State', 'Sales' và đổi tên cột 'Sales' thành 'SumSales'.
- Gom nhóm sales_by_state_test theo 'Month/Year' và 'State', tính tổng SumSales và reset index.
- Loại bỏ các hàng có tháng 01/2020 trong sales_by_state_test.
- Đọc file csv“ popular_2019” chứa thông tin dân số của từng bang của Mỹ và merge với file csv“ name_state_usa” chứa mã và tên của từng bang để lấy mã bang (state_code), tên bang (state_name) và số dân của từng bang trong năm 2019.
- Tiếp theo, merge sales_by_state_test với merged_df theo 'State' và “ state_code” để tạo DataFrame“ merged_df_sales_popular” chứa thông tin về tổng doanh số bán hàng, số dân và tên của từng bang.
- Cuối cùng, dùng biểu đồ choropleth của plotly (px.choropleth) để trực quan hóa“ merged_df_sales_popular”. Chỉ định địa điểm (locations), thời gian (animation_frame), phạm vi (scope), màu sắc (color) và thông tin hiện khi rê chuột qua (hover_data) và khi nhấp chuột (hover_name).

```
sales_by_state_test = total_sales_2019[['Order Date', 'State', 'Sales']]
sales_by_state_test = sales_by_state_test.reset_index()
sales_by_state_test = sales_by_state_test.set_index('Order Date')
sales_by_state_test['Month/Year'] = sales_by_state_test.index.strftime('%m/%Y')
sales_by_state_test = sales_by_state_test[['Month/Year', 'State', 'Sales']]
sales_by_state_test.columns = ['Month/Year', 'State', 'SumSales']
sales_by_state_test = sales_by_state_test.groupby(['Month/Year', 'State'])['SumSales'].sum().reset_index()
sales_by_state_test = sales_by_state_test[sales_by_state_test['Month/Year'] != '01/2020']

popular_2019 = pd.read_csv("https://raw.githubusercontent.com/NguyenTrieu903/Sales_Product_Data_Analysis/main/input/2019_Census_US_Population_Data_By_State_Lat_Long.csv")
name_state_usa = pd.read_csv("https://raw.githubusercontent.com/NguyenTrieu903/Sales_Product_Data_Analysis/main/input/usa_states.csv")
merged_df = pd.merge(popular_2019, name_state_usa, left_on='STATE', right_on='state_name')[['state_code','state_name','POPESTIMATE2019']]
merged_df_sales_popular = pd.merge(sales_by_state_test,merged_df, left_on='State', right_on='state_code')
merged_df_sales_popular=merged_df_sales_popular.drop('state_code',axis=1)

fig1 = px.choropleth(merged_df_sales_popular, locations='State', locationmode="USA-states", animation_frame="Month/Year", scope="usa", color='SumSales',
title='Total Sales by State in 2019', color_continuous_midpoint=sales_by_state_test["SumSales"].mean(), hover_data=['POPESTIMATE2019'], hover_name='state_name')\
.update_layout(title={'x': 0.5, 'y': 0.8, 'font_color': 'blue'})

✓ 2.3s                                                                                                                                    Pytho
```

Hình 31. Đoạn mã vẽ biểu đồ choropleth thể hiện tổng doanh số bán hàng của của hàng theo từng bang của Mỹ năm 2019

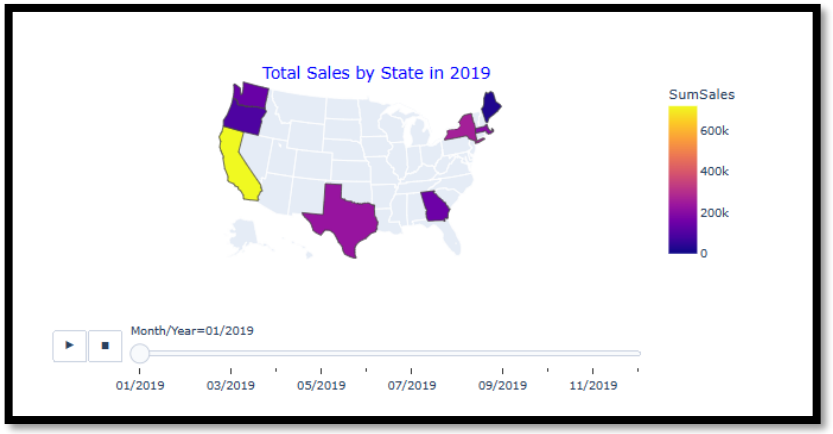
Bước tiếp theo, nhóm sẽ dùng hàm plot() để tạo một biểu đồ dựa trên plot_dic. Tiếp theo, nhóm sẽ sử dụng plotly.offline.plot() để tạo một chuỗi

