EDA Overview
EDA processes
EDA questions list
EDA Insights (Summary)
Contribution

Capstone EDA notebook

Group 7: Richard Lim, Varun Selvam, Nikita Muddapati, Meenakshi Hariharan

2025-02-23

EDA Overview

Background:

SCCU(Swire Coca-Cola United States) tries to optimize logistics by transitioning customers selling below a specific annual volume to an Alternate Route to Market (ARTM). There is an annual 400 gallons volume threshold used to distinguish the customers between the direct delivery route and ARTM. However, SCCU is looking for a more cost-efficient strategy to decide new threshold for optimizing logistics which is driving better operational efficiency and more revenues.

Requirement:

1. The analysis will focus on classifying which customers must be included in ARTM or Direct route, and which volume threshold would be optimal to decide for the classification.
2. The analysis will focus on two key customer segments.

- 1st Group: Local Market Partners that buy fountains only: Customers who buy only fountain drinks and no CO2, cans, or bottles.
- 2nd Group: This group includes all customers, regardless of whether they are local market partners or not, and includes those purchasing CO2, cans, bottles, or fountain drinks.

Questions:

- What factors or characteristics distinguish customers with annual sales exceeding the determined volume threshold from those below this threshold?
- How can SCCU uses historical sales data, or other Customer Characteristics to predict which ARTM customers have the potential to grow beyond the volume threshold annually?
- How can these insights be integrated into the routing strategy to support long-term growth while maintaining logistical efficiency?
- What levers can be employed to accelerate volume and share growth at growth-ready, high-potential customers?

EDA processes

1. Import libraries

```
# import libraries
library(tidyverse)
library(janitor)
library(skimr)
library(psych)
library(glue)
library(here)
library(readxl)
```

2. Import Datasets

- There are 4 datasets used for the analysis, which contains address, customer profile, delivery cost, and transaction history.

```
# import datasets
address_df<- read_csv(here("Dataset",
     "customer_address_and_zip_mapping.csv"))
profile_df <- read_csv(here("Dataset","customer_profile.csv"))
delivery_cost_df <- read_xlsx(here("Dataset","delivery_cost_data.xlsx"))
trans_df <- read_csv(here("Dataset","transactional_data.csv"))
```

3. Dataset Profiling & Exploration

3-1. Address Dataset Profile

Variables can be described as below.

- Zip: ZIP code for the location.
- Full address: Full address information seperated by , including city, state, county, region, and latitude/longitude.
- Full address is listed in the order of zipcode, city, state full name, state acronym, county, FIPS codes, latitude, longitude

```
sample_n(address_df, 10)
```

zip	full address
<dbl>	<chr>
2534	02534,Cataumet,Massachusetts,MA,Barnstable,1,41.6694,-70.6234
42131	42131,Etoile,Kentucky,KY,Barren,9,36.8134,-85.9173
2375	02375,South Easton,Massachusetts,MA,Bristol,5,42.0257,-71.0988
2641	02641,East Dennis,Massachusetts,MA,Barnstable,1,41.7426,-70.162
2143	02143,Somerville,Massachusetts,MA,Middlesex,17,42.3829,-71.1028
42275	42275,Roundhill,Kentucky,KY,Edmonson,61,37.256,-86.407
66012	66012,Bonner Springs,Kansas,KS,Wyandotte,209,39.0672,-94.9227
67753	67753,Rexford,Kansas,KS,Thomas,193,39.4267,-100.7461
1851	01851,Lowell,Massachusetts,MA,Middlesex,17,42.6315,-71.3329
2044	02044,Hingham,Massachusetts,MA,Plymouth,23,42.2418,-70.8898

1-10 of 10 rows

3-2. Customer Profile Dataset Profile

Variables can be described as below.

- Customer Number: Unique identifying number of customer
- Primary Group Number: The group number of which customer mainly belongs to
- Frequent Order Type: The order type that customer mainly uses
- First Delivery Date: The date that first delivery was made
- On Boarding Date: The date that first transaction was made
- Cold Drink Channel: General channel category for cold drink purchases (e.g., “DINING”)
- Trade Channel: Detailed channel classification (e.g., “OTHER DINING & BEVERAGE”)
- Sub Trade Channel: Sub-classification within the trade channel (e.g., “OTHER DINING”)
- Local Market Partner: Whether customer is local market partner (True or False)
- CO2 Customer: Whether customer purchases CO2 product or not (True or False)
- Zip Code: customer address zip code which is connected with Zip variable in `address_df`

```
sample_n(profile_df,10)
```

CUSTOMER_NUMBER	PRIMARY_GROUP_NUMBER	FREQUENT_ORDER_TYPE	FIRST_DELIVERY_DATE
<dbl>	<dbl>	<chr>	<chr>
501298963	NA	SALES REP	12/2/2021
600554657	NA	SALES REP	4/1/2017
600076325	NA	SALES REP	3/3/2016
600081091	NA	SALES REP	3/9/2016
600065058	1685	SALES REP	4/30/2018
501648539	NA	SALES REP	5/23/2024
501677771	NA	SALES REP	7/12/2024
501208148	265	SALES REP	6/4/2021
500391024	NA	SALES REP	3/14/2018
600685400	405	SALES REP	5/1/2017

1-10 of 10 rows | 1-4 of 11 columns

3-3. Delivery Cost Dataset Profile

Variables can be described as below.

- Cold Drink Channel: The main functional category of commerce
- Vol Range: The annual volume range of products
- Applicable to: which category of products that volumes apply to
- Median Delivery Cost: Median cost of delivery per cost type
- Cost type: the unit by measuring the cost
    - Fountain → Measured in gallons (Per Gallon)
    - Bottles and Cans → Measured in cases (Per Case).

```
sample_n(delivery_cost_df,10)
```

Cold Drink Channel <chr>	Vol Range <chr>	Applicable To <chr>	Median Delivery Cost <dbl>	Cost Type <chr>
ACCOMMODATION	1350+	Fountain	0.4226513	Per Gallon
WORKPLACE	1200 - 1349	Bottles and Cans	0.6666636	Per Case
EVENT	900 - 1049	Fountain	1.2977950	Per Gallon
WORKPLACE	300 - 449	Fountain	1.8754015	Per Gallon
GOODS	450 - 599	Fountain	1.6121354	Per Gallon
GOODS	0 - 149	Fountain	4.6197646	Per Gallon
PUBLIC SECTOR	600 - 749	Bottles and Cans	2.5093015	Per Case
GOODS	450 - 599	Bottles and Cans	3.8644162	Per Case
WORKPLACE	900 - 1049	Fountain	1.0005297	Per Gallon
PUBLIC SECTOR	150 - 299	Bottles and Cans	4.1584496	Per Case

1-10 of 10 rows

3-4. Transaction Dataset Profile

Variables can be described as below.

