Exploratory Data Analysis of supermarket sales in Myanmar

Dr. Li Zhihuan, PhD. in Engineering

OUTLINE

- Executive summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

EXECUTIVE SUMMARY

- An Exploratory Data Analysis (EDA) was conducted on a dataset of sales for a supermarket chain in Myanmar
- The raw CSV file was imported into a SQLite database and pre-processed
- The SQLite DB file was then imported into a Pandas dataframe to undergo further processing
- The analysis was then conducted with PyGWalker

INTRODUCTION

- Myanmar is a developing country that may offer exciting growth opportunities if it manages to get over its political troubles
- This EDA aims to provide greater insight into the sales data of the supermarket
- Cost reduction measures can generate greater income and increase market share

METHODOLOGY

- The raw CSV file was downloaded from Kaggle
- The file consists of the sales data from January to March 2019
- The raw CSV file was imported into a SQLite database and pre-processed
- SQLite was chosen to demonstrate proficiency as it was built into Python

```
df=pd.read_csv('supermarket.csv')
conn=sqlite3.connect('supermarket.db')
df.to_sql(name='supermarket', con=conn, if_exists='replace', index=False)
conn.commit()
```

[2] ✓ 0.0s

```
%config SqlMagic.displaylimit = 15 ···
```

```
%load_ext sql
%sql sqlite:///supermarket.db
```

[4] ✓ 0.1s

METHODOLOGY

- The SQL table was first checked for any duplicate values
- No duplicate values were found

```
%%sql
SELECT "Invoice ID", COUNT("Invoice ID") AS id_count, SUM(COUNT("Invoice ID")) OVER() AS total_count
FROM supermarket GROUP BY "Invoice ID" having count("Invoice ID")>1
```

[6] ✓ 0.0s

Running query in 'sqlite:///supermarket.db'

Invoice ID	id_count	total_count

It seems that there are no duplicated entries.

METHODOLOGY

- The SQL table was next checked for any NULL values
- There were no NULL values found

```
%%sql
SELECT *
FROM supermarket
WHERE ("Invoice ID" or "City" or "Customer type" or "Gender" or "Product line" or "Unit price" or "Quantity" or "Total" or "Date" or "Time" or "Payment" or cogs or "gross
```

[8] ✓ 0.0s Python

Running query in 'sqlite:///supermarket.db'

Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	cogs	gross margin percentage	gross income	Rating

There are no null values in the table.

METHODOLOGY

- The SQL table was then converted back into a Pandas dataframe
- The ‘Branch’ column was dropped as each branch was situated in a different city, and the ‘City’ column will be more informative in the EDA

```
df.drop(inplace=True, axis=1, columns='Branch')
df.describe(include='all')
```

[11] ✓ 0.0s Python

	Invoice ID	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	cogs	gross margin percentage	gross income	Rati
count	1000	1000	1000	1000	1000	1000.000000	1000.000000	1000.000000	1000.000000	1000	1000	1000	1000.00000	1.000000e+03	1000.000000	1000.000
unique	1000	3	2	2	6	NaN	NaN	NaN	NaN	89	506	3	NaN	NaN	NaN	Na
top	750-67-8428	Yangon	Member	Female	Fashion accessories	NaN	NaN	NaN	NaN	2/7/2019	19:48	Ewallet	NaN	NaN	NaN	Na
freq	1	340	501	501	178	NaN	NaN	NaN	NaN	20	7	345	NaN	NaN	NaN	Na
mean	NaN	NaN	NaN	NaN	NaN	55.672130	5.510000	15.379369	322.966749	NaN	NaN	NaN	307.58738	4.761905e+00	15.379369	6.972
std	NaN	NaN	NaN	NaN	NaN	26.494628	2.923431	11.708825	245.885335	NaN	NaN	NaN	234.17651	6.131498e-14	11.708825	1.718
min	NaN	NaN	NaN	NaN	NaN	10.080000	1.000000	0.508500	10.678500	NaN	NaN	NaN	10.17000	4.761905e+00	0.508500	4.000
25%	NaN	NaN	NaN	NaN	NaN	32.875000	3.000000	5.924875	124.422375	NaN	NaN	NaN	118.49750	4.761905e+00	5.924875	5.500
50%	NaN	NaN	NaN	NaN	NaN	55.230000	5.000000	12.088000	253.848000	NaN	NaN	NaN	241.76000	4.761905e+00	12.088000	7.000
75%	NaN	NaN	NaN	NaN	NaN	77.935000	8.000000	22.445250	471.350250	NaN	NaN	NaN	448.90500	4.761905e+00	22.445250	8.500
max	NaN	NaN	NaN	NaN	NaN	99.960000	10.000000	49.650000	1042.650000	NaN	NaN	NaN	993.00000	4.761905e+00	49.650000	10.000

