PCPP Module 6
Capstone project
E-commerce

Cohort 38
Group 2
Group members: Tham Mun Yang, Muhammad Afiq Bin Aini, Daniel Khong

Background of our team and motivation

Our company name is “A.L.I. PAPA”

We are a e-commerce startup that plans to launch our new e-commerce platform to obtain first mover advantage in upcoming e-commerce space in 2010-2011.

When the e-commerce industry was nascent and e-commerce is still catching on…

We need to apply data analytics to assist us to navigate our operations to success

Our aspiration

A.L.I. PAPA stands for

A: Availability of inventory

L: Low competitive prices

I: Instant - Quick delivery time, focused logistic network to handle the in-demand areas and periods

PAPA: We aspire to be the ‘PAPA’ of the e-commerce space

Rundown

1. Data understanding
2. Data Cleanup
    a. Treat NaN or None values for CustomerID and Description
    b. Treat Duplicates (Dropped)
    c. Check for cancelled orders (Retained)
    d. StockCode discrepancies (Dropped)
    e. Treat $0 Unit Prices (Dropped)
    f. InvoiceDate column
3. Preliminary EDA and visualization
4. Hypothesis 1
5. Hypothesis 2
6. Hypothesis 3
7. Conclusion

Data Understanding

The dataset consists of 541,909 entries across 8 columns. Below is a detailed overview of each column:

#	Column	Remarks
1	InvoiceNo	Contains the invoice number for each transaction, where each number can represent multiple items purchased in a single transaction.
2	StockCode	Product code for each item in the inventory
3	Description	Product descriptive name. *Some missing data - with 540,455 non-null values (99.73% complete)*
4	Quantity	Indicates the number of products purchased in each transaction.
5	InvoiceDate	Date and time when the purchase transaction occurred.
6	UnitPrice	The price of a single item
7	CustomerID	A unique identifier assigned to each customer. *Significant missing data - with only 406,829 non-null entries (75.07% complete)*
8	Country	The country of residence for the customer.

Observations:

1. To address missing values in the Description and CustomerID columns
2. The InvoiceDate column should be converted into datetime format, which will facilitate further time series analysis.
3. Noted that a single customer can have multiple transactions

Data Understanding

1. Quantity:

- Average quantity of products in a transaction is approximately 9.55.
- The negative values suggest returned or cancelled orders (which needs to be addressed)
- 25% of the Quantity values are at or below 1, and 75% are at or below 10, meaning that most transactions involve small quantities.

2. UnitPrice:

- Average unit price of the products is approximately 4.61.
- The min. unit price include negative values, which needs to be addressed (as negative prices don't make sense)
- Most products are relatively inexpensive as 25% of the UnitPrice values are at or below 1.25, and 75% are at or below 4.13

3. CustomerID:

- There are 406,829 non-null entries, indicating missing values in the dataset (which needs to be addressed)

4. InvoiceNo:

- There are 25,900 unique invoice numbers, indicating 25,900 separate transactions.

5. StockCode:

- There are 4,070 unique stock codes representing different products.

8. Description:

- There are 4,223 unique product descriptions. Despite this, there are 4,070 unique stocks, indicating multiple descriptions for the same StockCode.
- There are some missing values in this column which need to be treated.

7. Country:

- The transactions come from 38 different countries, with a majority originating from United Kingdom.

Data Cleanup

a. Treat NaN or None values for CustomerID and Description
b. Treat Duplicates
c. Check for cancelled orders
d. StockCode discrepancies
e. Treat $0 Unit Prices
f. InvoiceDate column
    a. format InvoiceDate to datetime
    b. add 'date', 'month', 'week', and 'time' columns

Data Cleanup

a. Treat NaN or None values for CustomerID and Description (Dropped)

```
# calculate % of null values for each column
null_percentage = df.isnull().mean() * 100
null_percentage
```

```
InvoiceNo       0.000000
StockCode       0.000000
Description     0.268311
Quantity        0.000000
InvoiceDate     0.000000
UnitPrice       0.000000
CustomerID     24.926694
Country         0.000000
dtype: float64
```

CustomerID column contains nearly 1/4 of missing data (24.93%)

Crucial to have accurate data on customer identifiers - since the clustering is based on customer behavior and preferences.