HTML, và đưa ra đầu ra dưới dạng div theo định dạng Angular. Cuối cùng, hàm sử dụng phương thức print() để xuất ra kết quả với tham số đầu vào là chuỗi HTML để hiển thị biểu đồ được tạo ra trong notebook.

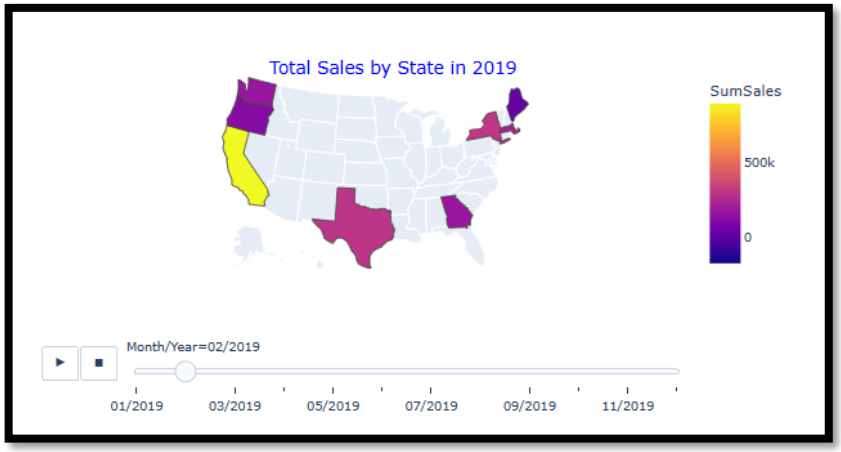
```
def plot(plot_dic, height=420, width=800, **kwargs):
    kwargs['output_type'] = 'div'
    plot_str = plotly.offline.plot(plot_dic, **kwargs)
    print('%%angular <div style="height: %ipx; width: %spx"> %s </div>' % (height, width, plot_str))


plot(fig1)
```