METHODOLOGY

- Some feature engineering was also performed
- The gross income and Cost Of Goods Sold (COGS) per unit was calculated and the ‘Date’ and ‘Time’ columns were turned into something easier to work with

```
for i in df:
    df['ucogs']=(df['cogs']/df['Quantity'])
    df['unit gross income'] = (df['gross income']/df['Quantity'])
df.Time.dtypes
```

[13] ✓ 0.0s

```
dtype('O')
```

```
df['Time']=df['Time'].str.replace(':','.')
df['Time'].astype(str).dtypes
```

[14] ✓ 0.0s

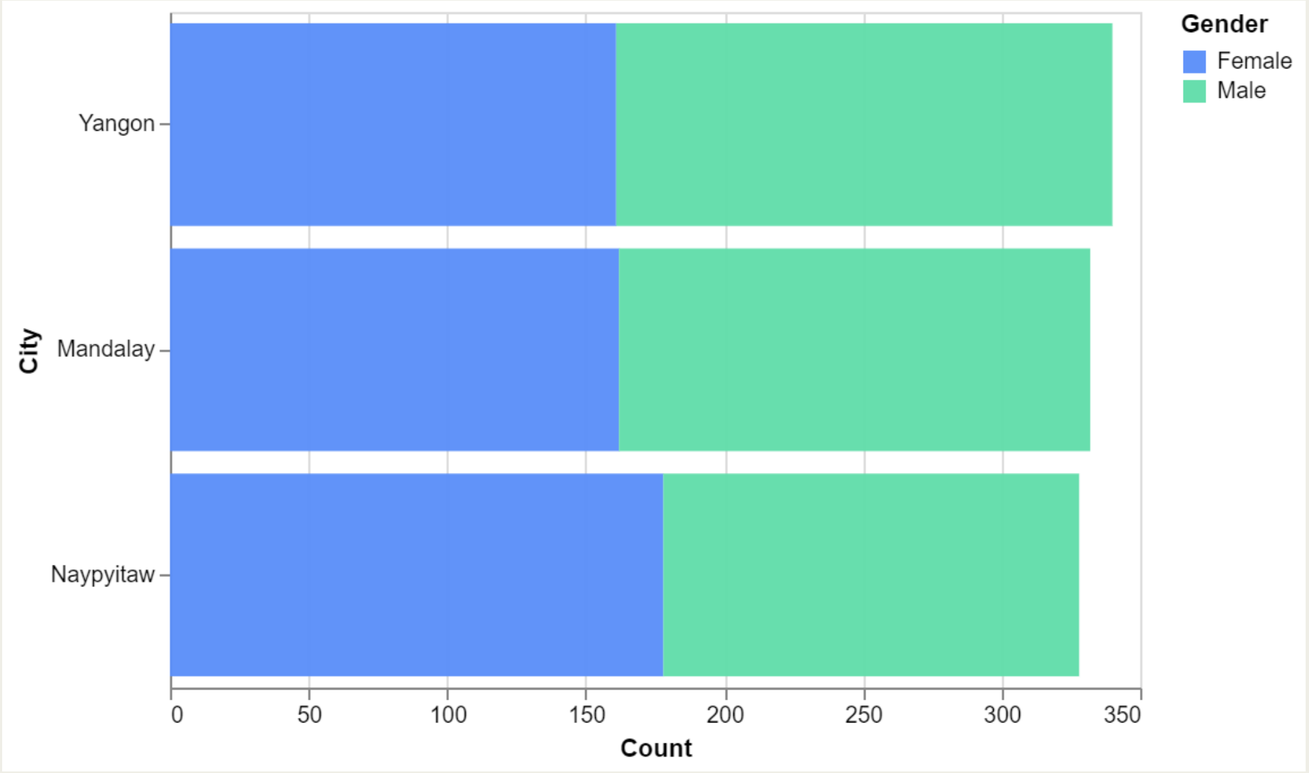
```
dtype('O')
```

```
df['hour']=df['Time'].str.split('.', expand=True)[0]
df['minutes']=df['Time'].str.split('.', expand=True)[1]
df.head()
```

