EDA

Alexia Wells

2025-02-19

Contents

Business Statement **1**

Exploration Tasks **2**
What Does Each Data File Look Like? . . . 2
customer_address_and_zip_mapping.csv . . . 2
customer_profile.csv . . . 3
delivery_cost_data.xlsx . . . 3
transactional_data.csv . . . 3
How Can the Datasets be Combined? . . . 4
address_zip + profile . . . 4
Updated customer_demo + Transactional . . . 4
Cleaning the Data . . . 4
How Does the Initial Data Look? . . . 4
Are there columns with near zero variance? . . . 5
What is the scope of the missing variable? . . . 6
Are there string errors in the data? . . . 7
Visualizations . . . 7
Bar Plots . . . 8
Principal Component Analysis . . . 11
Correlation Analysis . . . 18
Feature Engineering . . . 19
Should Volume Range be Set Up as a Binning Variable? . . . 19
Add Total Delivery Costs Calculation . . . 20
How exactly should this be calculated? If we have negatives, do we just not calculate the cost? 20
Should this only be calculated by ordered or delivered? Cause I thought delivered but now I am unsure? . . . 20
Finalize Data Cleaning . . . 20
Are there outliers? . . . 20
What is the Distribution of the Target Variable? . . . 20
How to Approach NAs? . . . 20

Results **20**
Future Plans . . . 21

Business Statement

Swire Coca-Cola is a leading tech company that puts clients at the center of their work. As a bottler and distributor throughout 13 Western states, operational efficiency is a necessity to remain successful and create long-term relationships with clientele. Swire Coca-Cola must avoid prematurely moving high-growth potential customers to white truck delivery (Alternate Route to Market) as this could severely impact revenue growth.

zip	full address
71018	71018,Cotton Valley,Louisiana,LA,Webster,119,32.819,-93.4259
71021	71021,Cullen,Louisiana,LA,Webster,119,32.9721,-93.4492
71023	71023,Doyline,Louisiana,LA,Webster,119,32.49,-93.3996
71024	71024,Dubberly,Louisiana,LA,Webster,119,32.5192,-93.2142
71039	71039,Heflin,Louisiana,LA,Webster,119,32.447,-93.2852
71055	71055,Minden,Louisiana,LA,Webster,119,32.6323,-93.2886

The purpose of this project is to reasonably maximize sales by predicting which clients have the potential to reach the high-performing SCCU optimal threshold of 400 gallons annually.

This will be a predictive analytics project that includes a supervised regression algorithm that determines the number of expected gallons annually. The model will take a balanced approach to inform client growth by taking into account historical sales data and other customer characteristics. A few of these customer demographics include the client's zip code, full address information, if they are a local market partner, etc. The target variable is the annual volume range.

The purpose of this EDA notebook is to take a look at the data and get an initial understanding. That means variable distributions and relationships will be explored. One of my main goals is to identify issues that may come up in the modeling process and consequently making any necessary fixes. The questions that will be explored in this notebook can be found in the table of contents.

Exploration Tasks

What Does Each Data File Look Like?

In total, the Swire Coca-Cola Team provided four data-files; Three CSV's and one was Excel file. Each file will be looked at individually.

```
# Load the libraries
library(vroom)
library(tidyverse)
library(tidymodels)
library(DataExplorer)
library(readxl)
library(caret)
library(kableExtra)
```

```
# Read in the data
address_zip <- vroom("customer_address_and_zip_mapping.csv")
profile <- vroom("customer_profile.csv")
delivery_cost <- read_xlsx("delivery_cost_data.xlsx")
transactional <- vroom("transactional_data.csv")
```

customer_address_and_zip_mapping.csv

This file has just two variables. On first glance, it does not seem entirely helpful considering the zip code and full address are completely randomized. It is important to note, the zip codes are real but they are blinded for data privacy purposes. With that said, it is still essential to investigate if there are any patterns.

```
address_zip |>
  head() |>
  kbl() |>
  kable_classic(full_width = F, html_font = "Cambria")
```

CUSTOMER_NUMBER	PRIMARY_GROUP_NUMBER	FREQUENT_ORDER_TYPE	FIRST_DELIVERY_D
501556470	376	MYCOKE LEGACY	1/2/2024
501363456	NA	SALES REP	4/14/2022
600075150	2158	SALES REP	3/4/2016
500823056	2183	OTHER	2/6/2019
600082383	1892	SALES REP	3/4/2016
600079420	NA	SALES REP	3/10/2016