- Transaction Date: Date of the transaction (YYYY-MM-DD format).
- Week: Week number of the year when the transaction occurred.
- Year: Year of the transaction occurred.
- Customer Number: Unique identifier for the customer.
- Order Type: Type of order placed
- Ordered Cases: The amount of cases that ordered
- Loaded Cases: The amount of cases that loaded in the truck
- Delivered Cases: The amount of cases that delivered to the customer
- Ordered Gallons: The amount of gallons that ordered
- Loaded Gallons: The amount of gallons that loaded in the truck
- Delivered Gallons: The amount of gallons that delivered to the customer
    - **Information 1**: One standard physical case equating to one gallon, allowing for a direct summation of cases and gallons.
    - **Information 2**: Negative delivered volume must be considered as a return.

```
sample_n(trans_df,10)
```

TRANSACTION_DATE <chr>	WEEK <dbl>	YEAR <dbl>	CUSTOMER_NUMBER <dbl>	ORDER_TYPE <chr>	ORDERED_CASES <dbl>
8/13/2024	33	2024	600069292	CALL CENTER	5
11/13/2023	46	2023	501297162	MYCOKE LEGACY	29
6/25/2024	26	2024	600265879	CALL CENTER	0
3/24/2023	12	2023	501058194	MYCOKE LEGACY	0
10/18/2023	42	2023	501325576	SALES REP	0
10/19/2023	42	2023	600076952	SALES REP	5
1/19/2023	3	2023	501081031	SALES REP	24
10/1/2024	40	2024	501645156	SALES REP	9
4/5/2024	14	2024	501524017	MYCOKE LEGACY	0
4/7/2023	14	2023	600076783	SALES REP	0

1-10 of 10 rows | 1-6 of 11 columns

4. Skimming of Dataset

```
skim(address_df)
```

Data summary

Name	address_df
Number of rows	1801
Number of columns	2

_____

Column type frequency:	
character	1
numeric	1
———————————	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
full address	0	1	45	73	0	1801	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
zip	0	1	28919.81	25588.64	1001	2153	21634	42440	71483	

```
skim(profile_df)
```

Data summary

Name	profile_df
Number of rows	30478
Number of columns	11
———————————	
Column type frequency:	
character	6
logical	2
numeric	3
———————————	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
FREQUENT_ORDER_TYPE	0	1	3	13	0	6	0
FIRST_DELIVERY_DATE	0	1	8	10	0	2401	0
ON_BOARDING_DATE	0	1	8	10	0	6487	0
COLD_DRINK_CHANNEL	0	1	5	13	0	9	0
TRADE_CHANNEL	0	1	6	28	0	26	0
SUB_TRADE_CHANNEL	0	1	4	27	0	48	0

Variable type: logical

skim_variable	n_missing	complete_rate	mean	count
LOCAL_MARKET_PARTNER	0	1	0.90	TRU: 27355, FAL: 3123
CO2_CUSTOMER	0	1	0.39	FAL: 18496, TRU: 11982

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	h
CUSTOMER_NUMBER	0	1.0	538301800.92	47950644.47	500245678	501164306	501573995	600075795	600975408	
PRIMARY_GROUP_NUMBER	18196	0.4	2779.85	2608.64	4	444	1892	4488	9999	
ZIP_CODE	0	1.0	30252.25	25953.08	1001	2155	21771	42762	71483	

```
skim(delivery_cost_df)
```

Data summary

Name	delivery_cost_df

Number of rows	160
Number of columns	5
_____	
Column type frequency:	
character	4
numeric	1
_____	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Cold Drink Channel	0	1	5	13	0	8	0
Vol Range	0	1	5	11	0	10	0
Applicable To	0	1	8	16	0	2	0
Cost Type	0	1	8	10	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Median Delivery Cost	0	1	2.6	1.71	0.37	1.33	2.24	3.47	8.59	

```
skim(trans_df)
```

Data summary

Name	trans_df
Number of rows	1045540
Number of columns	11
_____	
Column type frequency:	
character	2
numeric	9
_____	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
TRANSACTION_DATE	0	1	8	10	0	723	0
ORDER_TYPE	0	1	3	13	0	7	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
WEEK	0	1	26.23	14.52	1.0	14	26	38.00	52.00
YEAR	0	1	2023.50	0.50	2023.0	2023	2023	2024.00	2024.00
CUSTOMER_NUMBER	0	1	546643776.32	49426585.56	500245678.0	501091920	501548213	600080939.00	600975408.00
ORDERED_CASES	0	1	26.85	126.76	0.0	0	7	18.50	8479.89
LOADED_CASES	0	1	25.92	122.79	0.0	0	7	18.00	8171.56
DELIVERED_CASES	0	1	25.13	121.52	-3132.0	0	6	17.33	8069.48
ORDERED_GALLONS	0	1	9.87	26.47	0.0	0	0	12.50	2562.50
LOADED_GALLONS	0	1	9.60	25.65	0.0	0	0	12.50	2562.50
DELIVERED_GALLONS	0	1	9.21	25.18	-1792.5	0	0	12.50	2292.50

5. Checking NA per variable

```
colSums(is.na(address_df))
```

```
##          zip full address
##            0            0
```

```
colSums(is.na(profile_df))
```

```
##      CUSTOMER_NUMBER PRIMARY_GROUP_NUMBER  FREQUENT_ORDER_TYPE
##                    0               18196                    0
##  FIRST_DELIVERY_DATE     ON_BOARDING_DATE   COLD_DRINK_CHANNEL
##                    0                    0                    0
##        TRADE_CHANNEL    SUB_TRADE_CHANNEL LOCAL_MARKET_PARTNER
##                    0                    0                    0
##         CO2_CUSTOMER             ZIP_CODE
##                    0                    0
```

```
colSums(is.na(delivery_cost_df))
```

```
##   Cold Drink Channel            Vol Range        Applicable To
##                    0                    0                    0
## Median Delivery Cost            Cost Type
##                    0                    0
```

```
colSums(is.na(trans_df))
```

```
##  TRANSACTION_DATE             WEEK             YEAR   CUSTOMER_NUMBER
##                 0                0                0                 0
##        ORDER_TYPE     ORDERED_CASES      LOADED_CASES   DELIVERED_CASES
##                 0                0                0                 0
##   ORDERED_GALLONS    LOADED_GALLONS DELIVERED_GALLONS
##                 0                0                0
```

- `PRIMARY_GROUP_NUMBER` has a 18196 missing values, which takes up 60% of `profile_df` dataset.

EDA questions list

- How many customers are partnered with Local Market Partners out of the entire customers?
- How many customers are purchasing C02 products out of entire customers?
- Which number can we extract out of transaction history?
- How many customers belongs to the direct route based on the original volume threshold? And how many customers belong to the ARTM based on the original volume threshold?
- Which customer characteristics have brought more profits from given transaction data?
    - CO2 vs Non-CO2
    - Local Market Partners vs Non-Local Market Partners
    - Cold Drink Channel
    - Frequent Order Type
- How many customers belongs to the Local Market Partners that buy fountains only? (Group Segment 1)
- How many Customers moved above and below the Threshold from 2023 to 2024?
- What is the Net change in customers moving between threshold categories? (Low Volume, Medium Volume, High Volume)
    - How many New customers appeared in 2024 compared to 2023?
- What percentage of customers upgraded or downgraded between categories?
    - Do customers who move to higher segments tend to have consistent increases in order volume or are they sporadic?
    - Are there specific patterns in customer order frequency that indicate a transition between volume categories?
- What are the key patterns in customer order volume reduction from 2023 to 2024.
- Among customers who reduced their order volume , what is the average percentage drop?

1. The summary table of Local Market Partner Customer

```
# the distribution of local market partner customers out of entire customers
table(profile_df$LOCAL_MARKET_PARTNER)
```

```
##
## FALSE  TRUE
##  3123 27355
```

```
round(prop.table(table(profile_df$LOCAL_MARKET_PARTNER)),2)
```

```
##
## FALSE  TRUE
##   0.1   0.9
```

Approximately, 90% of listed customers belong to the local market partners, which indicates that they are smaller, regionally focused customers who serve their local communities. They tend to show their reliance on local market dynamics and consistent purchasing patterns.