[15] ✓ 0.0s

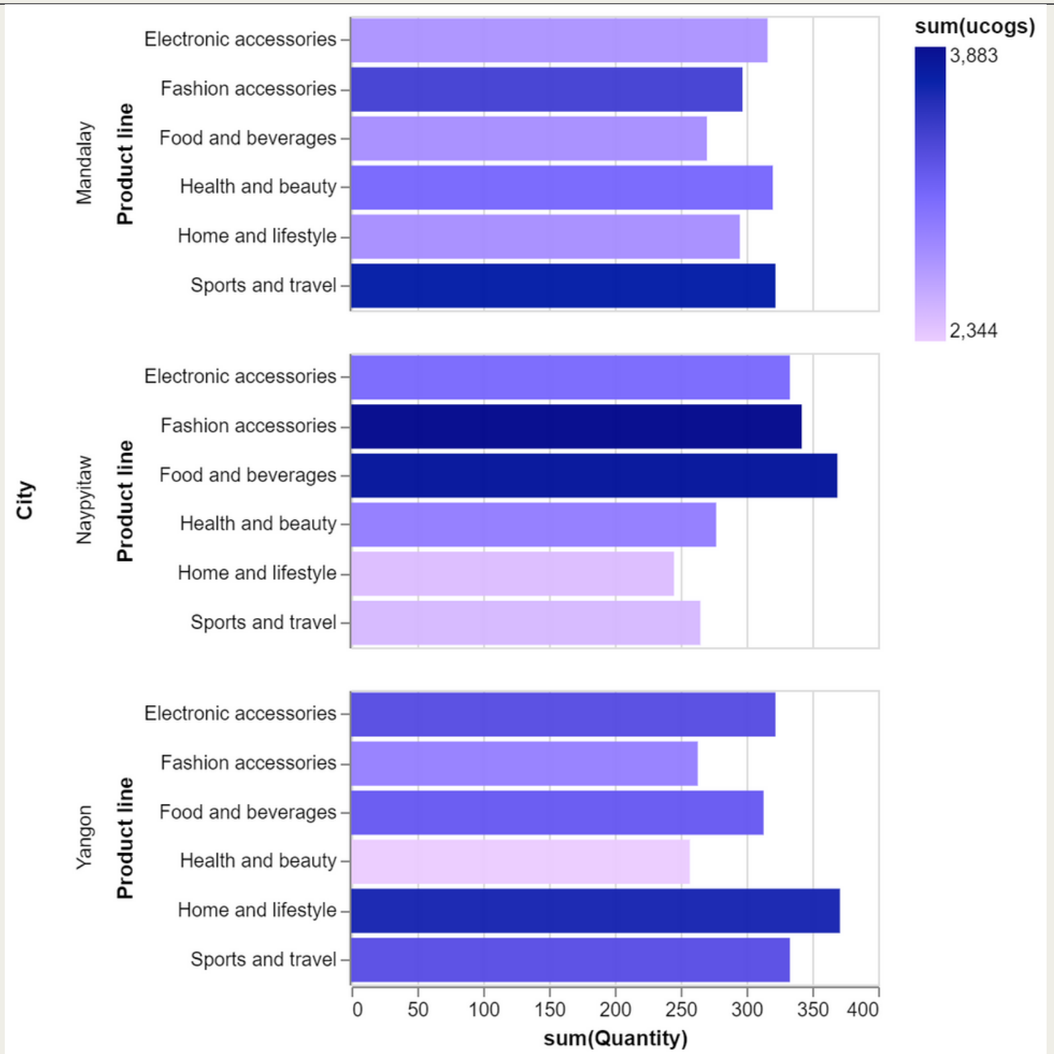
RESULTS

- Each branch had similar sales volume, and the customer base is split evenly gender-wise
- Population of:
  - Yangon: 5,160,512
  - Mandalay: 1,726,889
  - Naypyitaw (capital): 924,608
- Even though the population of Yangon is much higher than Mandalay or Naypyitaw, it has the same sales volume
- A second branch in Yangon may be worth looking into