Hình 32. Đoạn mã hiển thị biểu đồ choropleth được tạo ra trong notebook

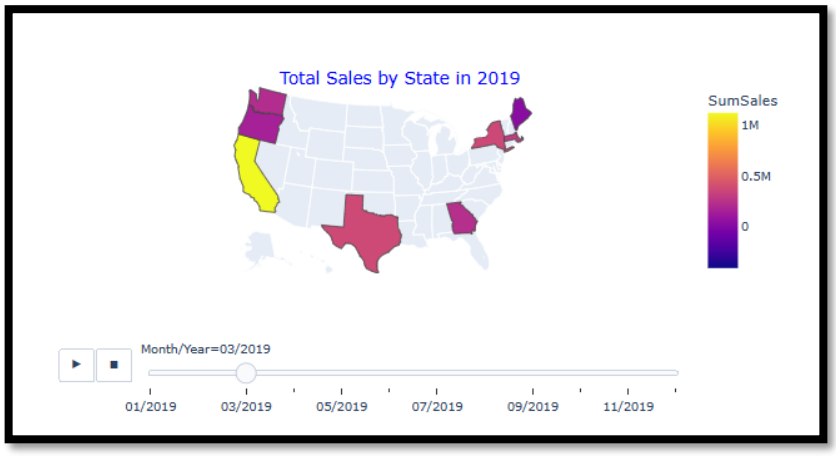
b) Kết quả



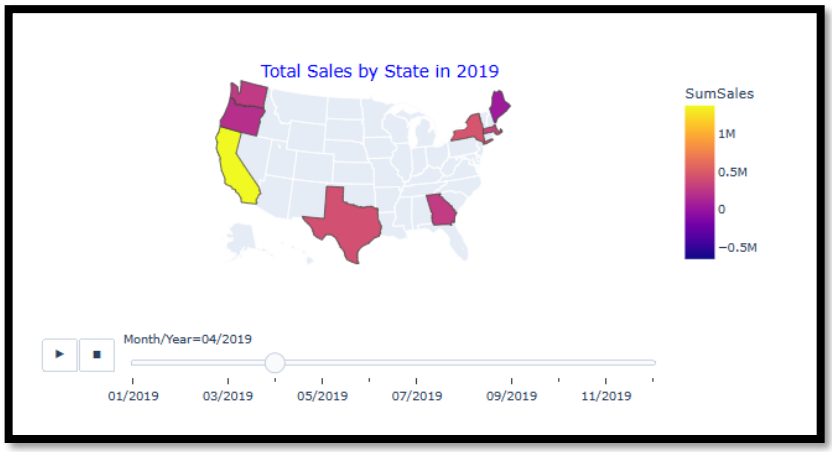
Hình 33. Biểu đồ choropleth thể hiện tổng doanh số bán hàng của của hàng theo từng bang của Mỹ vào tháng 1/2019



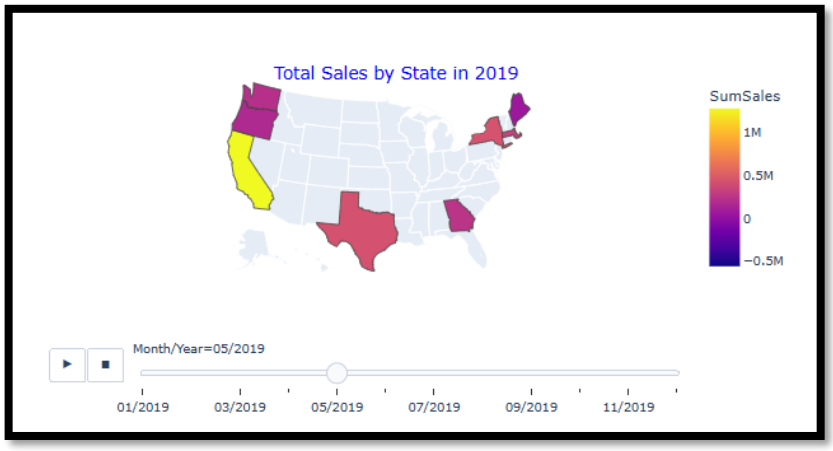
Hình 34. Biểu đồ choropleth thể hiện tổng doanh số bán hàng của của hàng theo từng bang của Mỹ vào tháng 2/2019



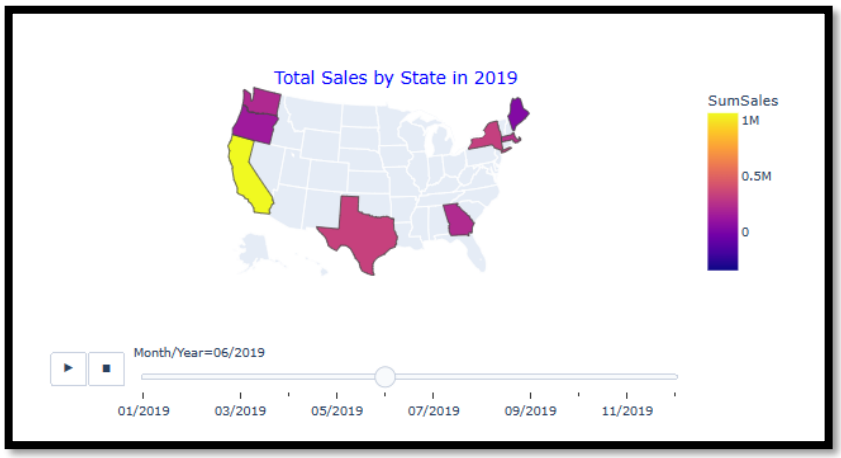
Hình 35. Biểu đồ choropleth thể hiện tổng doanh số bán hàng của của hàng theo từng bang của Mỹ vào tháng 3/2019



