DATASCI200 - Project 2

December 2024, **Ankita Mathur**

Contents

Overview:

In this report, I'll conduct a deep analysis of the brand's sales based on the key attributes such as frequency, basket size. Example: How many are single/multiple purchasers? What is the frequency of multi-purchase? Which are the typical unique quantities, revenue per order of a Basket? Based on this analysis, I'll arrive at a statistical segmentation of the brand's audience based on Revenue and key suggestions to increase the outreach of the brand.

```
df = pd.read_csv('MIQ_data_cleaned.csv')
df.columns = [i.lower().replace(' ', '_') for i in df.columns]
df.head(5)
```

	timestamp	user_id	ip_address	product_name	product_id	is_first_order	user_gender	payment_type	number_of_products	order_coupon_code	city	country_province	user_birt
0	2017-10-21 20:52:31	0	49.227.243.31	Kendrick	SE043SH31VBS	0	female	cc@braintree	1	YAY15	Kaitaia	NaN	1993-1 00:(
1	2017-10-21 20:55:12	0	111.220.172.119	Smartwatch Bradshaw Gold	MI329AC90OUB	1	male	pbi@afterpay	1	NaN	Shepparton	VIC	
2	2017-10-21 20:56:41	0	210.84.59.179	Classic Slides	SA849SH69SAK	1	female	cc@braintree	1	NaN	Essendon	VIC	
3	2017-10-21 20:59:27	0	59.167.79.119	Bonaire Flared Sleeve Tunic	SH045AA22AGR	0	female	pbi@afterpay	1	NaN	Karabar	NSW	
4	2017-10-21 21:00:56	0	1.129.107.188	Tall Tales Man Style Pants	MA146AA45RAK	0	female	cc@braintree	1	NaN	Mosman	NSW	

Handling NAs

```
print(df.isna().sum()/df.shape[0]*100) # get percentage
```

```
Shape :: (49999, 15)
NA ::
timestamp            0.000000
user_id              0.000000
ip_address           0.000000
product_name         0.000000
```

product_id 0.000000
is_first_order 0.000000
user_gender 2.348047
payment_type 0.000000
number_of_products 0.000000
order_coupon_code 76.211524
city 0.674013
country_province 8.360167
user_birthday 54.551091
country 0.000000
revenue 0.000000
dtype: float64

- From the given data we can see that the **birthday** is missing to a varying extent. There can be some cool imputation model that can be built based on the type of product bought. Given the time constraint, it is currently taken out of scope. So, we drop it.
- We drop the column **country province** as of now.
- We will impute the **coupons** with 'na' because it can unearth information that if a particular product is sku is only of promo type.
- We drop the rows where **gender** and the **city** are missing.
- With the NA removal, we have lost $(49999-48491)/49999*100 = 3.02$% of the data which is acceptable.

Cleaning Text Fields

We can see that there's discrepancy in the city names wrt casing. So, we lower and strip all the fields which can have text.
Let's do a quick check on the city and the country.

Checking Uniques and Cleaning

Let's see some of the unique value counts and see if they make some sense.

```
timestamp             47497
user_id               42757
ip_address            40621
product_name          32526
product_id            35060
is_first_order            3
user_gender               3
payment_type              5
number_of_products      258
order_coupon_code      3025
city                   4461
country                   3
revenue               12153
dtype: i
```

Before we discuss any further, let's check the unique values of and also the datatypes.

- Checking is_first_order :: ['0' '1' 'undefined']
- Checking country :: ['nz' 'au' 'undefined']
- Checking payment_type :: ['cc@braintree' 'pbi@afterpay' 'paypal@braintree' 'nopayment' 'undefined']
- Checking user_gender :: ['female' 'male' 'undefined']

So, we unearth the hidden **undefined** values present in the data. We need to get rid of them for future analysis.

NA ::
timestamp 0.000000
user_id 0.000000
ip_address 0.000000
product_name 0.030934
product_id 0.030934
is_first_order 0.030934
user_gender 0.030934
payment_type 0.030934
number_of_products 0.030934
order_coupon_code 0.030934
city 0.030934
country 0.030934
revenue 0.030934
dtype: float64

So, we find only 0.3% of the data has noise. Let's get rid of them.

- Number of products as 258** is also a bit confusing as it is highly unlikely that a person is shopping that many.
- Since the **product name** and the **product IDs** have different unique count, then we can infer that there is **no 1:1 mapping between them**, which is a bit confusing.
- Since we have lesser ids than the timestamp, we can infer that we have **repeat-customers**.
- From the above uniques, we can see that the number of IPs and user IDs are different. Although it can be the case because no IPs are static nowadays, we can just check and in case find the IPs are varying for the same customer, we **can get rid of IP** too.

Analyzing number_of_products and creating a SKU master data

Mostly all except number of products could not be casted to integer because of *invalid literal for int() with base 10: '1,1'*. This makes us rethink the level at which the data is present and decide on it.

```
df[['timestamp', 'user_id', 'ip_address', 'user_gender','city', 'country',
'product_id', 'number_of_products']].head(10)
```

So, the first thing to note here is row 6 has 2 transactions (NI537AA69RPY and NI537AA97PBM). So, essentially **each transaction shows a basket** which we may keep like this or melt it. Let's see how different are the number_of_products are.

around 264 transactions have noise and the number of product names does not map with the number of product ids. For cleaning the data, we need to get rid of them too. We normalized all the product names by taking the longest possible available description for the same item.

With the removals, we have lost $(49999−48214)/49999∗100=3.57$ % of the data till now.

So, in these examples, we can confirm that the number of products actually shows the quantity.

However, in the initial head, there seems to be one more observation. `The user_id==0` is making purchases at a gap of 2 minutes from different countries and also has different gender.

This makes us question the following things.
* Is the user_id consistent with the demographics?
* Is the country/city is based on IP?

Analyzing Customer and the Demographics

```
demographics_cols = ['user_gender', 'country']
t = df.groupby('user_id').agg(dict(zip(demographics_cols,
['nunique']*len(demographics_cols))))

# get metrics where user has more than 1 gender and country
print(f'Unique Customers :: {t.shape[0]}')
print(f'Customers who have more than 1 demographics info:: {[(c, t[t[c]>1].shape[0]) for c in
demographics_cols]}')
print(f'Transaction which comprises of these users :: {df[df.user_id.isin(list(set([j for i in
[t[t[c]>1].index for c in demographics_cols] for j in i])))].shape[0]/df.shape[0]*100:.2f}%')
```

Unique Customers :: 42542
Customers who have more than 1 demographics info:: [('user_gender', 61), ('country', 43)]
Transaction which comprises of these users :: 1.20%
Since it is only around 1.2% of the transactions, we get rid of them.