2. The summary table of of CO2 customer

```
# the distribution of CO2 customers out of entire customers
table(profile_df$CO2_CUSTOMER)
```

```
##
## FALSE  TRUE
## 18496 11982
```

```
round(prop.table(table(profile_df$CO2_CUSTOMER)),2)
```

```
##
## FALSE  TRUE
##  0.61  0.39
```

Approximately, 40% of listed customer belongs to the CO2 customer, which represents that they have purchased carbon dioxide materials.

3. Total number of transaction

- Total number of customer
- Total volume of cases
- Total volume of gallons
- Total transaction period

```
trans_df %>%
  summarise(customer_n = n_distinct(CUSTOMER_NUMBER))
```

customer_n
<int>
30322

1 row

```
trans_df %>%
  summarise(case_volume = sum(ORDERED_CASES),
            gallon_volume = sum(ORDERED_GALLONS),
            total_volume = case_volume + gallon_volume)
```

case_volume	gallon_volume	total_volume
<dbl>	<dbl>	<dbl>
28074470	10323337	38397807

1 row

```
max(as.Date(trans_df$TRANSACTION_DATE, format="%m/%d/%Y"))
```

```
## [1] "2024-12-31"
```

```
min(as.Date(trans_df$TRANSACTION_DATE, format="%m/%d/%Y"))
```

```
## [1] "2023-01-01"
```

30322 customers have transacted 28,074,470 cases and 10,323,337 gallons (total 38,397,807 units) with SCCU from 1/1/2023 to 12/31/2024. (2 years)

4. Transaction history per customer

```
trans_history <-
trans_df %>%
  mutate(TRANSACTION_DATE = as.Date(TRANSACTION_DATE, format="%m/%d/%Y")) %>%
  #mutate(CUSTOMER_NUMBER = as.integer(CUSTOMER_NUMBER)) %>%
  group_by(CUSTOMER_NUMBER) %>%
  summarise(
            FIRST_TRANSACTION_DATE = min(TRANSACTION_DATE),
            LAST_TRANSACTION_DATE = max(TRANSACTION_DATE),
            TRANS_DAYS = LAST_TRANSACTION_DATE - FIRST_TRANSACTION_DATE + 1,
            TRANS_COUNT = n(),
            TRANS_COUNT_2023 = sum((year(TRANSACTION_DATE) == 2023)),
            TRANS_COUNT_2024 = sum((year(TRANSACTION_DATE) == 2024)),
            ANNUAL_VOLUME_CASES_2023 = sum((year(TRANSACTION_DATE) == 2023) * ORDERED_CASES, na.rm = TRUE),
            ANNUAL_VOLUME_GALLON_2023 = sum((year(TRANSACTION_DATE) == 2023) * ORDERED_GALLONS, na.rm = TRUE),
            ANNUAL_VOLUME_CASES_2024 = sum((year(TRANSACTION_DATE) == 2024) * ORDERED_CASES, na.rm = TRUE),
            ANNUAL_VOLUME_GALLON_2024 = sum((year(TRANSACTION_DATE) == 2024) * ORDERED_GALLONS, na.rm = TRUE),
            ANNUAL_VOLUME_2023 = sum((year(TRANSACTION_DATE) == 2023) * (ORDERED_CASES + ORDERED_GALLONS), na.rm = TRUE),
            AVG_ORDER_VOLUME_2023 = ANNUAL_VOLUME_2023 / TRANS_COUNT_2023,
            ANNUAL_VOLUME_2024 = sum((year(TRANSACTION_DATE) == 2024) * (ORDERED_CASES + ORDERED_GALLONS), na.rm = TRUE),
            AVG_ORDER_VOLUME_2024 = ANNUAL_VOLUME_2024 / TRANS_COUNT_2024,
            CHANGED_VOLUME = ANNUAL_VOLUME_2024 - ANNUAL_VOLUME_2023,
            PERCENT_CHANGE = round(CHANGED_VOLUME/ANNUAL_VOLUME_2023,2) * 100,
            THRESHOLD_2023 = ifelse(ANNUAL_VOLUME_2023 >= 400, 'above', 'below'),
            THRESHOLD_2024 = ifelse(ANNUAL_VOLUME_2024 >= 400, 'above', 'below'),
  ) %>%
  ungroup()

trans_history
```

CUSTOMER_NUMBER <dbl>	FIRST_TRANSACTION_DATE <date>	LAST_TRANSACTION_DATE <date>	TRANS_DAYS <drtn>
500245678	2023-01-09	2024-11-20	682 days
500245685	2023-01-06	2024-08-16	589 days
500245686	2023-03-07	2024-12-17	652 days
500245687	2023-02-06	2024-10-28	631 days
500245689	2023-01-13	2024-12-26	714 days
500245690	2023-01-26	2024-12-23	698 days
500245695	2023-01-04	2024-12-04	701 days
500245698	2023-01-13	2024-12-23	711 days
500245701	2023-01-03	2024-05-13	497 days
500245704	2023-01-10	2024-12-26	717 days

1-10 of 10,000 rows | 1-4 of 19 columns

Previous **1** 2 3 4 5 6 … 1000Next

```
colSums(is.na(trans_history))
```

```
##           CUSTOMER_NUMBER    FIRST_TRANSACTION_DATE     LAST_TRANSACTION_DATE
##                         0                         0                         0
##                TRANS_DAYS               TRANS_COUNT          TRANS_COUNT_2023
##                         0                         0                         0
##          TRANS_COUNT_2024  ANNUAL_VOLUME_CASES_2023 ANNUAL_VOLUME_GALLON_2023
##                         0                         0                         0
##  ANNUAL_VOLUME_CASES_2024 ANNUAL_VOLUME_GALLON_2024        ANNUAL_VOLUME_2023
##                         0                         0                         0
##     AVG_ORDER_VOLUME_2023        ANNUAL_VOLUME_2024     AVG_ORDER_VOLUME_2024
##                      4270                         0                       721
##            CHANGED_VOLUME            PERCENT_CHANGE            THRESHOLD_2023
##                         0                       137                         0
##            THRESHOLD_2024
##                         0
```

- calculation of ANNUAL_VOLUME = AVG_ORDER_VOLUME (Order Volume) * TRANS_COUNT (Frequency) for certain year (2023 vs 2024)

```
# 2023 above vs below threshold
table(trans_history$THRESHOLD_2023)
```

```
##
## above below
##  7745 22577
```

```
prop.table(table(trans_history$THRESHOLD_2023))
```

```
##
##     above     below
## 0.2554251 0.7445749
```

```
# 2024 above vs below threshold
table(trans_history$THRESHOLD_2024)
```

```
##
## above below
##  7867 22455
```

```
prop.table(table(trans_history$THRESHOLD_2024))
```

```
##
##     above     below
## 0.2594486 0.7405514
```

- approximately, 25% of customers are above the original volume threshold (400 annual volume), whereas 75% of customers remain below the threshold in both 2023 and 2024. It appears that the proportion of customer group haven't changed much between 2 years.

```
thres_change_customer <-
trans_history %>%
  filter(THRESHOLD_2023 != THRESHOLD_2024)

thres_change_customer
```

CUSTOMER_NUMBER <dbl>	FIRST_TRANSACTION_DATE <date>	LAST_TRANSACTION_DATE <date>	TRANS_DAYS <drtn>
500245698	2023-01-13	2024-12-23	711 days
500245791	2023-01-10	2024-12-24	715 days
500245851	2023-10-11	2023-10-17	7 days
500245864	2023-02-23	2024-08-23	548 days
500246054	2023-01-13	2023-12-29	351 days
500249461	2023-01-10	2024-12-17	708 days
500263851	2023-03-03	2024-12-20	659 days
500264574	2023-01-06	2024-12-27	722 days
500264805	2023-01-12	2024-12-19	708 days
500266407	2023-01-11	2024-12-18	708 days