Cold Drink Channel	Vol Range	Applicable To	Median Delivery Cost	Cost Type
WORKPLACE	0 - 149	Bottles and Cans	8.064950	Per Case
WORKPLACE	150 - 299	Bottles and Cans	4.165646	Per Case
WORKPLACE	300 - 449	Bottles and Cans	2.991558	Per Case
WORKPLACE	450 - 599	Bottles and Cans	2.524222	Per Case
WORKPLACE	600 - 749	Bottles and Cans	2.056886	Per Case
WORKPLACE	750 - 899	Bottles and Cans	1.999564	Per Case

customer_profile.csv

This file focuses on detailed information about customers, onboarding, and their purchasing behavior. There are eleven columns and 30,478 rows.

```
profile |>
  head()|>
  kbl() |>
  kable_classic(full_width = F, html_font = "Cambria")
```

delivery_cost_data.xlsx

This dataset is slightly simpler considering there are only 160 rows and 5 columns relating to delivery costs. It will be helpful for calculating the total delivery costs.

```
delivery_cost |>
  head()|>
  kbl() |>
  kable_classic(full_width = F, html_font = "Cambria")
```

transactional_data.csv

The dataset keeps track of transactional information, specifically order quantities and delivery metrics. There are 1,045,540 rows and 11 columns. For context, fountains drinks are measured in gallons and bottles/cans are measured in cases.

```
transactional |>
  head()|>
  kbl() |>
  kable_classic(full_width = F, html_font = "Cambria")
```

TRANSACTION_DATE	WEEK	YEAR	CUSTOMER_NUMBER	ORDER_TYPE	ORDERED_CASES
1/5/2023	1	2023	501202893	MYCOKE LEGACY	1.0
1/6/2023	1	2023	500264574	MYCOKE LEGACY	12.5
1/9/2023	2	2023	501174701	MYCOKE LEGACY	2.0
1/11/2023	2	2023	600586532	SALES REP	18.0
1/17/2023	3	2023	501014325	SALES REP	29.0
1/23/2023	4	2023	600567384	CALL CENTER	1.5

How Can the Datasets be Combined?

address_zip + profile

My first initial thought was to combine these two datasets together by zip. I ended up doing a left join. This would a good move that prevented data loss.

```
# Checking for duplicated info in address_zip, there is 0
sum(duplicated(address_zip))
```

```
## [1] 0
```

```
# Our current dimensions
dim(address_zip)
```

```
## [1] 1801    2
```

```
dim(profile)
```

```
## [1] 30478    11
```

```
# Rename zip column of address_zip
address_zip <- address_zip |>
  rename('ZIP_CODE' = zip)

# Conduct the join
customer_demo <- left_join(profile, address_zip, by = "ZIP_CODE")
```

Updated customer_demo + Transactional

From there, I joined the dataset created above to the transactional data by the customer number.

```
# Join
joined_data <- left_join(transactional, customer_demo, by = "CUSTOMER_NUMBER")
```

Cleaning the Data

How Does the Initial Data Look?