Country vs City
How can a city have 2 countries?

city	country
akaroa	2
albany	2
alexandra	2
aoraki mount cook	2
ashburton	2

Now definitely this can be true, that is a place with same name co-exist in two countries. But we have been checking on Google Maps and some of them exist only in a Australia and not New Zealand. What we come up with is, normalizing by mode.

Payments

There is something called 'no-payment'. What does it mean? Let's check the revenue earned from them.

```
f[df.payment_type == 'nopayment']['revenue'].value_counts()
```

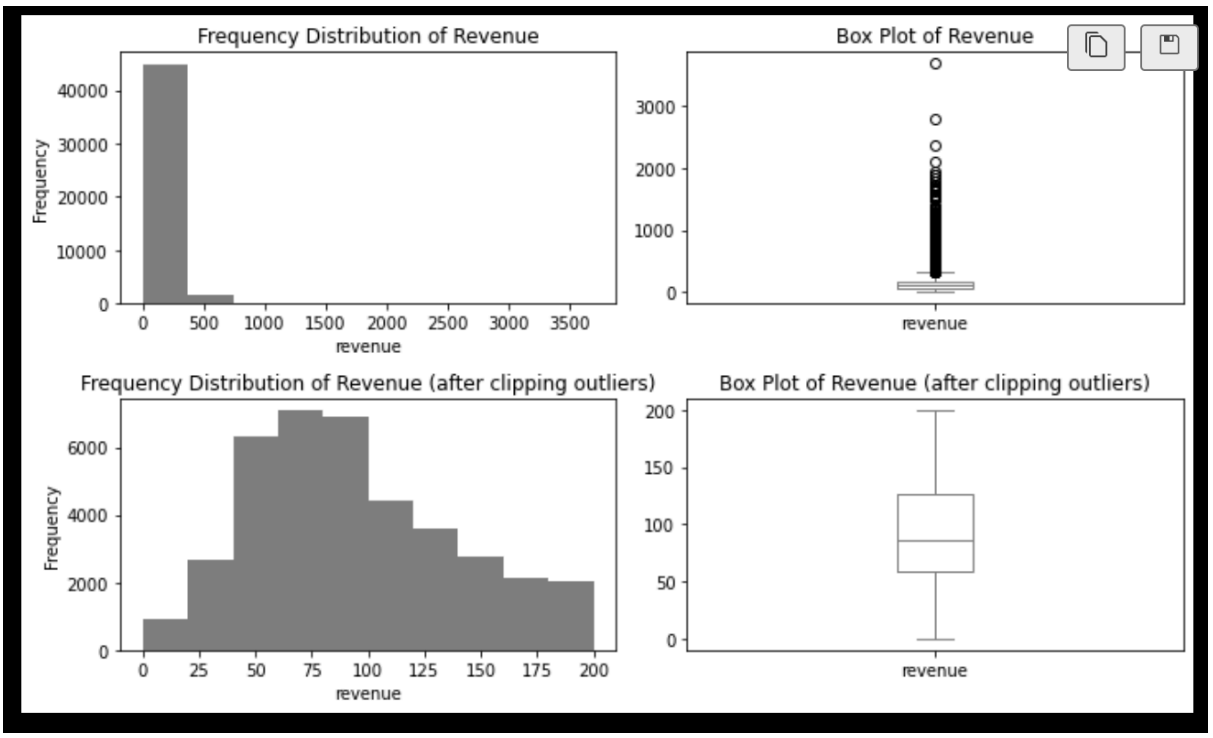
We can see that these transactions contributed nothing to revenue. Let's drop them too. We have lost $(49999−46905)/49999∗100=6.2$ % data because of all the treatments.
Hopefully, the data is clean now. Also, we already identified some of the columns for whom we can see the distributions.

Feature Engineering

```
# create some new columns from timestamp
df['timestamp'] = pd.to_datetime(df['timestamp'])
df['date'] = df.timestamp.apply(lambda x: str(x).split()[0])
df['hour'] = df.timestamp.apply(lambda x: str(x).split()[1].split(':')[0])
df['weekday'] = df.timestamp.apply(lambda x: x.weekday()) # Monday is 0, Sunday == 6
df['week_no'] = df.timestamp.apply(lambda x: x.week)
```