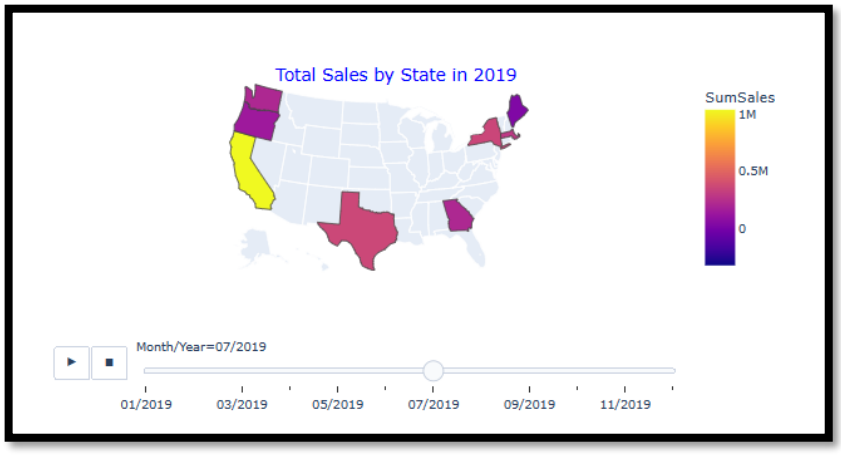
Hình 36. Biểu đồ choropleth thể hiện tổng doanh số bán hàng của của hàng theo từng bang của Mỹ vào tháng 4/2019



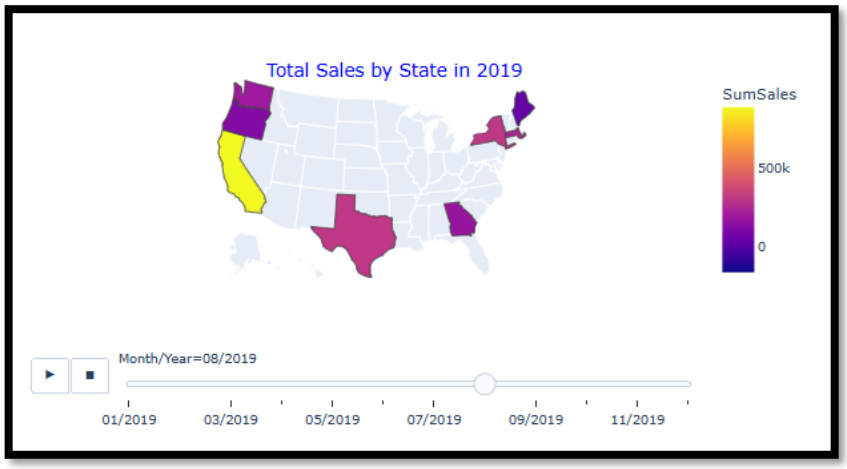
Hình 37. Biểu đồ choropleth thể hiện tổng doanh số bán hàng của của hàng theo từng bang của Mỹ vào tháng 5/2019



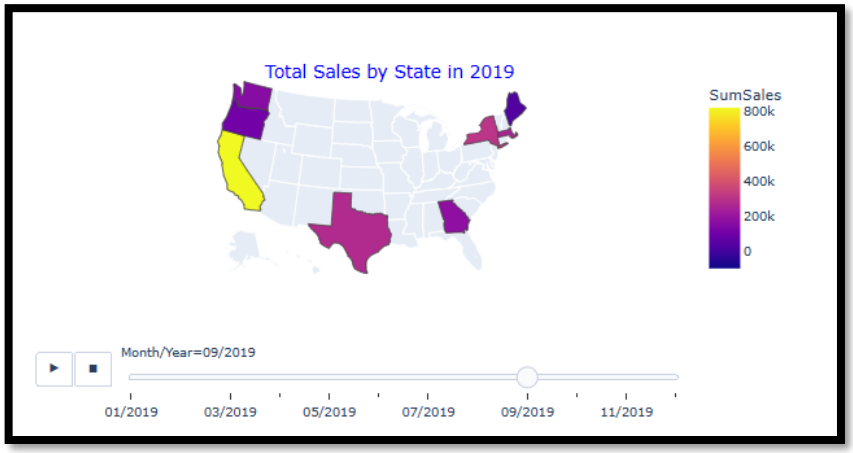
Hình 38. Biểu đồ choropleth thể hiện tổng doanh số bán hàng của của hàng theo từng bang của Mỹ vào tháng 6/2019