Luckily, there is no duplicated data. I did need to change the types of certain columns.

```
# Check for duplicated data
anyDuplicated(joined_data)
```

```
## [1] 0
```

```
# Change categorical variables to factors and transaction_date to date variable
full_data <- joined_data |>
  mutate(across(c(CUSTOMER_NUMBER, ORDER_TYPE, PRIMARY_GROUP_NUMBER,
                  FREQUENT_ORDER_TYPE, COLD_DRINK_CHANNEL,
                  TRADE_CHANNEL, SUB_TRADE_CHANNEL, LOCAL_MARKET_PARTNER,
                  CO2_CUSTOMER, ZIP_CODE), as.factor)) |>
  mutate(across(c(TRANSACTION_DATE, FIRST_DELIVERY_DATE, ON_BOARDING_DATE),
                ~ as.Date(., format = "%m/%d/%Y")))

# Check that the changes look good
str(full_data)
```

```
## tibble [1,045,540 x 22] (S3: tbl_df/tbl/data.frame)
##  $ TRANSACTION_DATE    : Date[1:1045540], format: "2023-01-05" "2023-01-06" ...
```

```
##  $ WEEK                : num [1:1045540] 1 1 2 2 3 4 4 4 4 5 ...
##  $ YEAR                : num [1:1045540] 2023 2023 2023 2023 2023 ...
##  $ CUSTOMER_NUMBER     : Factor w/ 30322 levels "500245678","500245685",..: 8061 155 7686 29919 5640
##  $ ORDER_TYPE          : Factor w/ 7 levels "CALL CENTER",..: 3 3 3 7 7 1 3 1 1 3 ...
##  $ ORDERED_CASES       : num [1:1045540] 1 12.5 2 18 29 1.5 6 0 4 18 ...
##  $ LOADED_CASES        : num [1:1045540] 1 12.5 2 16 29 1.5 5 0 1 17 ...
##  $ DELIVERED_CASES     : num [1:1045540] 1 12.5 2 16 29 1.5 5 0 1 17 ...
##  $ ORDERED_GALLONS     : num [1:1045540] 90 0 0 2.5 0 0 0 25 0 0 ...
##  $ LOADED_GALLONS      : num [1:1045540] 90 0 0 2.5 0 0 0 25 0 0 ...
##  $ DELIVERED_GALLONS   : num [1:1045540] 90 0 0 2.5 0 0 0 25 0 0 ...
##  $ PRIMARY_GROUP_NUMBER: Factor w/ 1020 levels "4","17","19",..: NA 313 NA 951 355 71 NA NA NA NA ..
##  $ FREQUENT_ORDER_TYPE : Factor w/ 6 levels "CALL CENTER",..: 6 5 4 6 6 6 6 5 6 6 ...
##  $ FIRST_DELIVERY_DATE : Date[1:1045540], format: "2021-05-07" "2018-03-23" ...
##  $ ON_BOARDING_DATE    : Date[1:1045540], format: "2021-04-02" "2015-12-08" ...
##  $ COLD_DRINK_CHANNEL  : Factor w/ 9 levels "ACCOMMODATION",..: 4 8 4 2 6 6 4 4 4 4 ...
##  $ TRADE_CHANNEL       : Factor w/ 26 levels "ACADEMIC INSTITUTION",..: 5 12 8 9 10 26 8 17 8 17 ...
##  $ SUB_TRADE_CHANNEL   : Factor w/ 48 levels "ASIAN FAST FOOD",..: 13 31 27 9 28 41 42 26 18 26 ...
##  $ LOCAL_MARKET_PARTNER: Factor w/ 2 levels "FALSE","TRUE": 2 2 2 1 2 2 2 2 2 2 ...
##  $ CO2_CUSTOMER        : Factor w/ 2 levels "FALSE","TRUE": 2 1 2 1 1 1 1 1 1 1 ...
##  $ ZIP_CODE            : Factor w/ 1799 levels "1001","1002",..: 1548 1227 295 134 1656 1575 849 161
##  $ full address        : chr [1:1045540] "66955,Mahaska,Kansas,KS,Washington,201,39.9845,-97.3453" "4
```

Are there columns with near zero variance?

No, all the columns look great.

```
nearZeroVar(full_data, saveMetrics = TRUE)
```