1-10 of 2,378 rows | 1-4 of 19 columns

Previous 1 2 3 4 5 6 … 238 Next

```
table(thres_change_customer$THRESHOLD_2023, thres_change_customer$THRESHOLD_2024)
```

```
##
##         above below
##   above     0  1128
##   below  1250     0
```

```
round(prop.table(table(thres_change_customer$THRESHOLD_2023, thres_change_customer$THRESHOLD_2024)),2)
```

```
##
##         above below
##   above  0.00  0.47
##   below  0.53  0.00
```

However, when we get into the depth, 2,378 (8%) customers experienced a change in volume based on the original volume threshold from 2023 to 2024 out of 30,322 total customers. Among them, 1,250 customers (around 4%) exceeded the threshold in 2024 from below threshold status, whereas 1,128 (around 4%) customers drops below the threshold.

5. Volume changes comparison

5-1. Changed volume statistics

```
# total customer growth statistics
trans_history %>%
  summarise(AVG_CHANGE_VOL = mean(CHANGED_VOLUME),
            MED_CHANGE_VOL = median(CHANGED_VOLUME),
            MIN_CHANGE_VOL = min(CHANGED_VOLUME),
            MAX_CHANGE_VOL = max(CHANGED_VOLUME))
```

AVG_CHANGE_VOL <dbl>	MED_CHANGE_VOL <dbl>	MIN_CHANGE_VOL <dbl>	MAX_CHANGE_VOL <dbl>
32.51572	0	-132830	86977

1 row

```
# below in both year growth statistics

trans_history %>%
  filter(THRESHOLD_2023 == 'below' & THRESHOLD_2024 == 'below') %>%
  summarise(AVG_CHANGE_VOL = mean(CHANGED_VOLUME),
            MED_CHANGE_VOL = median(CHANGED_VOLUME),
            MIN_CHANGE_VOL = min(CHANGED_VOLUME),
            MAX_CHANGE_VOL = max(CHANGED_VOLUME))
```

AVG_CHANGE_VOL <dbl>	MED_CHANGE_VOL <dbl>	MIN_CHANGE_VOL <dbl>	MAX_CHANGE_VOL <dbl>
6.849459	1.5	-393	399.009

1 row

```
# above in both year growth statistics

trans_history %>%
  filter(THRESHOLD_2023 == 'above' & THRESHOLD_2024 == 'above') %>%
  summarise(AVG_CHANGE_VOL = mean(CHANGED_VOLUME),
            MED_CHANGE_VOL = median(CHANGED_VOLUME),
            MIN_CHANGE_VOL = min(CHANGED_VOLUME),
            MAX_CHANGE_VOL = max(CHANGED_VOLUME))
```

AVG_CHANGE_VOL <dbl>	MED_CHANGE_VOL <dbl>	MIN_CHANGE_VOL <dbl>	MAX_CHANGE_VOL <dbl>
5.785284	-17	-132830	82637.21

1 row

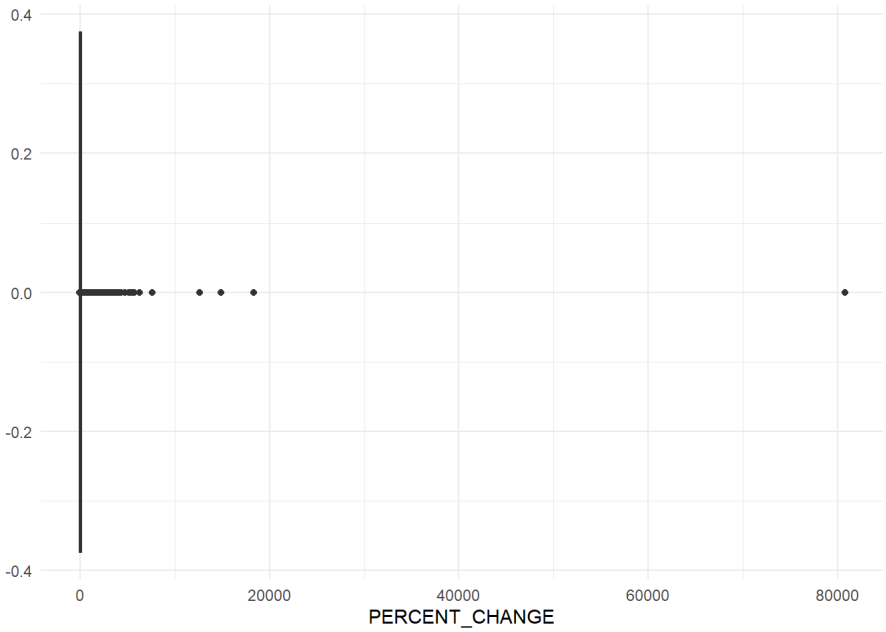
```
# potential growth customer statistics
trans_history %>%
  filter(THRESHOLD_2023 == 'below' & THRESHOLD_2024 == 'above') %>%
  summarise(AVG_CHANGE_VOL = mean(CHANGED_VOLUME),
            MED_CHANGE_VOL = median(CHANGED_VOLUME),
            MIN_CHANGE_VOL = min(CHANGED_VOLUME),
            MAX_CHANGE_VOL = max(CHANGED_VOLUME))
```

AVG_CHANGE_VOL <dbl>	MED_CHANGE_VOL <dbl>	MIN_CHANGE_VOL <dbl>	MAX_CHANGE_VOL <dbl>
1035.36	418	8.5	86977