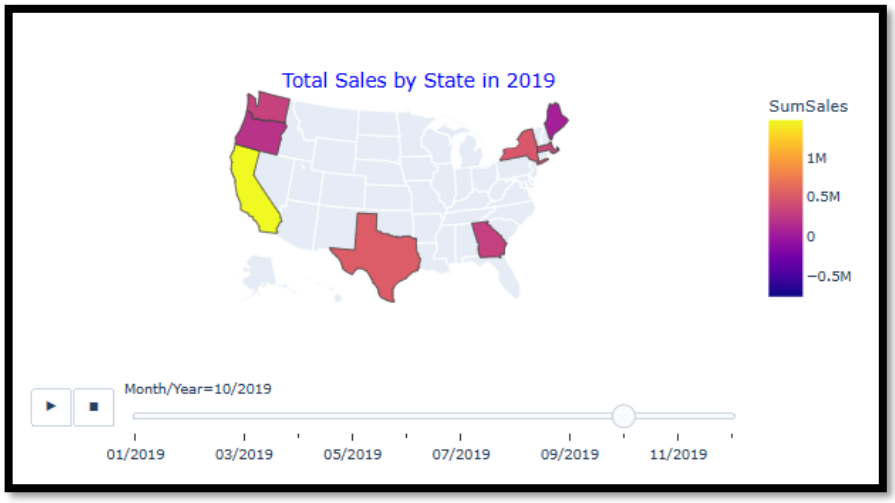
Hình 39. Biểu đồ choropleth thể hiện tổng doanh số bán hàng của của hàng theo từng bang của Mỹ vào tháng 6/2019



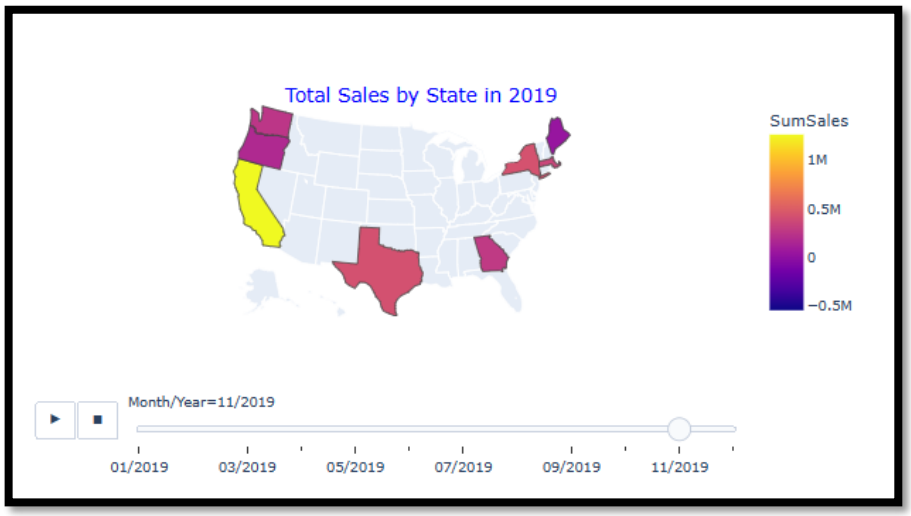
Hình 40. Biểu đồ choropleth thể hiện tổng doanh số bán hàng của của hàng theo từng bang của Mỹ vào tháng 8/2019