Therefore - to drop missing CustomerIDs will maintain the integrity of the clusters and further analysis. In addition, dropping missing CustomerID will also remove rows with the missing values in the Description column.

Data Cleanup

b. Treat Duplicates (Dropped)

```
# chk for duplicate rows
duplicates = df.duplicated()
df[duplicates]
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
517	536409	21866	UNION JACK FLAG LUGGAGE TAG	1	12/1/2010 11:45	1.25	17908.0	United Kingdom
527	536409	22866	HAND WARMER SCOTTY DOG DESIGN	1	12/1/2010 11:45	2.10	17908.0	United Kingdom
537	536409	22900	SET 2 TEA TOWELS I LOVE LONDON	1	12/1/2010 11:45	2.95	17908.0	United Kingdom
539	536409	22111	SCOTTIE DOG HOT WATER BOTTLE	1	12/1/2010 11:45	4.95	17908.0	United Kingdom
555	536412	22327	ROUND SNACK BOXES SET OF 4 SKULLS	1	12/1/2010 11:49	2.95	17920.0	United Kingdom
...	...	...	...	...	...	...	...	...
541675	581538	22068	BLACK PIRATE TREASURE CHEST	1	12/9/2011 11:34	0.39	14446.0	United Kingdom
541689	581538	23318	BOX OF 6 MINI VINTAGE CRACKERS	1	12/9/2011 11:34	2.49	14446.0	United Kingdom
541692	581538	22992	REVOLVER WOODEN RULER	1	12/9/2011 11:34	1.95	14446.0	United Kingdom
541699	581538	22694	WICKER STAR	1	12/9/2011 11:34	2.10	14446.0	United Kingdom
541701	581538	23343	JUMBO BAG VINTAGE CHRISTMAS	1	12/9/2011 11:34	2.08	14446.0	United Kingdom

5225 rows × 8 columns

- Dataset might have completely identical rows, incl. identical transaction time.
  - might be data recording errors rather than genuine repeated transactions.

- Keeping duplicate rows risk potential inaccuracies & removing them will help in achieving a cleaner dataset
  - builds more accurate customer clusters based on their unique purchasing behaviors

Data Cleanup

c. Check for cancelled orders (Retained)

```
df[df["Quantity"] < 0]
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
141	C536379	D	Discount	-1	12/1/2010 9:41	27.50	14527.0	United Kingdom
154	C536383	35004C	SET OF 3 COLOURED FLYING DUCKS	-1	12/1/2010 9:49	4.65	15311.0	United Kingdom
235	C536391	22556	PLASTERS IN TIN CIRCUS PARADE	-12	12/1/2010 10:24	1.65	17548.0	United Kingdom
236	C536391	21984	PACK OF 12 PINK PAISLEY TISSUES	-24	12/1/2010 10:24	0.29	17548.0	United Kingdom
237	C536391	21983	PACK OF 12 BLUE PAISLEY TISSUES	-24	12/1/2010 10:24	0.29	17548.0	United Kingdom
...	...	...	...	...	...	...	...	...

Assumption is made that negative “Quantity” where “InvoiceNo” begins with the letter 'C' indicates returned items or cancelled orders

Data Cleanup

c. Check for cancelled orders (Retained)

```
df["is_Cancelled"] = df["InvoiceNo"].apply(lambda x: True if x.startswith('C') else False)
df["is_Cancelled"].value_counts(normalize=True)
```

```
is_Cancelled
False    0.977909
True     0.022091
Name: proportion, dtype: float64
```

The percentage of cancelled transactions in the dataset is: 2.21%

```
df[df["is_Cancelled"]].describe().drop('CustomerID', axis=1).T
```

	count	mean	std	min	25%	50%	75%	max
Quantity	8872.0	-30.774910	1172.249902	-80995.00	-6.00	-2.00	-1.00	-1.0
UnitPrice	8872.0	18.899512	445.190864	0.01	1.45	2.95	4.95	38970.0

Cancelled transactions are retained but revenue figures for these transactions will be adjust to zero.

Data Cleanup

d. StockCode discrepancies (Dropped)

The aim is to cluster customers based on their product purchasing behaviors, hence records with peculiar stock codes records should be excluded from the dataset.