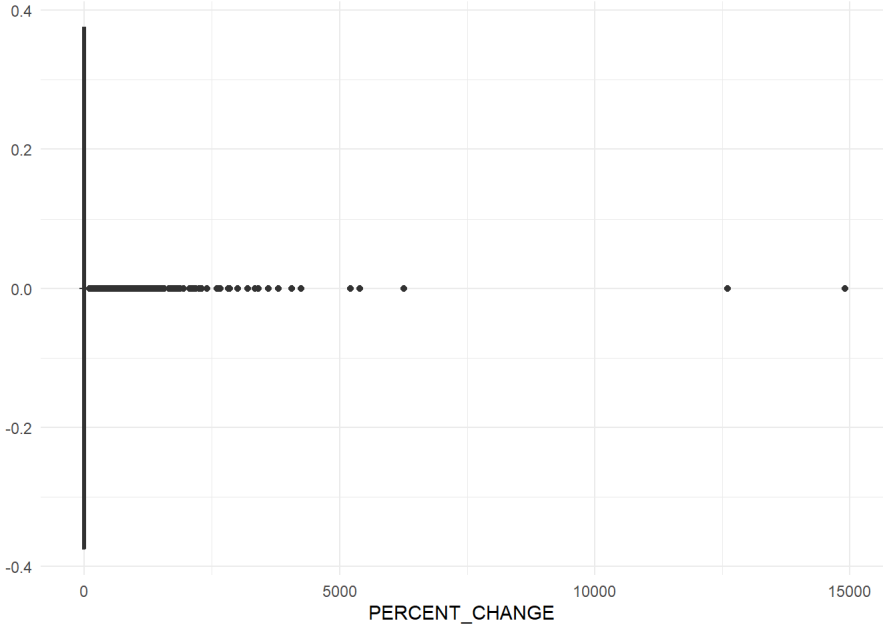
1 row

5-2. Changes in volume percent distribution

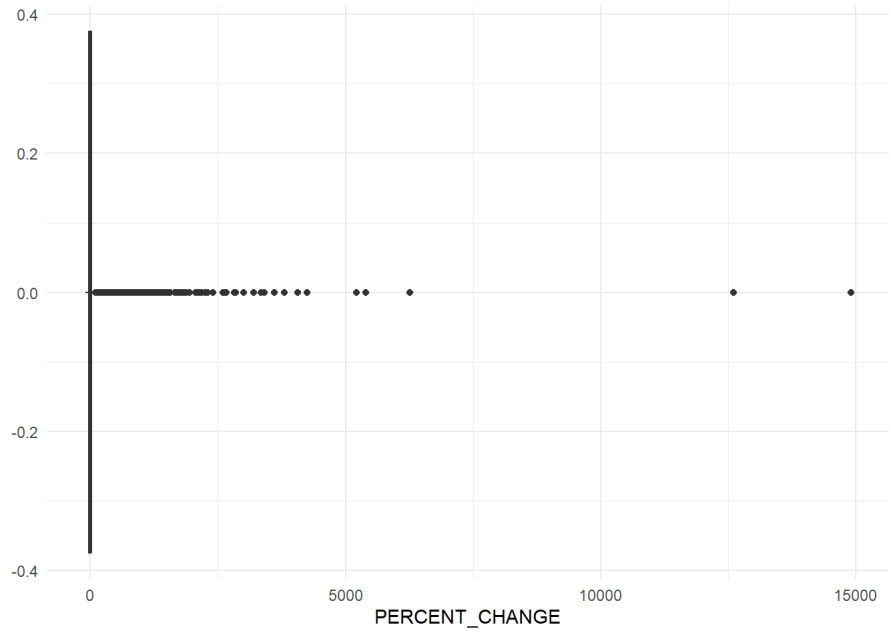
```
# total customer
trans_history %>%
  ggplot() +
  geom_boxplot(aes(x = PERCENT_CHANGE)) +
  theme_minimal()
```



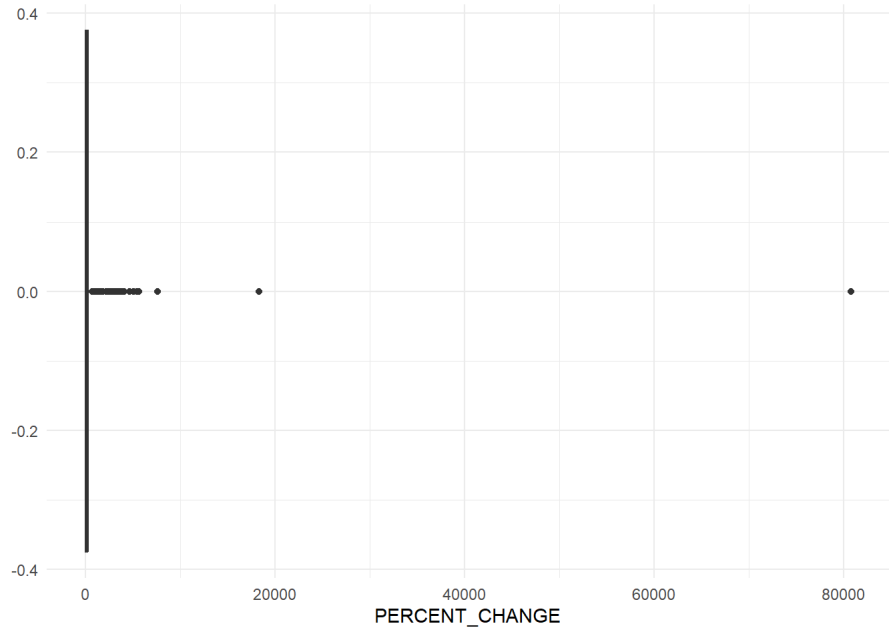

```
# both below customer
trans_history %>%
  filter(THRESHOLD_2023 == 'below' & THRESHOLD_2024 == 'below') %>%
  ggplot() +
  geom_boxplot(aes(x = PERCENT_CHANGE), na.rm = TRUE) +
  theme_minimal()
```




```
# both above customer
trans_history %>%
  filter(THRESHOLD_2023 == 'below' & THRESHOLD_2024 == 'below') %>%
  ggplot() +
  geom_boxplot(aes(x = PERCENT_CHANGE), na.rm = TRUE) +
  theme_minimal()
```




```
# potential growth customer
trans_history %>%
  filter(THRESHOLD_2023 == 'below' & THRESHOLD_2024 == 'above') %>%
  ggplot() +
  geom_boxplot(aes(x = PERCENT_CHANGE)) +
  theme_minimal()
```



6. Combining the Dataset (Data Modeling)

In order to take in-depth analysis per each of customer's attributes, we've combined the customer profile `profile_df` data with `trans_history` , joined by `CUSTOMER_NUMBER` variable.

```
trans_profile_df <- left_join(trans_history, profile_df, by = 'CUSTOMER_NUMBER')
sample_n(trans_profile_df,10)
```

CUSTOMER_NUMBER <dbl>	FIRST_TRANSACTION_DATE <date>	LAST_TRANSACTION_DATE <date>	TRANS_DAYS <drtn>
501164717	2024-01-23	2024-09-10	232 days
501215682	2023-02-01	2024-11-21	660 days
600258465	2023-01-06	2024-12-27	722 days
600685563	2023-01-05	2024-12-19	715 days
501546256	2023-08-23	2024-07-26	339 days
600260819	2023-01-09	2024-12-17	709 days

CUSTOMER_NUMBER <dbl>	FIRST_TRANSACTION_DATE <date>	LAST_TRANSACTION_DATE <date>	TRANS_DAYS <drtn>
600567040	2023-02-02	2024-12-12	680 days
600068126	2023-01-05	2024-11-21	687 days
600082683	2023-01-26	2024-11-21	666 days
501266455	2023-01-06	2024-12-26	721 days

1-10 of 10 rows | 1-4 of 29 columns

Variable comparison analysis

7-1. Local Market Partner Comparison

```
volume_2023 <- sum(trans_profile_df$ANNUAL_VOLUME_2023, na.rm = TRUE)
volume_2024 <- sum(trans_profile_df$ANNUAL_VOLUME_2024, na.rm = TRUE)

trans_profile_df %>%
  group_by(LOCAL_MARKET_PARTNER) %>%
  summarise(TOTAL_VOL_2023 = sum(ANNUAL_VOLUME_2023),
            TOTAL_VOL_2024 = sum(ANNUAL_VOLUME_2024),
            PERCENT_2023 = (TOTAL_VOL_2023 / volume_2023) * 100,
            PERCENT_2024 = (TOTAL_VOL_2024 / volume_2024) * 100,
            AVG_VOL_2023 = mean(ANNUAL_VOLUME_2023),
            AVG_VOL_2024 = mean(ANNUAL_VOLUME_2024),
            MED_VOL_2023 = median(ANNUAL_VOLUME_2023),
            MED_VOL_2024 = median(ANNUAL_VOLUME_2024),
            COUNT_2023 = sum(TRANS_COUNT_2023),
            COUNT_2024 = sum(TRANS_COUNT_2024),
            ABOVE_THRES_2023 = sum(THRESHOLD_2023 == 'above'),
            ABOVE_THRES_2024 = sum(THRESHOLD_2024 == 'above')
  )
```