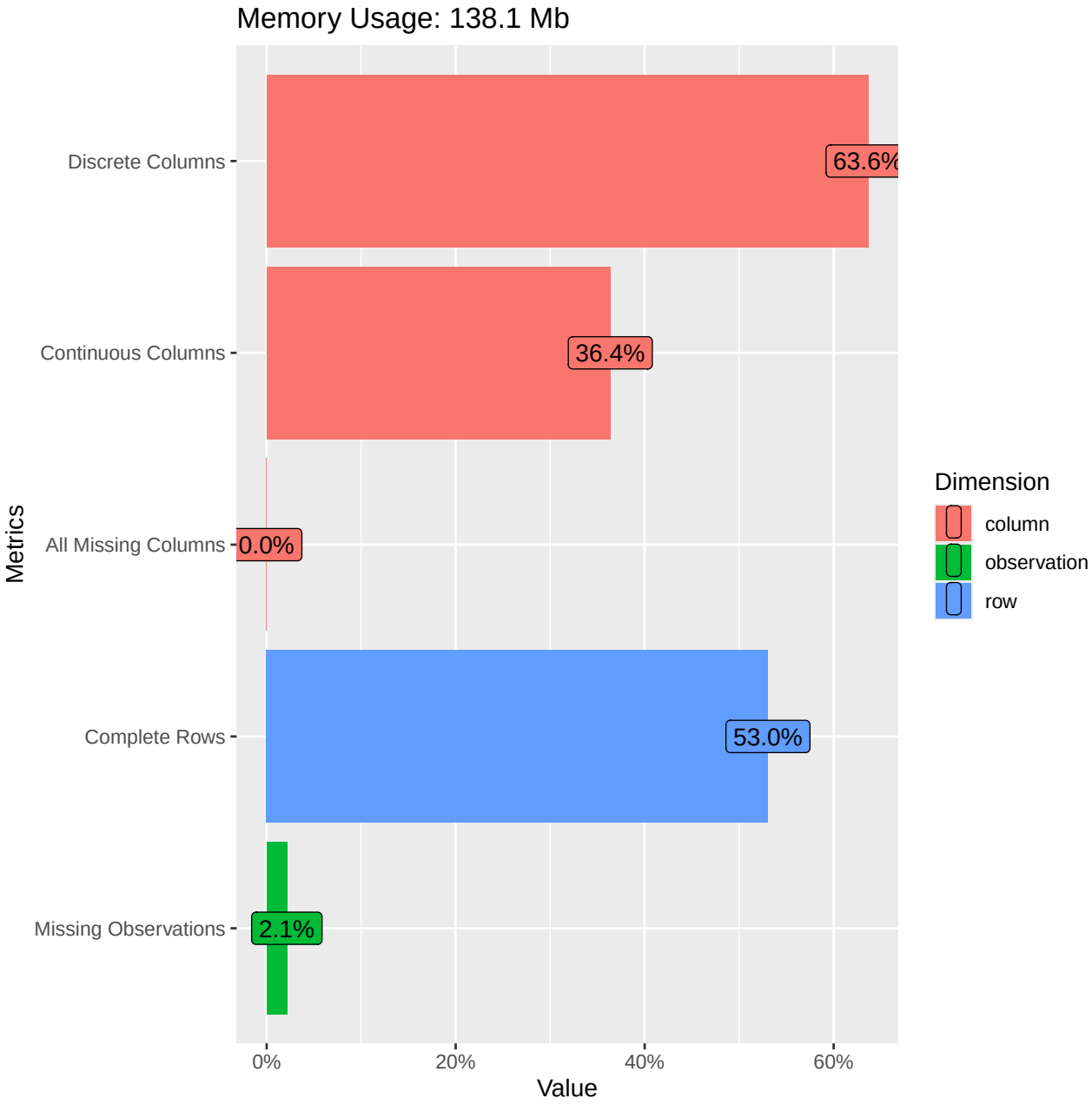
```
##                      freqRatio percentUnique zeroVar   nzv
## TRANSACTION_DATE      1.020093  0.0691508694   FALSE FALSE
## WEEK                  1.007206  0.0049735065   FALSE FALSE
## YEAR                  1.013961  0.0001912887   FALSE FALSE
## CUSTOMER_NUMBER       1.228125  2.9001281634   FALSE FALSE
## ORDER_TYPE            1.107733  0.0006695105   FALSE FALSE
## ORDERED_CASES         7.488479  0.4758306712   FALSE FALSE
## LOADED_CASES          7.559491  0.4382424393   FALSE FALSE
## DELIVERED_CASES       8.345553  0.6865351876   FALSE FALSE
## ORDERED_GALLONS       9.666112  1.0319069572   FALSE FALSE
## LOADED_GALLONS        9.440847  0.9500353884   FALSE FALSE
## DELIVERED_GALLONS     9.971936  0.9487920118   FALSE FALSE
## PRIMARY_GROUP_NUMBER  3.034575  0.0975572431   FALSE FALSE
## FREQUENT_ORDER_TYPE   4.006904  0.0005738661   FALSE FALSE
## FIRST_DELIVERY_DATE   1.047164  0.2296420988   FALSE FALSE
## ON_BOARDING_DATE      1.258895  0.6193928496   FALSE FALSE
## COLD_DRINK_CHANNEL    3.044745  0.0008607992   FALSE FALSE
## TRADE_CHANNEL         1.502409  0.0024867533   FALSE FALSE
## SUB_TRADE_CHANNEL     1.918214  0.0045909291   FALSE FALSE
## LOCAL_MARKET_PARTNER  4.656429  0.0001912887   FALSE FALSE
## CO2_CUSTOMER          1.566139  0.0001912887   FALSE FALSE
## ZIP_CODE              1.100090  0.1720641965   FALSE FALSE
## full address          1.100090  0.1720641965   FALSE FALSE
```

What is the scope of the missing variable?