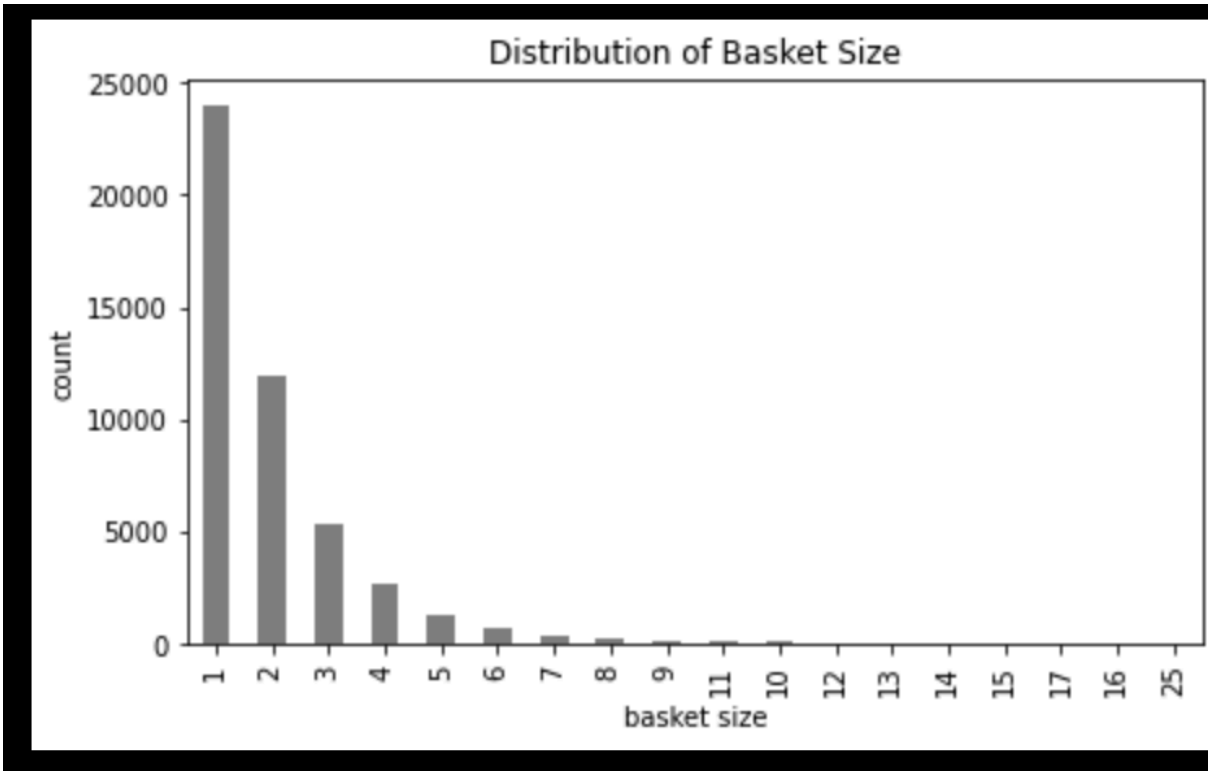
Visualizations

Revenue

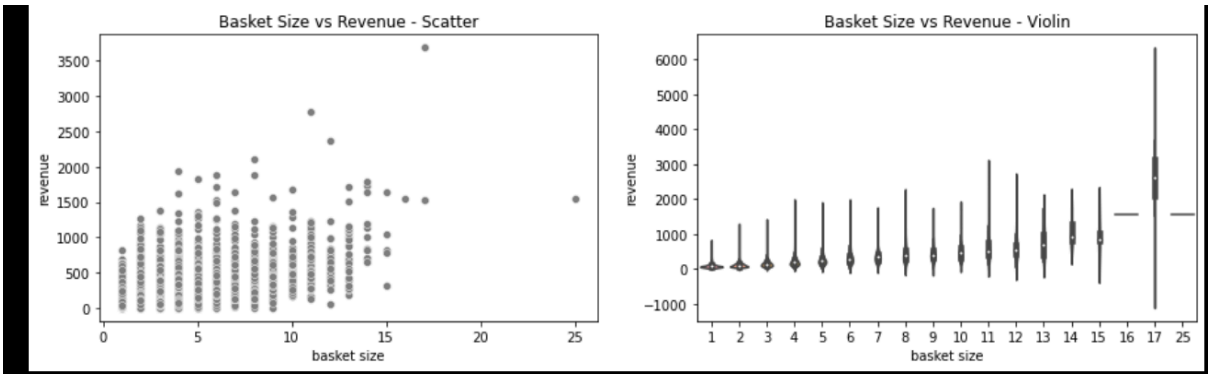


This gives rise to the question - Is the revenue related with the number of products bought (basket size)?

Basket Size



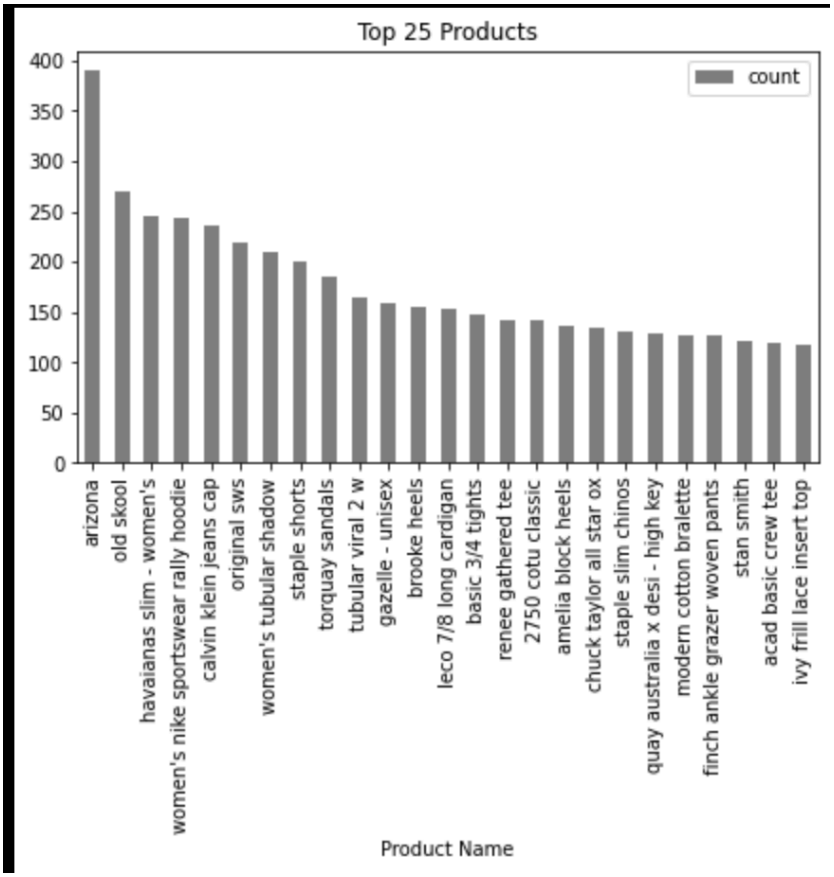
We can note that most basket size is less than 3.

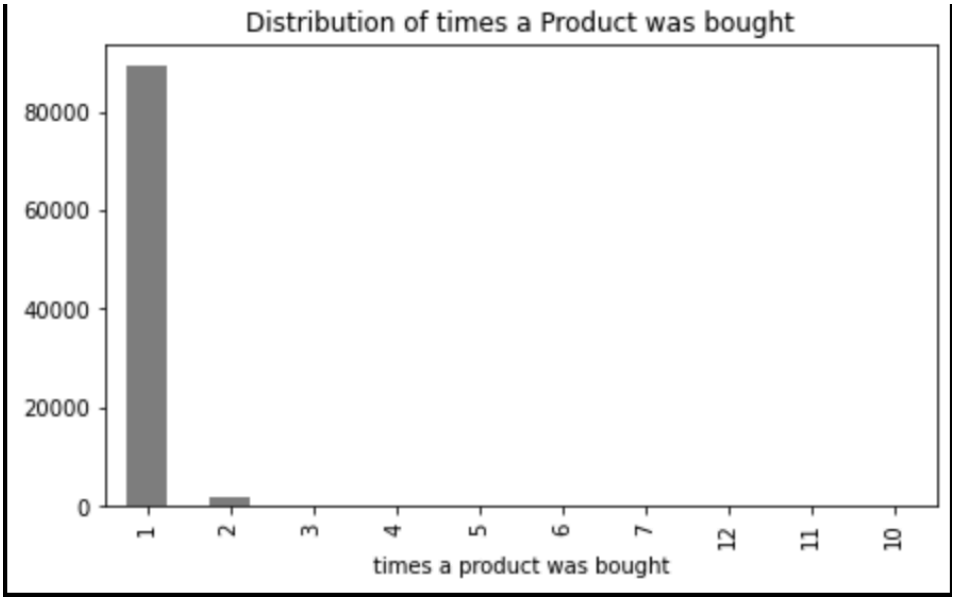


Total Sales :: 6310086.73
Average Basket Size :: 2.00
Average Revenue per transaction :: 134.53
Average Revenue per quantity of Product :: 78.24