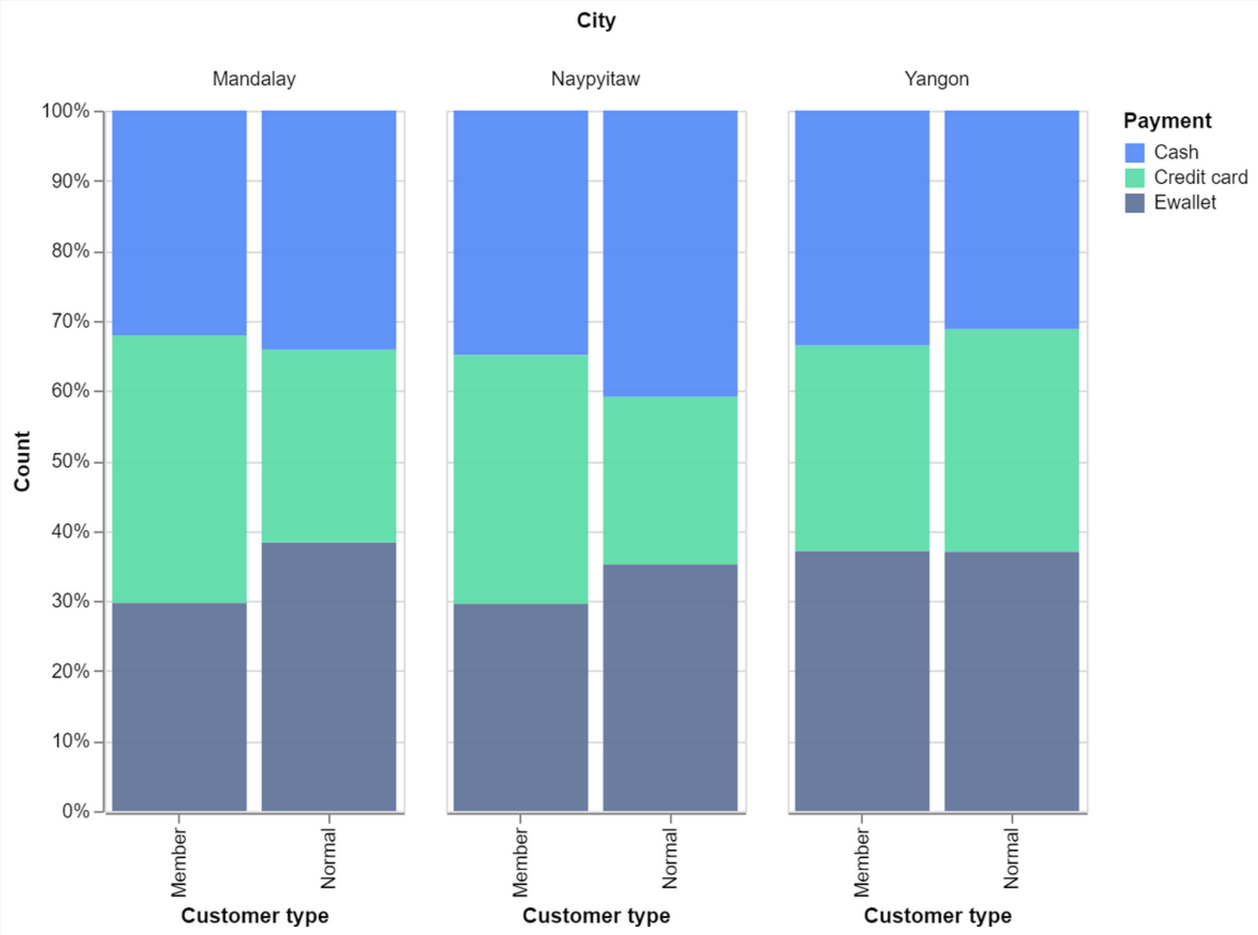
RESULTS

- The COGS for the different categories vary among cities
- Highest cost of goods sold for:
  - Mandalay: Sports and travel
  - Naypyitaw: Food and beverages
  - Yangon: Home and lifestyle
- The goods may be sourced from different suppliers
- COGS could be reduced could be reduced by building a centralised warehouse to store goods from the cheapest supplier
- A feasibility study should be done