Focus then remains strictly on genuine product transactions, leading to a more accurate and meaningful analysis.

```
df["StockCode"].nunique()
```

```
3684
```

```
df["len_StockCode"] = df["StockCode"].str.strip().str.len()
df["len_StockCode"].value_counts(normalize=True)
```

```
len_StockCode
5     0.911791
6     0.082703
4     0.003028
1     0.001337
7     0.000737
2     0.000334
3     0.000040
12    0.000030
Name: proportion, dtype: float64
```

Majority of the unique stock codes (3676 out of 3684) contain exactly 5 numeric characters, which seems to be the standard format for representing product codes in this dataset.

Data Cleanup

e. Treat $0 Unit Prices (Dropped)

The minimum unit price value should be zero - suggesting transactions where the unit price = 0 potentially indicating a free item or a data entry error.

```
df[df["UnitPrice"]==0]
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	C
6842	537197	22841	ROUND CAKE TIN VINTAGE GREEN	1	12/5/2010 14:02	0.0	
22619	539263	22580	ADVENT CALENDAR GINGHAM SACK	4	12/16/2010 14:36	0.0	
25551	539722	22423	REGENCY CAKESTAND 3 TIER	10	12/21/2010 13:45	0.0	
29374	540372	22090	PAPER BUNTING RETROSPOT	24	1/6/2011 16:41	0.0	
29376	540372	22553	PLASTERS IN TIN SKULLS	24	1/6/2011 16:41	0.0	

Data Cleanup

f. InvoiceDate column

- format InvoiceDate to datetime
- add 'date', 'month', 'week', and 'time' columns

To better analyze trends over time for seasonality and cyclical behavior.

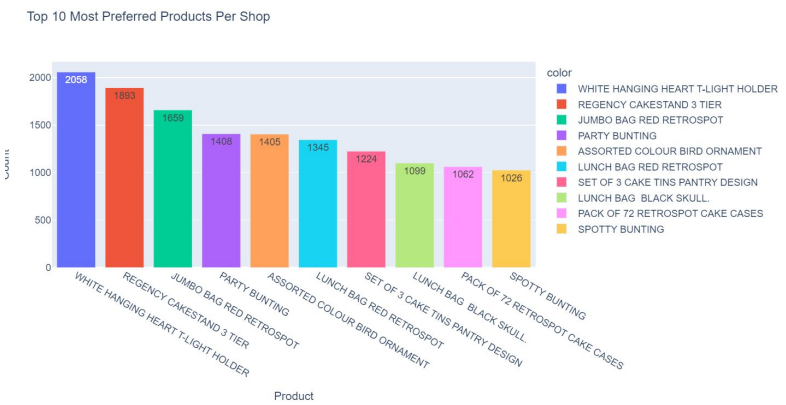
Before

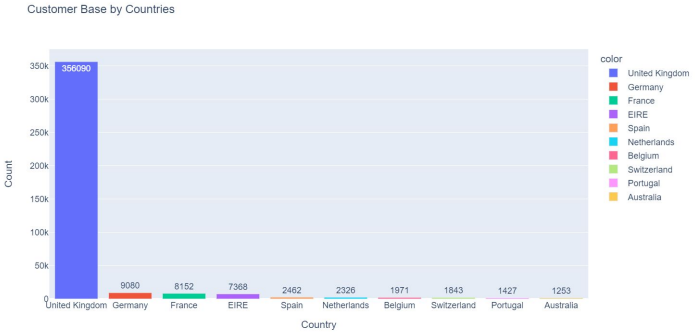
#	Column	Non-Null Count	Dtype
0	InvoiceNo	541909 non-null	object
1	StockCode	541909 non-null	object
2	Description	540455 non-null	object
3	Quantity	541909 non-null	int64
4	InvoiceDate	541909 non-null	object

After

#	Column	Non-Null Count	Dtype
0	InvoiceNo	541909 non-null	object
1	StockCode	541909 non-null	object
2	Description	540455 non-null	object
3	Quantity	541909 non-null	int64
4	InvoiceDate	541909 non-null	datetime64[ns]
5	UnitPrice	541909 non-null	float64
6	CustomerID	406829 non-null	float64
7	Country	541909 non-null	object
8	date	541909 non-null	object
9	month	541909 non-null	object
10	week	541909 non-null	object

Preliminary EDA and visualization before developing our hypothesis





3. Product Analysis

In [41]:
```
# best selling products

best_sellers = df.groupby(['StockCode', 'Description'])['Quantity'].sum().sort_values(ascending=False)
print(best_sellers.head(10))
```