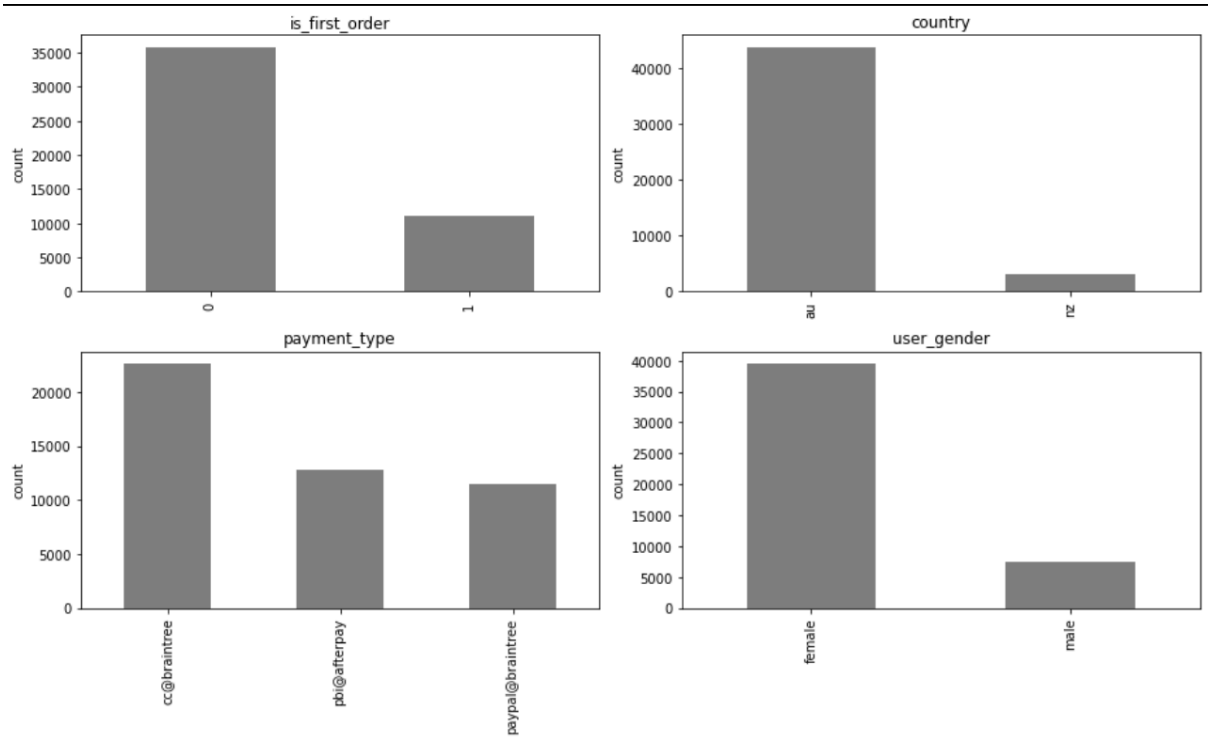
Products

Shape of SKU master :: 25578
Shape of SKU master unique SKU names :: 19423
Shape of unique SKU sold by names :: 19203
Shape of unique SKU sold by ids :: 25269
It has to be noted that 2 products can have same name, but different ids.
Overall SKU sales :: 93671 or 93671



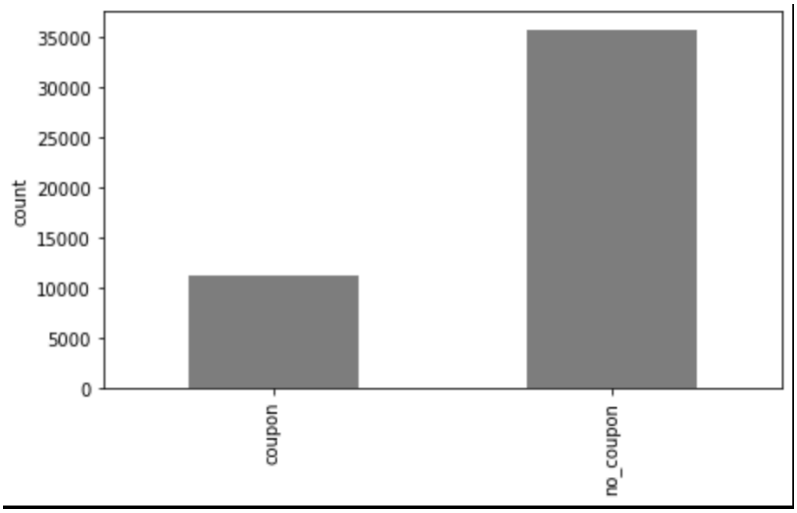


3.4 Frequencies - is_first_order, country, payment_type and user_gender

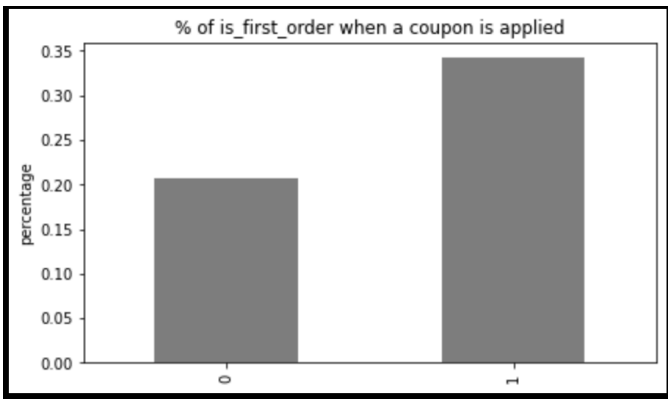


We can see that we have a number of repeat customers. Let's see the % of coupon used among the repeat customers and the new ones.