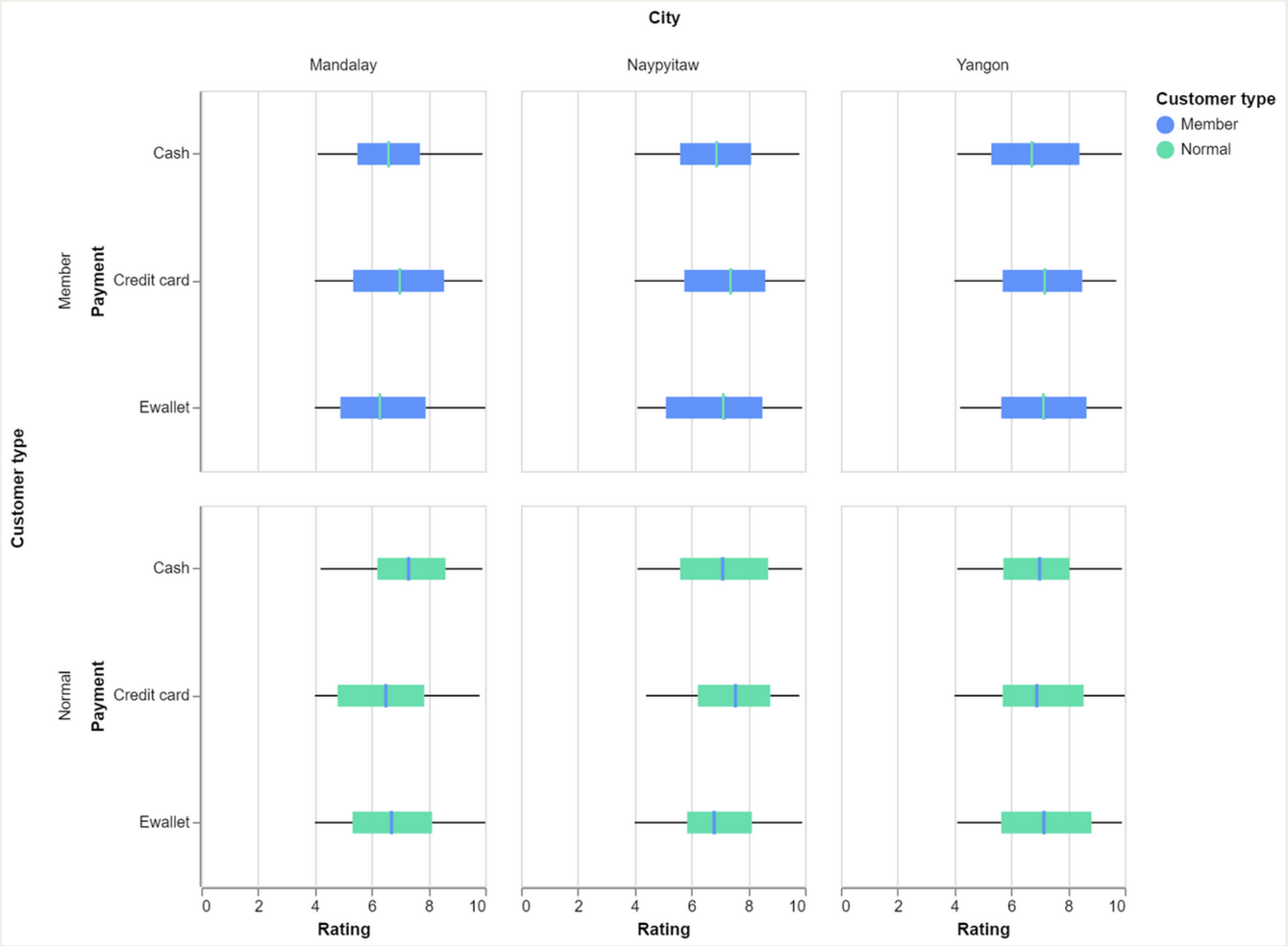
RESULTS

- Adoption of cashless payment methods is widespread
- There is potential to collaborate with credit card and fintech companies
- Members seem more likely to pay via credit card and less likely via cash and ewallets