```
StockCode  Description
84077      WORLD WAR 2 GLIDERS ASSTD DESIGNS     53119
85099B     JUMBO BAG RED RETROSPOT               44963
84879      ASSORTED COLOUR BIRD ORNAMENT         35215
85123A     WHITE HANGING HEART T-LIGHT HOLDER    34128
21212      PACK OF 72 RETROSPOT CAKE CASES       33386
22197      POPCORN HOLDER                        30492
23084      RABBIT NIGHT LIGHT                    27045
22492      MINI PAINT SET VINTAGE                25880
22616      PACK OF 12 LONDON TISSUES             25305
21977      PACK OF 60 PINK PAISLEY CAKE CASES    24129
Name: Quantity, dtype: int64
```

Using the cleaned up data to perform preliminary EDA and visualizations to gain insights on data relationships between the numerical and categorical data to help ourselves to develop hypothesis. E.g. Groupby stats

Hypothesis statements at a glance:

1) Low cost products would generate the most revenue for our business, therefore we should focus our product offering on low cost products.

2) E-commerce purchasing and transactions are concentrated on the weekends and after-work hours as people return home.

3) Purchasing transactions increase during specific seasons such as end of year holiday seasons, this pattern is also true for e-commerce. For European countries, purchase transactions would peak during the end of year holiday season.

Hypothesis 1:

Hypothesis: *“Low cost products would generate the most revenue for our business, therefore we should focus our product offering on low cost products. ”*