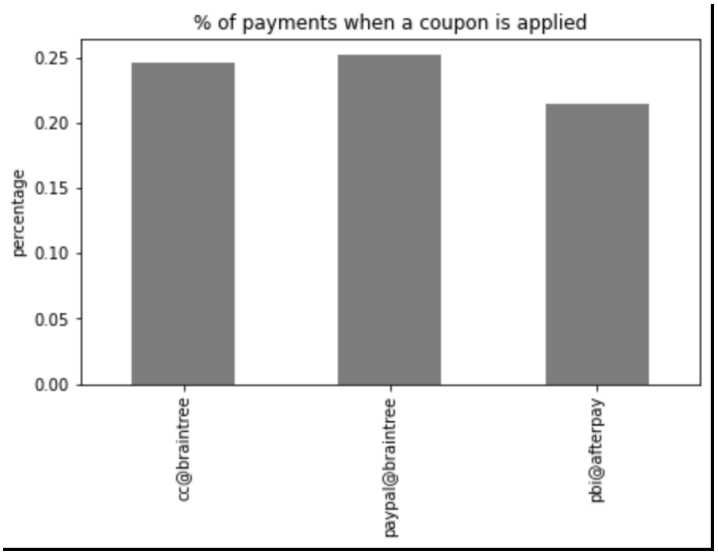
Coupons



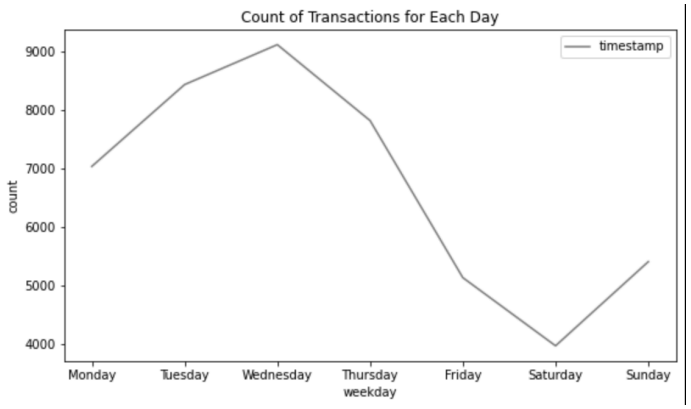
Next, we will examine number of first time orders when a coupon is used.

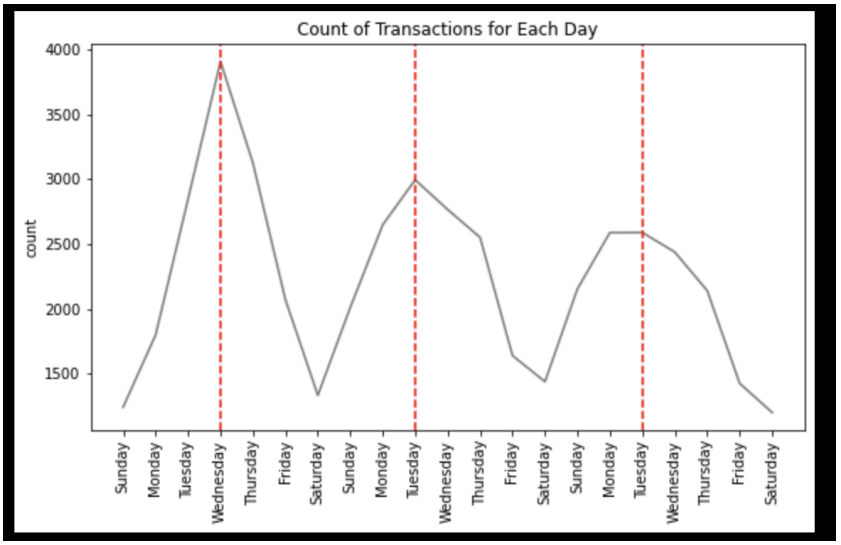


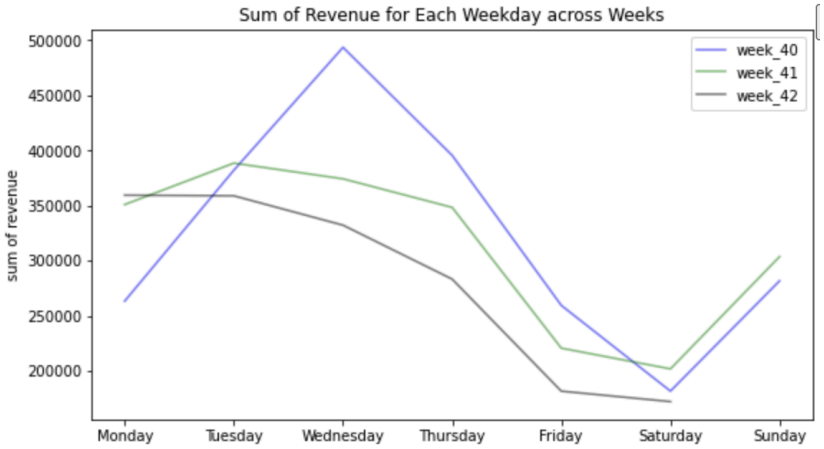
Now, examining the payment modes of the coupons.