LOCAL_MARKET_PARTNER <lgl>	TOTAL_VOL_2023 <dbl>	TOTAL_VOL_2024 <dbl>	PERCENT_2023 <dbl>	PERCENT_2024 <dbl>
FALSE	5332519	5310790	28.5071	26.96945
TRUE	13373414	14381084	71.4929	73.03055

2 rows | 1-5 of 13 columns

7-2. C02 customer Comparison

```
trans_profile_df %>%
  group_by(CO2_CUSTOMER) %>%
  summarise(TOTAL_VOL_2023 = sum(ANNUAL_VOLUME_2023),
            TOTAL_VOL_2024 = sum(ANNUAL_VOLUME_2024),
            PERCENT_2023 = (TOTAL_VOL_2023 / volume_2023) * 100,
            PERCENT_2024 = (TOTAL_VOL_2024 / volume_2024) * 100,
            AVG_VOL_2023 = mean(ANNUAL_VOLUME_2023),
            AVG_VOL_2024 = mean(ANNUAL_VOLUME_2024),
            MED_VOL_2023 = median(ANNUAL_VOLUME_2023),
            MED_VOL_2024 = median(ANNUAL_VOLUME_2024),
            COUNT_2023 = sum(TRANS_COUNT_2023),
            COUNT_2024 = sum(TRANS_COUNT_2024),
            ABOVE_THRES_2023 = sum(THRESHOLD_2023 == 'above'),
            ABOVE_THRES_2024 = sum(THRESHOLD_2024 == 'above')
  )
```

CO2_CUSTOMER <lgl>	TOTAL_VOL_2023 <dbl>	TOTAL_VOL_2024 <dbl>	PERCENT_2023 <dbl>	PERCENT_2024 <dbl>
FALSE	12304118	12919326	65.77655	65.6074
TRUE	6401815	6772548	34.22345	34.3926

2 rows | 1-5 of 13 columns

7-3. Frequent order type Comparison

```
trans_profile_df %>%
  group_by(FREQUENT_ORDER_TYPE) %>%
  summarise(TOTAL_VOL_2023 = sum(ANNUAL_VOLUME_2023),
            TOTAL_VOL_2024 = sum(ANNUAL_VOLUME_2024),
            PERCENT_2023 = (TOTAL_VOL_2023 / volume_2023) * 100,
            PERCENT_2024 = (TOTAL_VOL_2024 / volume_2024) * 100,
            AVG_VOL_2023 = mean(ANNUAL_VOLUME_2023),
            AVG_VOL_2024 = mean(ANNUAL_VOLUME_2024),
            MED_VOL_2023 = median(ANNUAL_VOLUME_2023),
            MED_VOL_2024 = median(ANNUAL_VOLUME_2024),
            COUNT_2023 = sum(TRANS_COUNT_2023),
            COUNT_2024 = sum(TRANS_COUNT_2024),
            ABOVE_THRES_2023 = sum(THRESHOLD_2023 == 'above'),
            ABOVE_THRES_2024 = sum(THRESHOLD_2024 == 'above')
  )
```

FREQUENT_ORDER_TYPE <chr>	TOTAL_VOL_2023 <dbl>	TOTAL_VOL_2024 <dbl>	PERCENT_2023 <dbl>	PERCENT_2024 <dbl>
CALL CENTER	179514.0	186631.8	0.9596635	0.9477604
EDI	149081.2	305437.8	0.7969731	1.5510854
MYCOKE LEGACY	246564.9	244420.9	1.3181106	1.2412271
MYCOKE360	381316.7	581339.1	2.0384802	2.9521774
OTHER	3753564.5	3612092.6	20.0661713	18.3430614
SALES REP	13995891.3	14761952.2	74.8206014	74.9646883

6 rows | 1-5 of 13 columns

7-4. Cold Drink Channel Comparison

```
trans_profile_df %>%
  group_by(COLD_DRINK_CHANNEL) %>%
  summarise(TOTAL_VOL_2023 = sum(ANNUAL_VOLUME_2023),
            TOTAL_VOL_2024 = sum(ANNUAL_VOLUME_2024),
            PERCENT_2023 = (TOTAL_VOL_2023 / volume_2023) * 100,
            PERCENT_2024 = (TOTAL_VOL_2024 / volume_2024) * 100,
            AVG_VOL_2023 = mean(ANNUAL_VOLUME_2023),
            AVG_VOL_2024 = mean(ANNUAL_VOLUME_2024),
            MED_VOL_2023 = median(ANNUAL_VOLUME_2023),
            MED_VOL_2024 = median(ANNUAL_VOLUME_2024),
            COUNT_2023 = sum(TRANS_COUNT_2023),
            COUNT_2024 = sum(TRANS_COUNT_2024),
            ABOVE_THRES_2023 = sum(THRESHOLD_2023 == 'above'),
            ABOVE_THRES_2024 = sum(THRESHOLD_2024 == 'above')
  )
```

COLD_DRINK_CHANNEL <chr>	TOTAL_VOL_2023 <dbl>	TOTAL_VOL_2024 <dbl>	PERCENT_2023 <dbl>	PERCENT_2024 <dbl>
ACCOMMODATION	476384.4	483019.35	2.54670235	2.45288662
BULK TRADE	4877746.7	5109930.39	26.07593428	25.94943632
CONVENTIONAL	5569.5	6052.25	0.02977398	0.03073476
DINING	5178051.2	5262747.86	27.68133134	26.72547961
EVENT	2377010.9	2448306.34	12.70725685	12.43307921
GOODS	1705056.7	2194385.48	9.11505824	11.14360898
PUBLIC SECTOR	999364.4	1027559.74	5.34249950	5.21819164
WELLNESS	622871.2	609083.30	3.32980584	3.09306918
WORKPLACE	2463877.7	2550789.64	13.17163762	12.95351368

9 rows | 1-5 of 13 columns

8. Group Segment #1

```
# Group 1: Local Market Partners that buy fountains only
group1_df <-
trans_profile_df %>%
  filter(!CO2_CUSTOMER
         & LOCAL_MARKET_PARTNER
         & ANNUAL_VOLUME_CASES_2023 == 0
         & ANNUAL_VOLUME_CASES_2024 == 0)

group1_df %>%
  summarise(TOTAL_VOLUME_2023 = sum(ANNUAL_VOLUME_GALLON_2023),
            TOTAL_VOLUME_2024 = sum(ANNUAL_VOLUME_GALLON_2024),
            ABOVE_THRES_2023 = sum(THRESHOLD_2023 == 'above'),
            ABOVE_THRES_2024 = sum(THRESHOLD_2024 == 'above'))
```

TOTAL_VOLUME_2023	TOTAL_VOLUME_2024	ABOVE_THRES_2023	ABOVE_THRES_2024
<dbl>	<dbl>	<int>	<int>
282140.3	292526.5	200	188