RESULTS

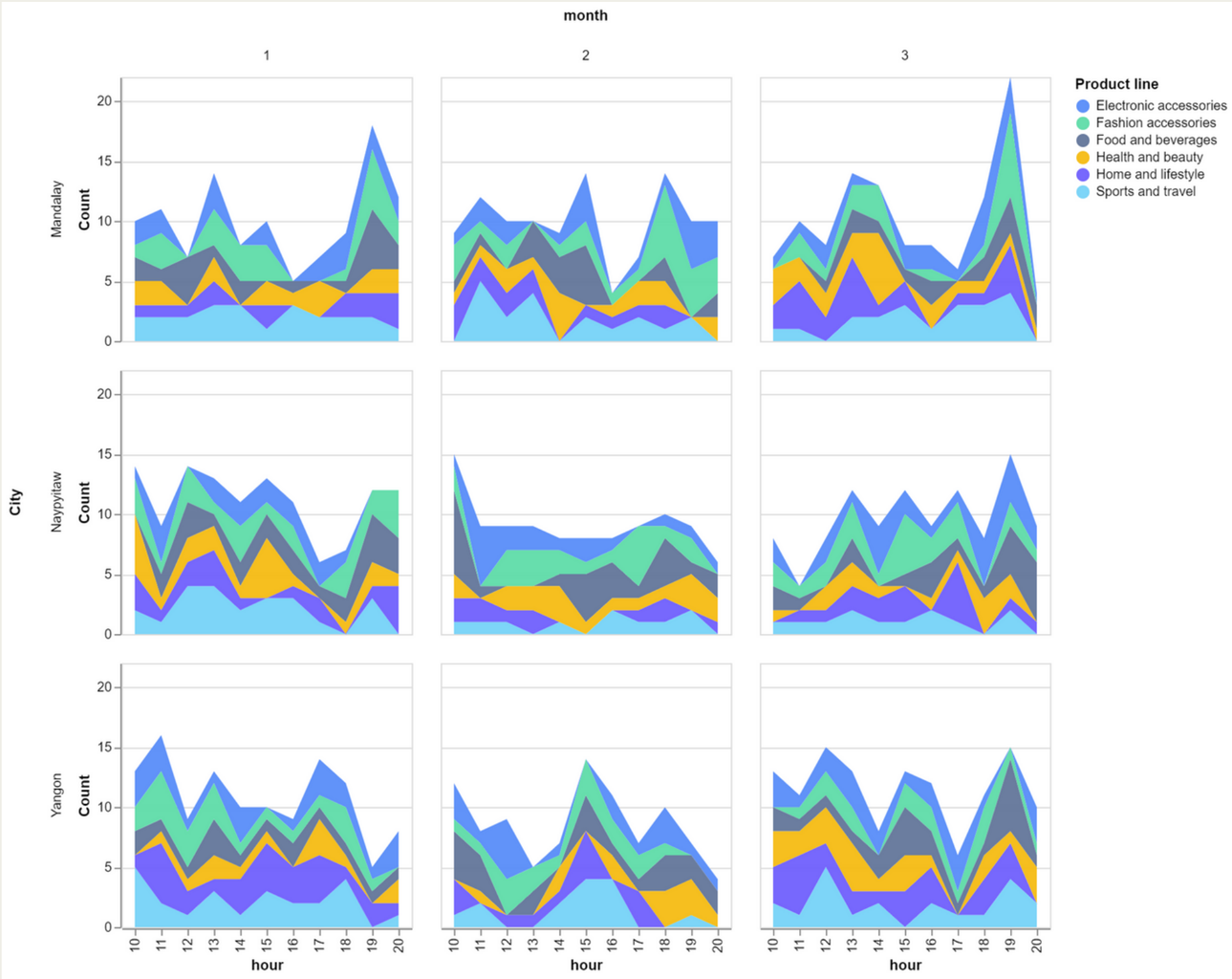
- The ratings for all customers paying via any payment methods are all similar
- Cashless payment methods are as easy to use as cash



Summary → Intro → Methodology → Results → Conclusion → Appendix

RESULTS

- The distribution of the types of goods sold with respect to the time of day does not show a discernable trend



CONCLUSION

- Sales volume of all cities are similar, but there is potential for higher sales volume, especially in Yangon
- Items with highest sales volume tend to have the highest COGS, by reducing the COGS for these items, profits will increase
  - A centralized warehouse can be built to distribute items bought from the cheapest suppliers
  - A study regarding the economical feasibility and logistical challenges should be commissioned
- The majority of customers pay via cashless methods
  - Collaboration with credit card companies and fintech companies may attract additional customers
- There are no discernable patterns in the ratings of members and-non members who pay by any methods
- There are no discernable patterns in the volume and types of goods sold throughout the day

REFERENCES

- Yangon. (2023, October 21). In Wikipedia. https://en.wikipedia.org/wiki/Yangon
- Naypyidaw. (2023, October 21). In Wikipedia. https://en.wikipedia.org/wiki/Naypyidaw
- Mandalay. (2023, October 21). In Wikipedia. https://en.wikipedia.org/wiki/Mandalay

POTENTIAL PROBLEMS WITH THE DATASET

- The decision to use this dataset was made after careful consideration among available datasets online
- However, there are oddities in this dataset such as all items as having the same gross margin percentage or the incredibly similar amount of transactions in all cities
  - Different items should have different gross margin percentage as there would be different levels of profitability (groceries should be less profitable than luxury items)
  - Larger cities would tend to have a higher sales volume
- The oddities are likely due to processing done by the uploader of the dataset, maybe to obscure confidential business information
- In a real world situation, I would attempt to verify the accuracy of the dataset, but for the purpose of demonstrating proficiency with data analysis tools or processes, this is a suitable dataset to use