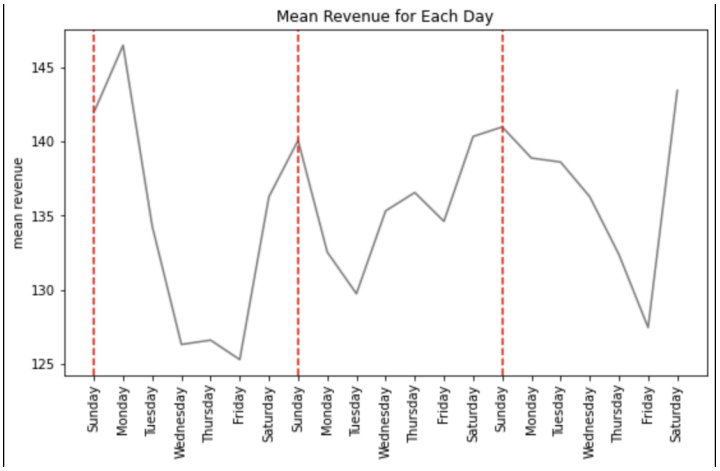
Number of Hits on Website

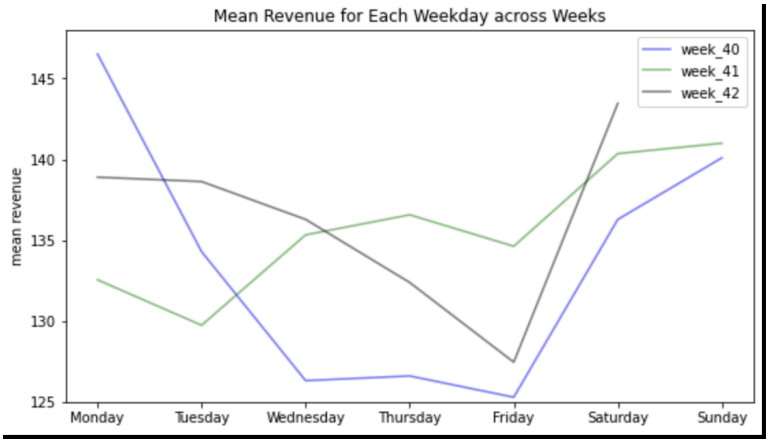




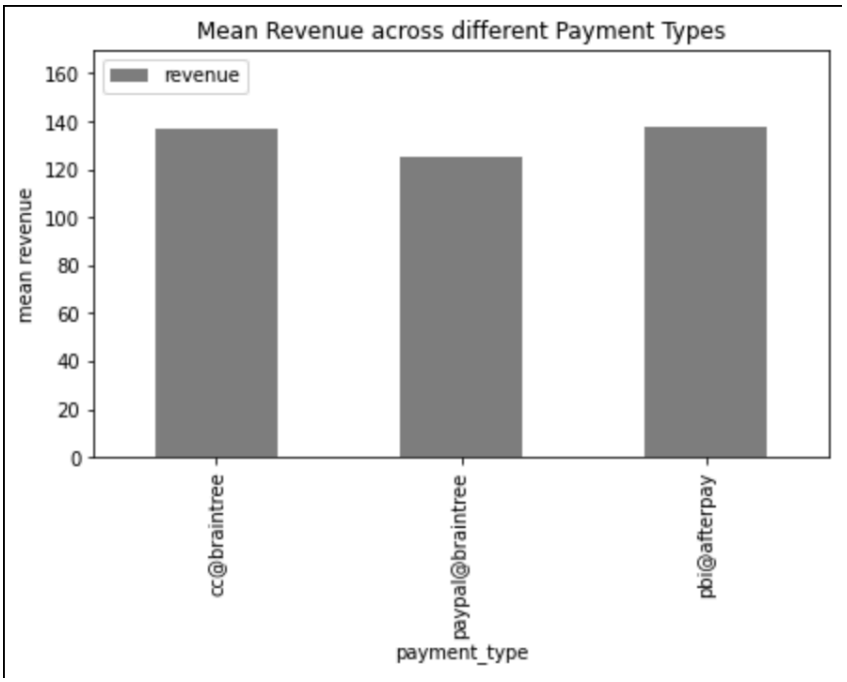


How is the Revenue affected by Days





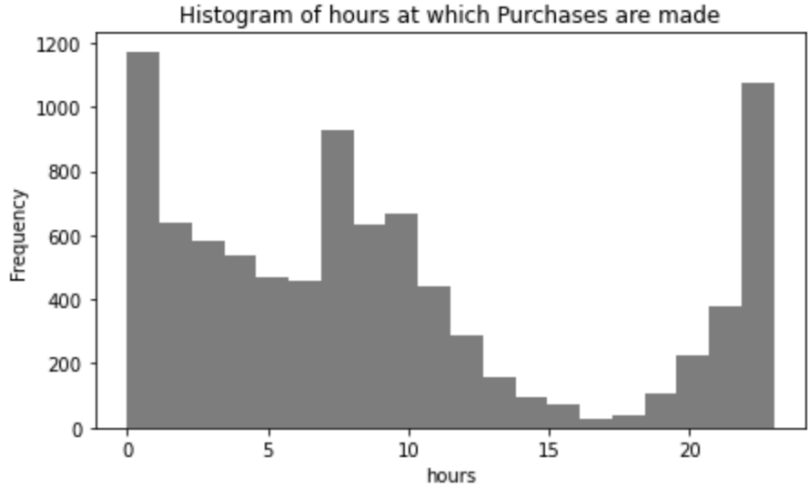
We can easily deduce that the mean revenue for the weekend sis higher than the weekdays. The trend id continued for the Week 40, 41 and 42 - forming a pattern in customer buying pattern.



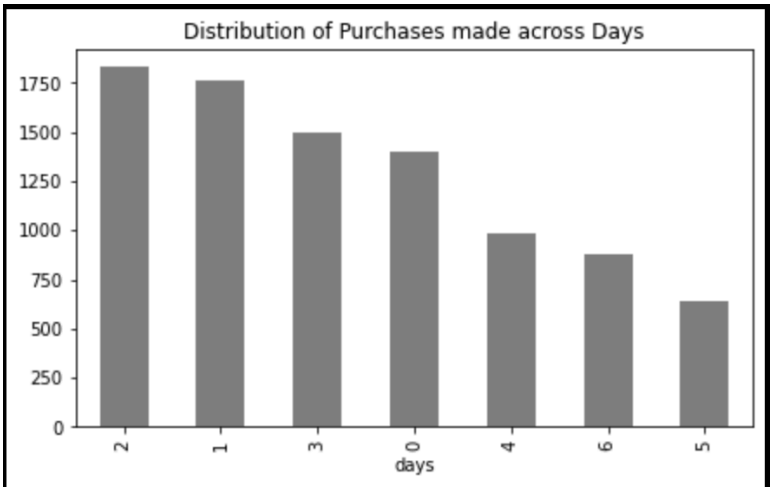
We can note that braintree has the highest mean revenues,

Multi Purchasers

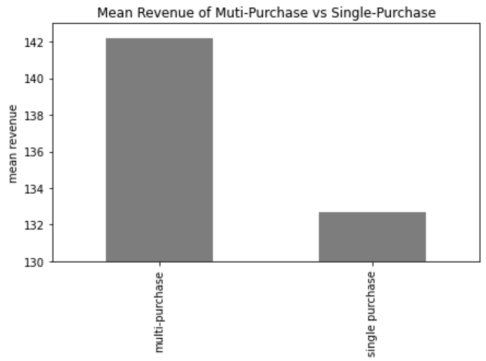
of customers who are doing multi-purchase :: 9.42
% of customers who are doing single-purchase :: 90.58

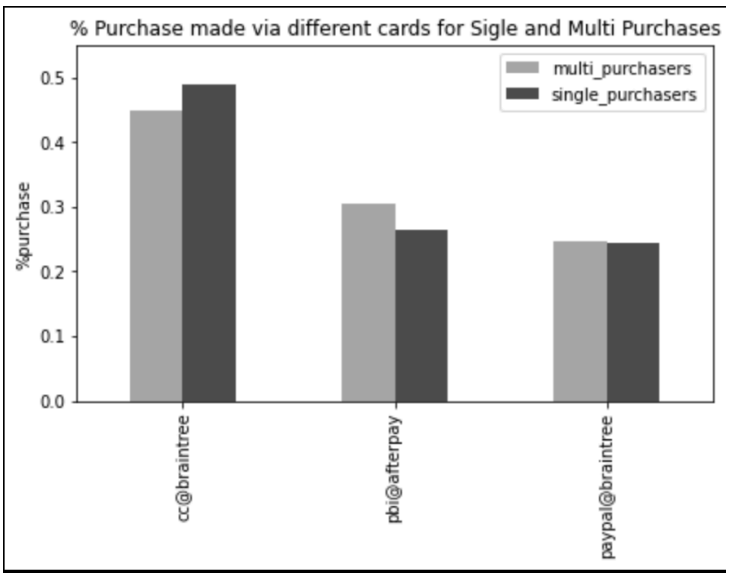


Distribution of purchases over days.