1 row

9. Threshold Comparison for 2023 and 2024

```
# Define threshold (400 gallons)
threshold <- 400

# Filter data for 2023 and 2024 only
transaction_filtered <- trans_df %>%
  filter(YEAR %in% c(2023, 2024))

# Summarize transactions per customer per year
customer_summary <- transaction_filtered %>%
  group_by(CUSTOMER_NUMBER, YEAR) %>%
  summarise(
    Total_Ordered_Cases = sum(ORDERED_CASES, na.rm = TRUE),
    Total_Ordered_Gallons = sum(ORDERED_GALLONS, na.rm = TRUE),
    Order_Frequency = n(),
    .groups = "drop"
  ) %>%
  # Add Total Volume Calculation
  mutate(
    Total_Volume = Total_Ordered_Cases + Total_Ordered_Gallons,
    Customer_Category = ifelse(Total_Ordered_Gallons >= threshold, "Above Threshold", "Below Threshold")
  ) %>%

  # Volume Segmentation
  mutate(
    Volume_Segment = case_when(
      Total_Volume >= 1000 ~ "High Volume",
      Total_Volume >= 500 ~ "Medium Volume",
      TRUE ~ "Low Volume"
    )
  ) %>%
  #  missing values
  mutate(
    Customer_Category = replace_na(Customer_Category, "Unknown"),
    Volume_Segment = replace_na(Volume_Segment, "Unknown")
  )

# Customers who changed from 2023-2024
threshold_change_customers <- customer_summary %>%
  select(CUSTOMER_NUMBER, YEAR, Customer_Category) %>%
  pivot_wider(names_from = YEAR, values_from = Customer_Category, values_fill = list(Customer_Category = "No Purchase")) %>%
  rename(Threshold_2023 = `2023`, Threshold_2024 = `2024`) %>%
  filter(Threshold_2023 != Threshold_2024)

threshold_transition_summary <- threshold_change_customers %>%
  group_by(Threshold_2023, Threshold_2024) %>%
  summarise(Customers_Transitioned = n(), .groups = "drop")

# Calculate net change in threshold categories
net_change_summary <- threshold_transition_summary %>%
  mutate(Change = case_when(
    Threshold_2023 == "Below Threshold" & Threshold_2024 == "Above Threshold" ~ Customers_Transitioned,
    Threshold_2023 == "Above Threshold" & Threshold_2024 == "Below Threshold" ~ -Customers_Transitioned,
    TRUE ~ 0
  )) %>%
  summarise(Net_Change = sum(Change))

# Track Customers Who Changed Volume Segments (Low/Medium/High)
volume_change_customers <- customer_summary %>%
  select(CUSTOMER_NUMBER, YEAR, Volume_Segment, Total_Volume, Order_Frequency) %>%
  pivot_wider(names_from = YEAR, values_from = c(Volume_Segment, Total_Volume, Order_Frequency),
              values_fill = list(Volume_Segment = "No Purchase", Total_Volume = 0, Order_Frequency = 0)) %>%
  rename(Volume_2023 = Volume_Segment_2023, Volume_2024 = Volume_Segment_2024,
         Volume_Ordered_2023 = Total_Volume_2023, Volume_Ordered_2024 = Total_Volume_2024,
         Order_Frequency_2023 = Order_Frequency_2023, Order_Frequency_2024 = Order_Frequency_2024)

# Identify customers with consistent or sporadic increases
volume_growth_analysis <- volume_change_customers %>%
  filter(Volume_2023 != "No Purchase" & Volume_2024 != "No Purchase" & Volume_2023 != Volume_2024) %>%
  mutate(Volume_Growth_Trend = case_when(
    Volume_Ordered_2024 > Volume_Ordered_2023 ~ "Consistent Growth",
    Volume_Ordered_2024 < Volume_Ordered_2023 ~ "Fluctuating",
    TRUE ~ "Stable"
  ))

# Identify patterns in customer order frequency changes
order_frequency_analysis <- volume_change_customers %>%
```

```
  mutate(Frequency_Change = Order_Frequency_2024 - Order_Frequency_2023,
         Frequency_Pattern = case_when(
           Frequency_Change > 0 ~ "Increasing Frequency",
           Frequency_Change < 0 ~ "Decreasing Frequency",
           TRUE ~ "Stable Frequency"
         ))

# Calculate average percentage drop for customers who reduced order volume
order_volume_drop_analysis <- volume_change_customers %>%
  filter(Volume_Ordered_2023 > 0 & Volume_Ordered_2024 < Volume_Ordered_2023) %>%
  mutate(Percentage_Drop = case_when(
    Volume_Ordered_2023 > 0 ~ ((Volume_Ordered_2023 - Volume_Ordered_2024) / Volume_Ordered_2023) * 100,
    TRUE ~ NA_real_  # Avoid division by zero
  ))

# Print summaries
print(order_frequency_analysis)
```

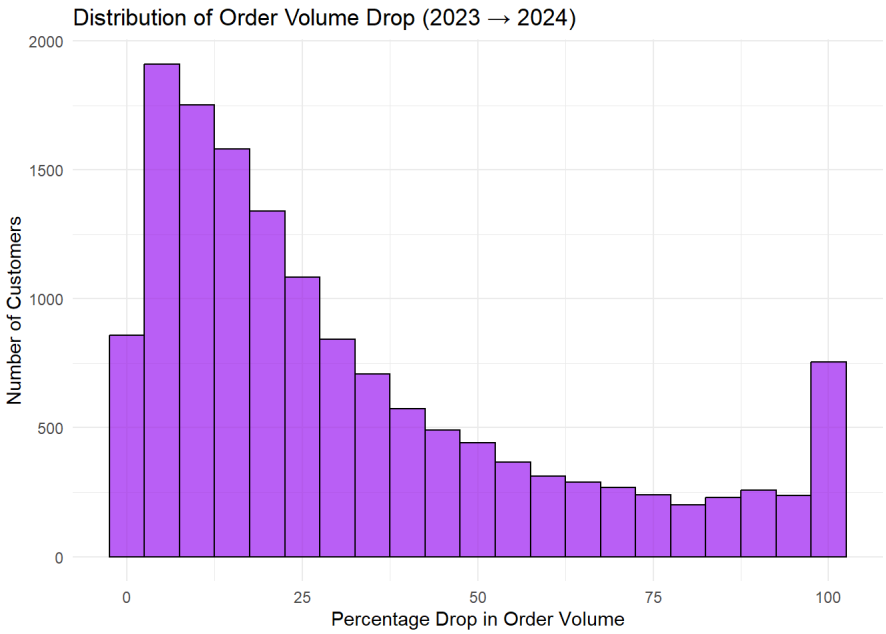
```
## # A tibble: 30,322 × 9
##    CUSTOMER_NUMBER Volume_2023   Volume_2024   Volume_Ordered_2023
##              <dbl> <chr>         <chr>                       <dbl>
##  1       500245678 Low Volume    Low Volume                  370
##  2       500245685 Medium Volume Low Volume                  602.
##  3       500245686 Low Volume    Low Volume                   17.5
##  4       500245687 Low Volume    Low Volume                  125
##  5       500245689 Medium Volume Medium Volume               546.
##  6       500245690 Low Volume    Low Volume                  325
##  7       500245695 High Volume   Medium Volume              1038.
##  8       500245698 Low Volume    High Volume                 282
##  9       500245701 Low Volume    Low Volume                  388
## 10       500245704 High Volume   High Volume                1585
## # ℹ 30,312 more rows
## # ℹ 5 more variables: Volume_Ordered_2024 <dbl>, Order_Frequency_2023 <int>,
## #   Order_Frequency_2024 <int>, Frequency_Change <int>, Frequency_Pattern <chr>
```

```
print(order_volume_drop_analysis)
```

```
## # A tibble: 14,742 × 8
##    CUSTOMER_NUMBER Volume_2023   Volume_2024   Volume_Ordered_2023
##              <dbl> <chr>         <chr>                       <dbl>
##  1       500245685 Medium Volume Low Volume                  602.
##  2       500245690 Low Volume    Low Volume                  325
##  3       500245695 High Volume   Medium Volume              1038.
##  4       500245701 Low Volume    Low Volume                  388
##  5       500245704 High Volume   High Volume                1585
##  6       500245725 High Volume   Medium Volume              1015
##  7       500245726 Low Volume    Low Volume                    60
##  8       500245732 Low Volume    Low Volume                    25
##  9       500245740 Low Volume    Low Volume                   129.
## 10       500245765 Low Volume    Low Volume                   139
## # ℹ 14,732 more rows
## # ℹ 4 more variables: Volume_Ordered_2024 <dbl>, Order_Frequency_2023 <int>,
## #   Order_Frequency_2024 <int>, Percentage_Drop <dbl>
```