How we will prove it: Analyze relationship between unit price of items and the revenue they generate. Then determine whether there is indeed a trend where lower-priced items lead to higher overall revenue.

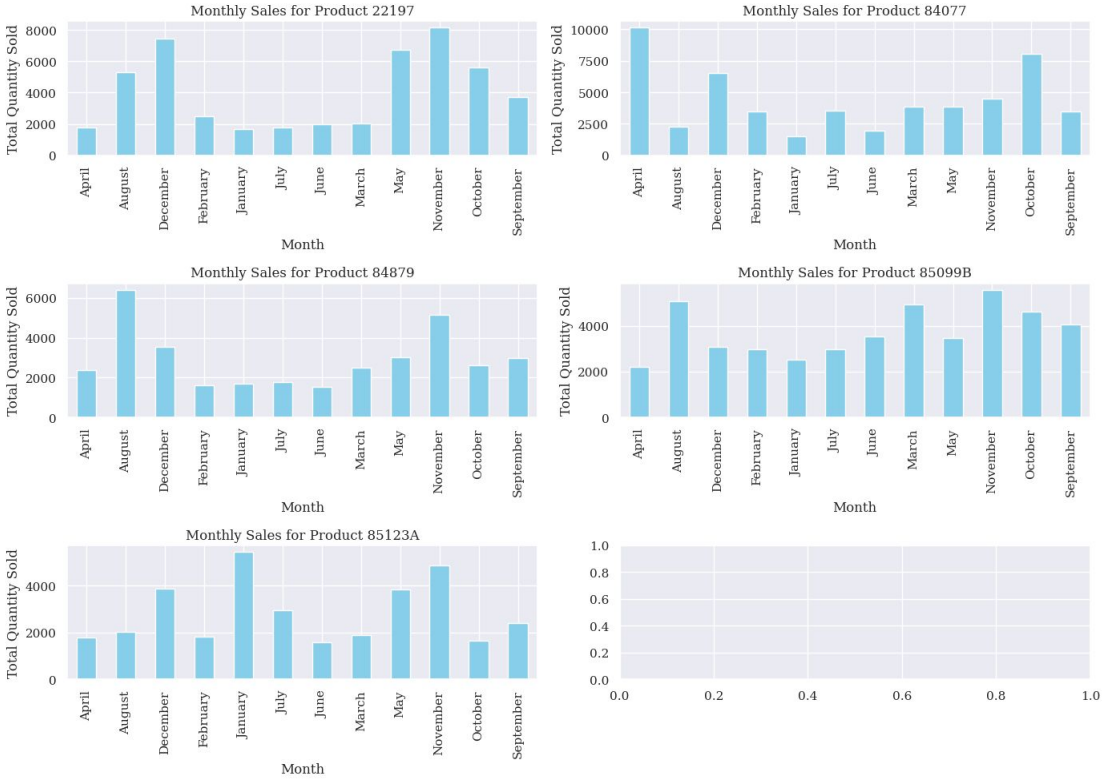
Hypothesis 1:

Conclusions

- "Low ($0-5)" category contributes significantly more to the total revenue compared to other categories
- Plot supplements the pricing strategy's effectiveness concerning revenue generation

Hypothesis holds

Hypothesis 1:



Further Insights

- Top 5 products are within the "Low ($0-5)" category.
- Bar plot demonstrates seasonal trends for monthly sales volumes for each of the top 5 products

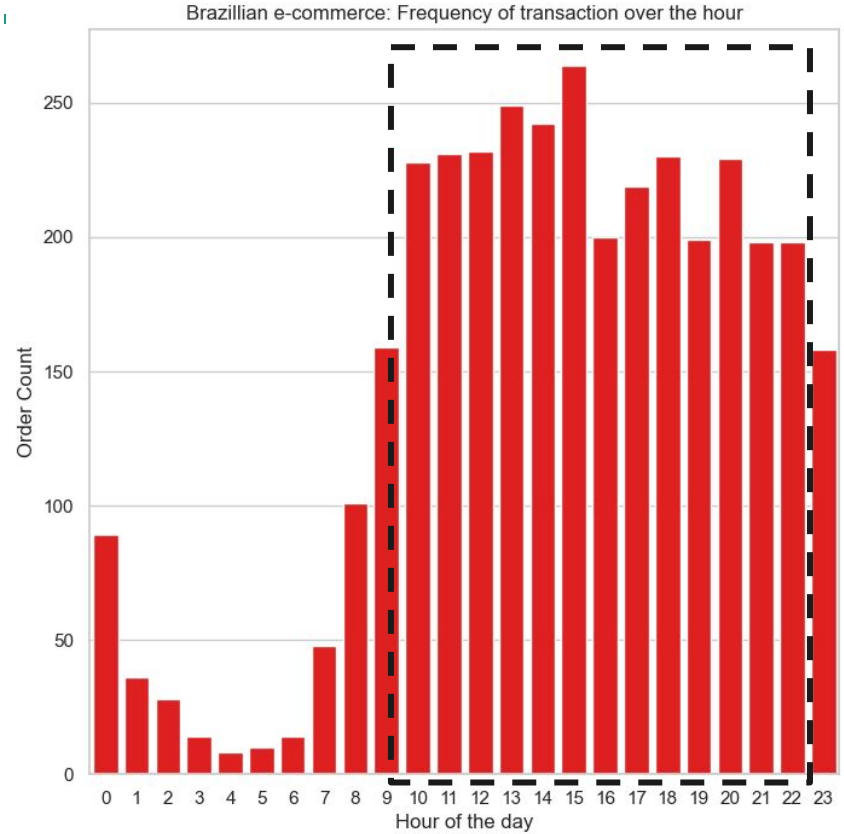
Hypothesis 2:

Hypothesis: *“E-commerce purchasing and transactions are concentrated on the weekends and after-work hours as people return home. ”*

How we will prove it: Plotting a heat map of the quantity of purchases each day (x-axis) versus the Hour of the Day (y-axis), we should expect to see darker shades on Days 5 and 6 (Saturday and Sunday) after 1700 hours

Hypothesis 2: We first looked at aggregate purchasing across the hours

And compared it with external sources for another country



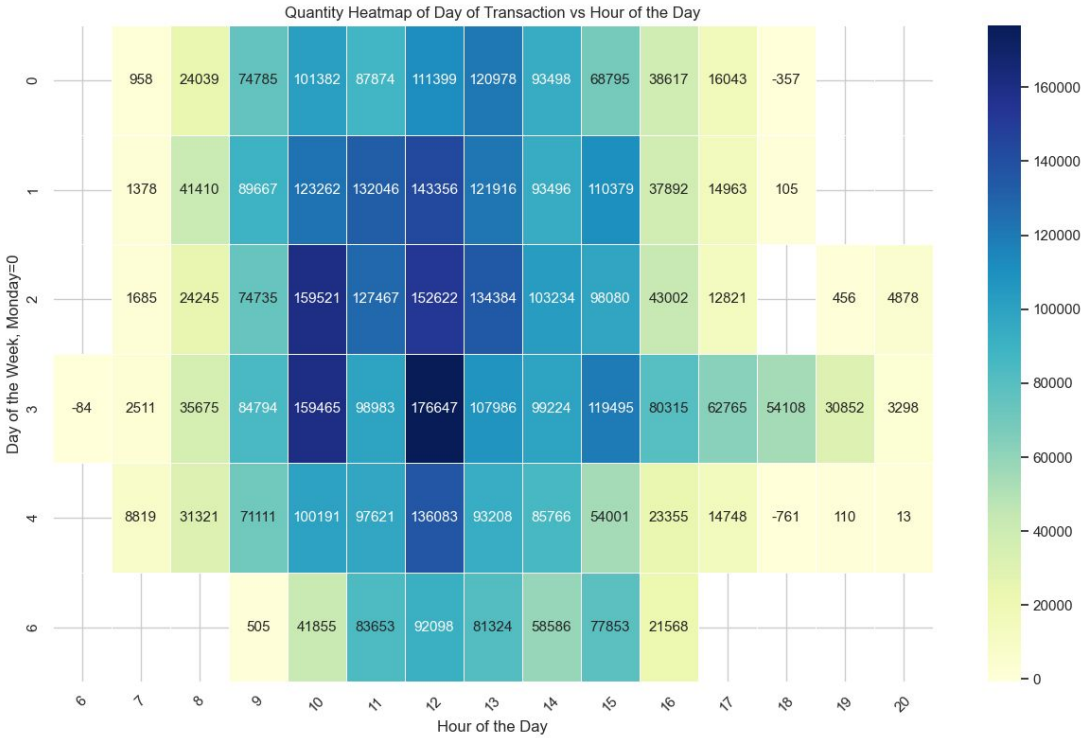
Observations

- Brazilian e-commerce volumes start ramping up from 10am, similar to UK
- Volumes peak at 3pm and start to taper from 4pm until midnight
- We also generated a heatmap from the external data source to compare with our heatmap to verify