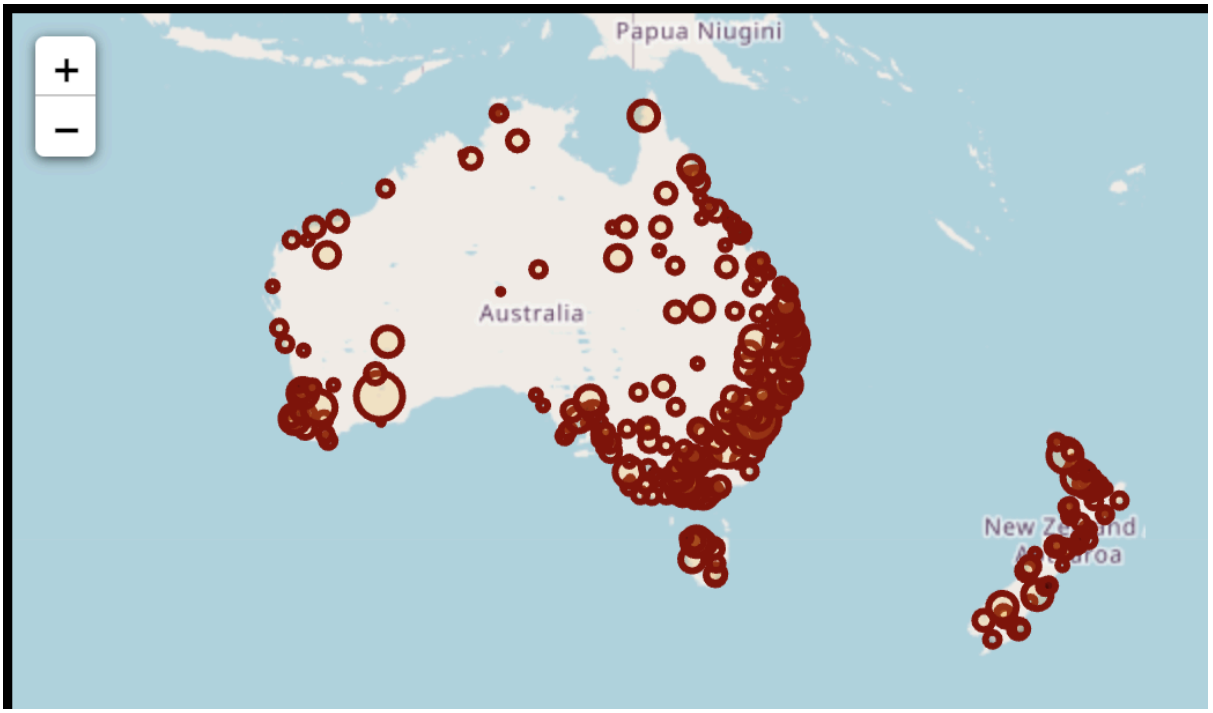


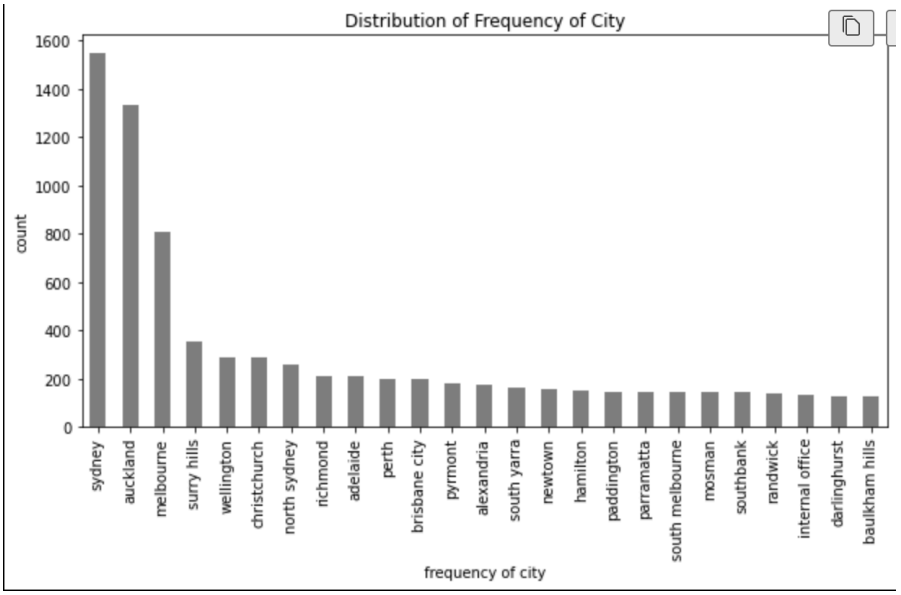
Mean Revenue of Muti-Purchase vs Single-Purchase