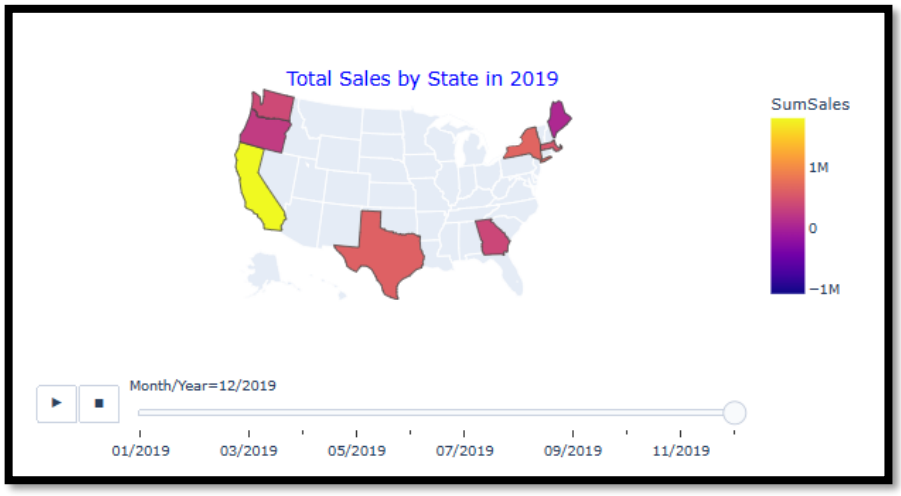
Hình 41. Biểu đồ choropleth thể hiện tổng doanh số bán hàng của của hàng theo từng bang của Mỹ vào tháng 9/2019



Hình 42. Biểu đồ choropleth thể hiện tổng doanh số bán hàng của của hàng theo từng bang của Mỹ vào tháng 10/2019



Hình 43. Biểu đồ choropleth thể hiện tổng doanh số bán hàng của của hàng theo từng bang của Mỹ vào tháng 11/2019



Hình 44. Biểu đồ choropleth thể hiện tổng doanh số bán hàng của của hàng theo từng bang của Mỹ vào tháng 12/2019

c) Nhận xét

Nhìn vào biểu đồ choropleth, nhóm thấy rằng biểu đồ đã cho thấy mức độ tương quan về tổng doanh thu của của hàng giữa các bang qua từng tháng trong năm 2019. Song, khi nhìn vào các tháng thì ta lại thấy có một điểm chung là bang California luôn luôn có đứng ở vị trí đầu và bang Maine luôn nằm ở vị trí cuối. Điều đó cho thấy rằng, việc khách hàng của của cửa hàng này tỷ lệ thuận với tổng số dân của các bang, khi California là bang có số dân lớn nhất của Mỹ.

Cũng qua biểu đồ, ta thấy rằng khách hàng của cửa hàng này nằm chủ yếu 8 bang của Mỹ.

Tóm lại, từ biểu đồ ta có thẻ đưa ra nhiều kết luận, từ đó đưa ra nhungwc định hướng quảng cáo phù hợp với từng bang của Mỹ giúp tăng doanh số bán hàng.

C. PHẦN KẾT LUẬN

1. Kết luận

Qua quá trình tìm hiểu, phân tích và thực nghiệm, nhóm đã đưa ra được những kết luận và nhận xét về tập dữ liệu, từ đó đưa ra một số định hướng đối với của hàng trong tương lai.

2. Kết quả đạt được

- Về mặt lý thuyết: nắm được cách sử dụng một số công nghệ cũng như một số công cụ trong quá trình phân tích như python, apache zeppelin,…
- Về mặt thực nghiệm: vẽ được một số sơ đồ phân tích, cũng như trực quan dữ liệu như biểu cột cột, biểu đồ đường, biểu đồ tròn,…

3. Hạn chế

- Việc chọn dữ liệu, phân tích hạn chế.
- Chưa thật sự hiểu sâu vào tập dữ liệu nên việc đưa ra kết luận còn nhiều thiếu xót.

HẾT