Source:

Brazilian E-Commerce Public Dataset by Olist
https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce/versions/3/data
Dataset: olist_classified_public_dataset.csv

Followed by a heat map through days of the week



Conclusions

- Most purchases are from Monday to Friday, with the highest concentrated on Wednesday and Thursday
- The heaviest purchasing is between 10am to 3pm
- Buyers do some purchasing on Sunday but do not purchase on Saturday

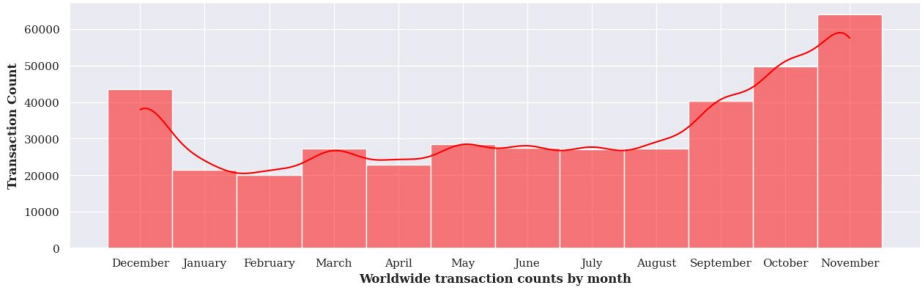
Hypothesis does not hold

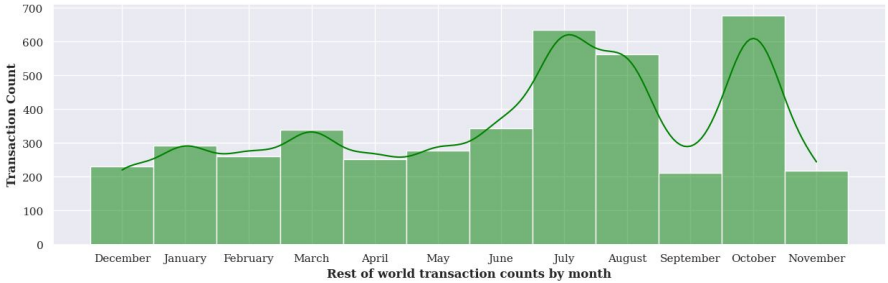
Hypothesis 3:

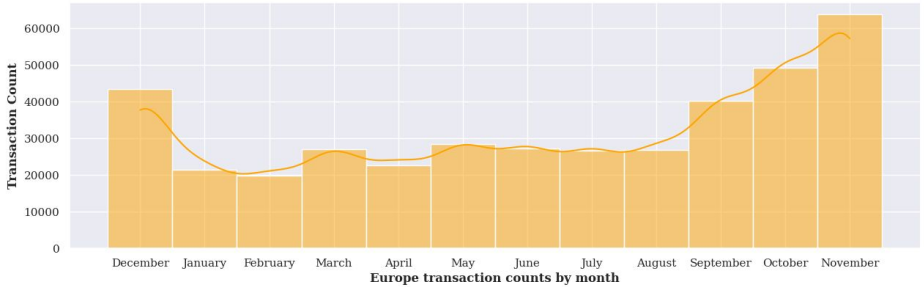
Hypothesis: *“Purchasing transactions increase during specific seasons such as end of year holiday seasons, this pattern is also true for e-commerce. For European countries, Purchasing transactions would peak during the end of year holiday season. ”*

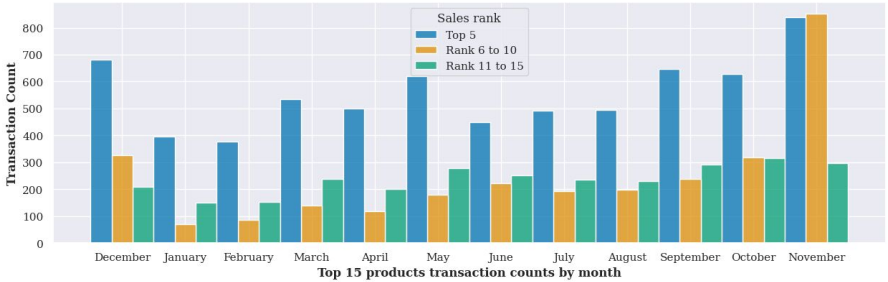
How we will prove it: Filter the processed dataset to obtain sub-datasets of worldwide, europe and rest of the world purchase transaction counts over macro time period of months of the year. Additional viewpoint using top 15 products transaction counts over months.
Next, plot the purchase transaction counts over months to determine whether the existing held fact that peak periods will at the end of the year.

Hypothesis 3: Trend of purchase transaction counts segmented by worldwide, regional. Top 15 product trends adds another viewpoint









Conclusion:

For worldwide and Europe, which is our primary market, the data analysis of transaction counts vs months confirms that the busiest period for our order management and fulfilment would be september to december.

Contrary to this, for rest of the world, the trend demonstrates that this busiest period is rather on July, August, October months.

Conclusion

Achieving our aspiration:

A: Availability of inventory, L: Low competitive prices, I: Instant

Our conclusions for our work will assist us, a start-up e-commerce company in 2010-2011 time period greatly in product mix and Inventory availability (from hypothesis 1) and logistic and resource allocation (from hypothesis 2 and 3).

For inventory availability, and hypothesis 1 conclusion, we will focus our product mix to on the bestsellers and low cost products since "Low ($0-5)" category contributes significantly more to the total revenue compared to other categories.

For logistics and computing server allocation, in micro timescale of week, we will prepare for hot periods within the time of week identified (mid week, early morning to noon time) so as to fulfil our aspiration of being 'available' and 'instant' to our customers.

For the macro timescale of months, we will stock up the best seller and low cost times (as identified to focus in hypothesis 1), during the anticipated peak months of september to december. So that we will not close sales to other competitors and increase customer satisfaction.