Geographic Importance



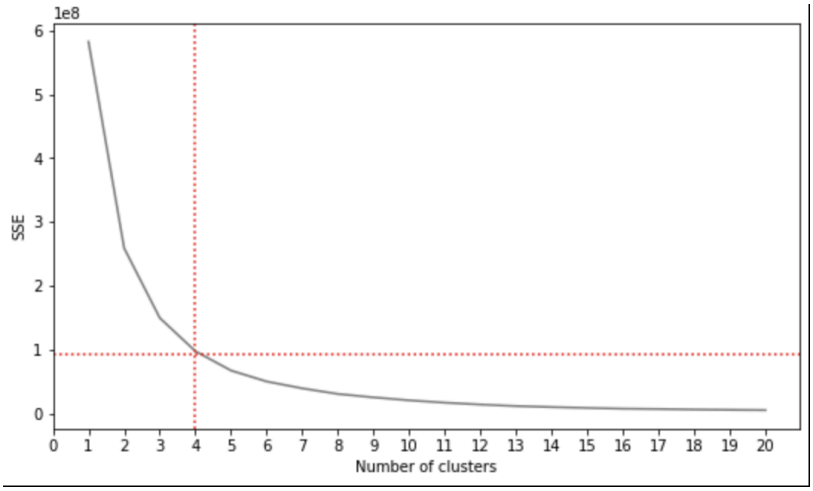


Customer Segmentation

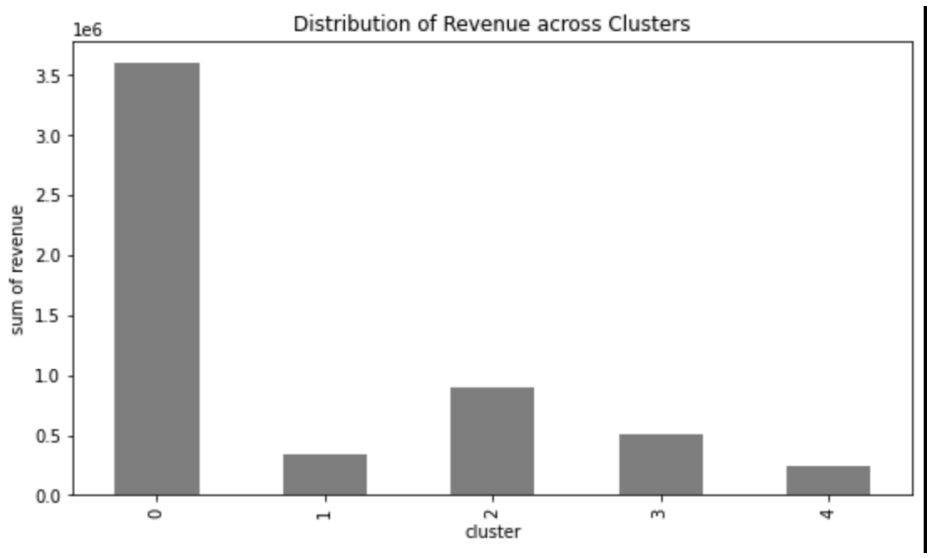
We are going to segment customers based on their last order recency, the frequency of ordering and the revenue amount. Characteristics such as country, gender will help us in providing promotions after we segment.

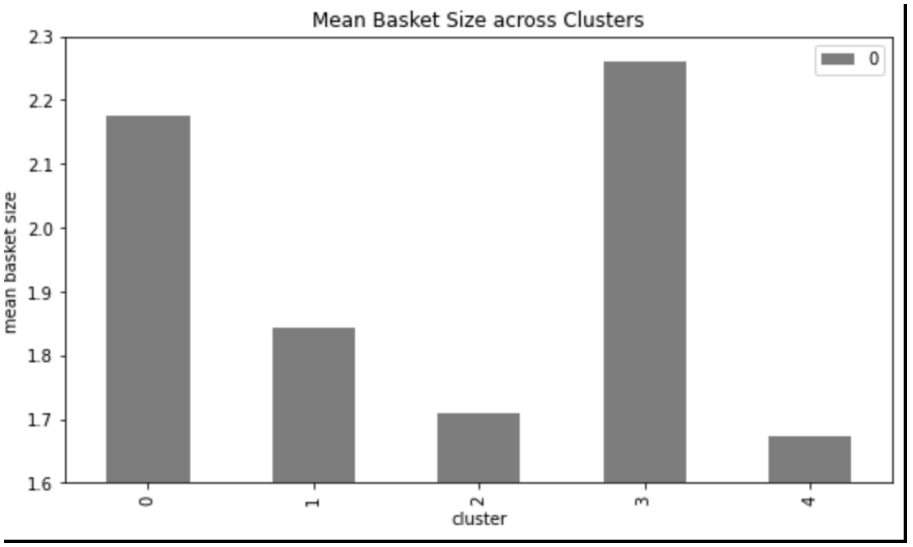
```
t = df.groupby('user_id').agg({
    'is_first_order': 'max',
    'user_gender': 'max',
    'country': 'max',
    'revenue': 'mean',
    'total_products': 'max'})

t = t.replace({'au': 0, 'nz':1, 'male': 0, 'female': 1})
t.head()
```

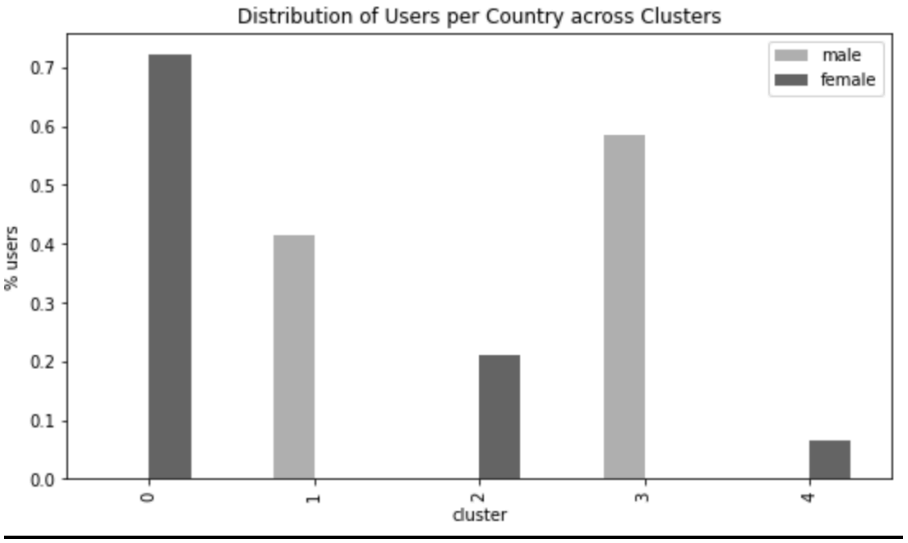
user_id	is_first_order	user_gender	country	revenue	total_products
1	0	0	0	113.55	2
2	1	1	0	81.78	2
3	0	1	0	63.63	1
4	0	1	0	81.77	1
5	0	1	0	127.18	2



From the above figure, we can see that the unscaled clustering had a lot of SSE in compared to the scaled one. We proceed with that one and use cluster as 5 because this is a good number after which the SSE almost converges.







We can see that **Cluster 0 and 3** has the users who have a considerable big basket size. **Cluster 4** has the users who does not have a considerable big basket size Here is the final customer profiles.

- **cluster 0**: has the **Australian, male, old-users** who have produce **best quality** and the **highest revenue**.
- **cluster 1**: has the **first-time, female buyers**.
- **cluster 2**: has the **first-time, male buyers.**
- **cluster 3**: has the **old female users** who have a **big basket size**.
- **cluster 4**: has the **New Zealand users** who produce **least quality revenue** and **neither have a big basket size**.

Based on our analysis and customer profiles, I propose a number of strategies on the way. However let's see the most important decisions.

- Target promotions on weekends for quick sales in the weekdays.
- Study user behaviour and page visits to recommend flagship products in the weekdays over weeks/months.
- Use the timing recommendation - (8-10 am)/(8-2 am) to remind/app-banner campaigns to attract customers.
- Can open a partnership with braintree to provide discounts who use that as a mode of payment.
- Use a better recommended (not MBA) for generating product affinity.
- Study the purchase pattern of users in cluster 0 to know their needs and recommend/remind products/alternatives in the right time.
- Give promo to cluster 1 and 2 so that they are attracted to the website.
- Recommend related products that for cluster 3 to make them purchase more.
- Study the behaviour of cluster 4 and try to engage them more by campaigning the brand in New Zealand and with promotional discounts.