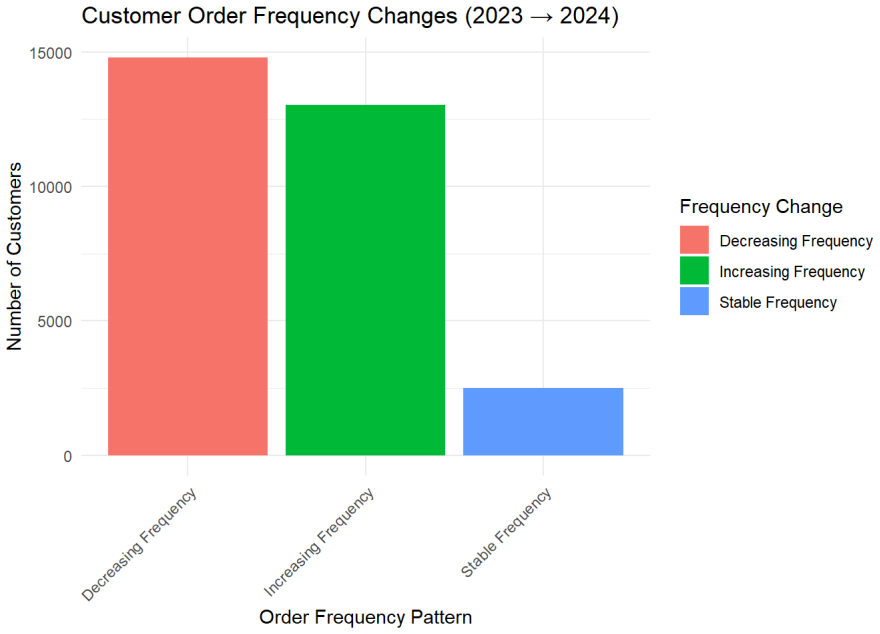
10. Visualization for Order Volume

```
# Visualization for Order Volume Drop Distribution
ggplot(order_volume_drop_analysis, aes(x = Percentage_Drop)) +
  geom_histogram(binwidth = 5, fill = "purple", alpha = 0.7, color = "black") +
  theme_minimal() +
  labs(title = "Distribution of Order Volume Drop (2023 → 2024)",
       x = "Percentage Drop in Order Volume",
       y = "Number of Customers")
```



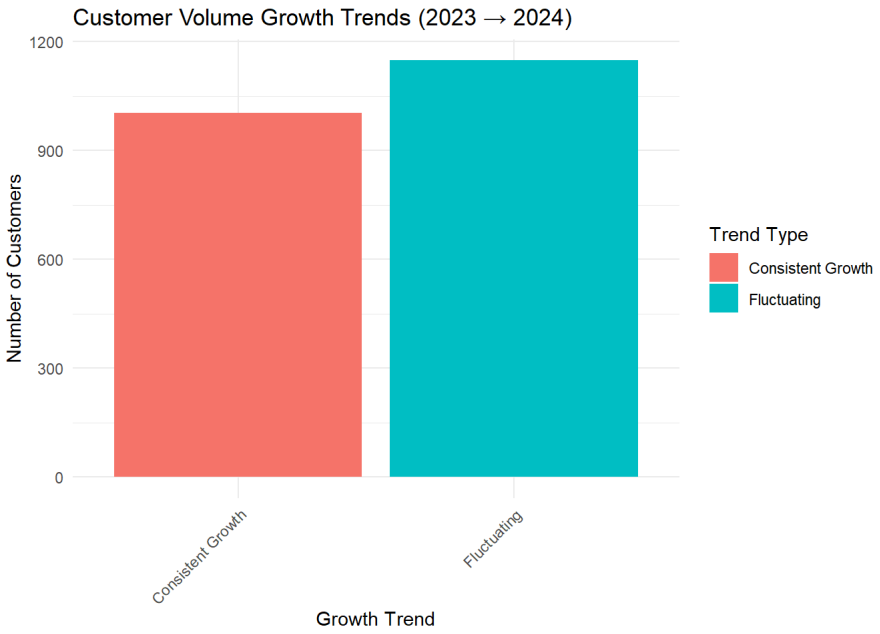
11. Visualization for Customer Order Frequency Changes

```
ggplot(order_frequency_analysis, aes(x = Frequency_Pattern, fill = Frequency_Pattern)) +
  geom_bar() +
  theme_minimal() +
  labs(title = "Customer Order Frequency Changes (2023 → 2024)",
       x = "Order Frequency Pattern",
       y = "Number of Customers",
       fill = "Frequency Change") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



12. Visualization for Volume Growth Trends

```
ggplot(volume_growth_analysis, aes(x = Volume_Growth_Trend, fill = Volume_Growth_Trend)) +
  geom_bar() +
  theme_minimal() +
  labs(title = "Customer Volume Growth Trends (2023 → 2024)",
       x = "Growth Trend",
       y = "Number of Customers",
       fill = "Trend Type") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



EDA Insights (Summary)

Out of EDA, we could find out below insights

- There are 30,322 unique customers from 2023/01/01 to 2024/12/31. (2 years) out of transaction history data.
- Even if approximately 90% of customers belongs to the Local Market Customers, their total volume of transaction takes up 72% of entire transaction volumes.
    - There is an 2% point increase of proportion in 2024 for Local Market Customers compared to 2023, which represents local retails growth potential.
    - Local market customers are likely to order +4 more frequencies with +4 less volume compared to non-local market customer.
- Even though there is not much change of ordering pattern between CO2 customer and Non CO2 customer in 2023 and 2024, median volume per order has increased by over 10% in 2024 compared to 2023 for C02 customer.
- SALES REP (sales representatives) remains in 75% of order type for 2 years transactions, followed by OTHERS, and MYCOKE360 (Digital Ordering Platform), which indicates that personal interaction is still significant to maintain the sales.
    - However, EDI ordering volume increase over 2 times more, and MYCOKE360 volumes increase by 1.5 times from 2023 to 2024.
- In terms of order volume percentage per year, Goods channel increase by 2% points from 2023 to 2024.
- BULK TRADES and DINING takes over 50% of entire transaction volume in both 2023 and 2024.
- 14,742 customers experienced a decline in order volume, including some high-volume customers moving to medium or low volume. Growth segment: Certain customers moved from low to high volume, indicating rising demand and potential need for priority servicing.
- Some customers crossed above or below the 400-gallon threshold, affecting route efficiency and delivery planning. Net Impact: Helps assess whether SCCU should expand direct delivery routes or refine ARTM logistics.
- Increased order frequency suggests growth potential, while decreased frequency may signal churn risk.

Contribution

- Richard Lim: Structuring and organizing the EDA notebook
- Varun Selvam: Yaml file formatting and data validation
- Nikita Muddapati: Delivery cost calculation and additional EDA questions
- Meenakshi Hariharan: Implementing threshold, volume analysis and key patterns in customer order, volume reduction from 2023 to 2024