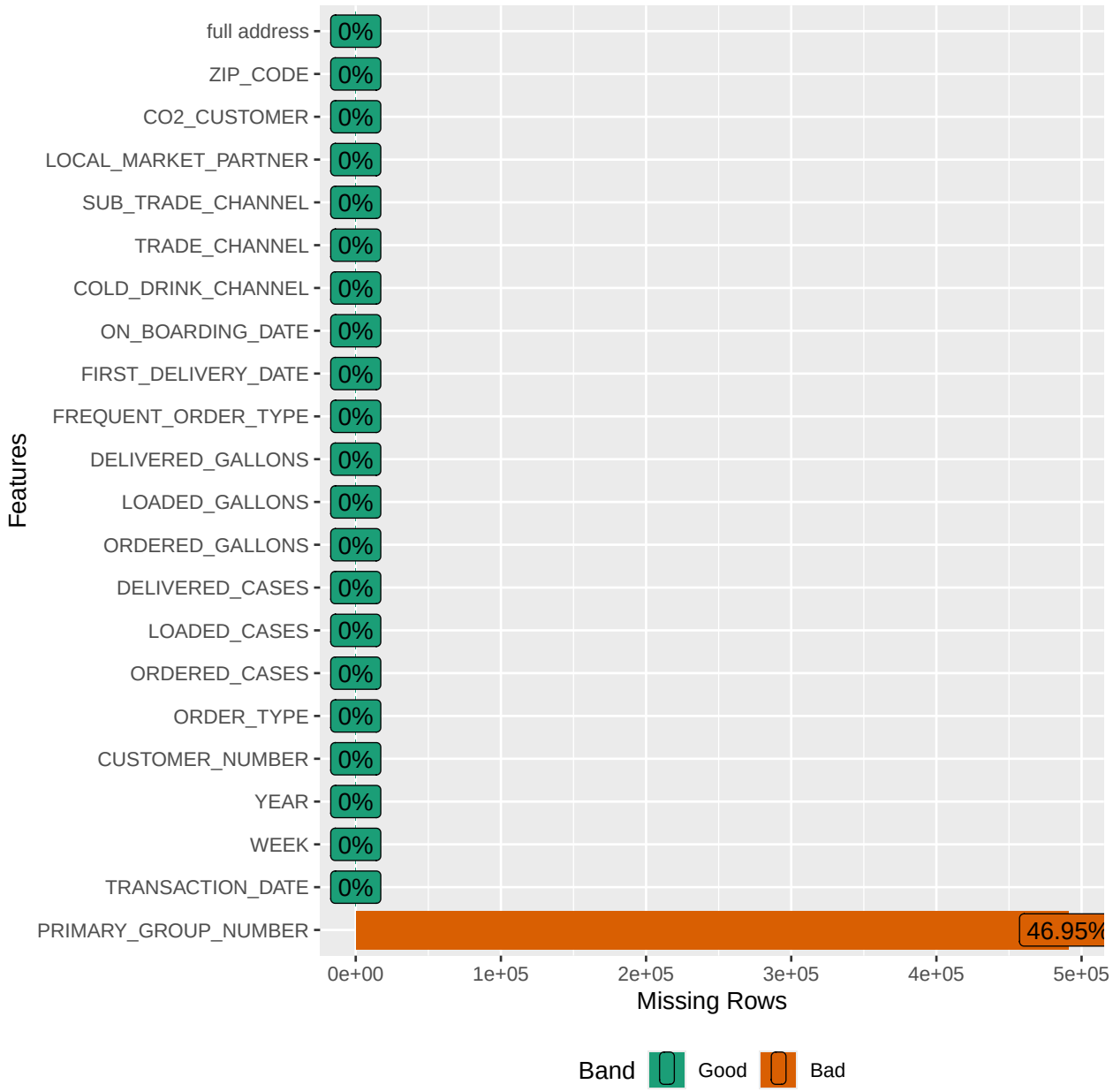
We are only missing 2.1% of observations. Because of this it would likely be best just to remove those instances. The leaders of Coca Cola shared that if there is missing data it likely doesn't have anything to do with informative missingness.

Looks like the missing data is all from one column, PRIMARY_GROUP_NUMBER.

```
full_data |>
  plot_intro()
```




```
full_data |>
  plot_missing()
```



Are there string errors in the data?

No, the string/factor data looks perfect.

```
unique(full_data$ORDER_TYPE)
unique(full_data$FREQUENT_ORDER_TYPE)
unique(full_data$COLD_DRINK_CHANNEL)
unique(full_data$TRADE_CHANNEL)
unique(full_data$SUB_TRADE_CHANNEL)
```

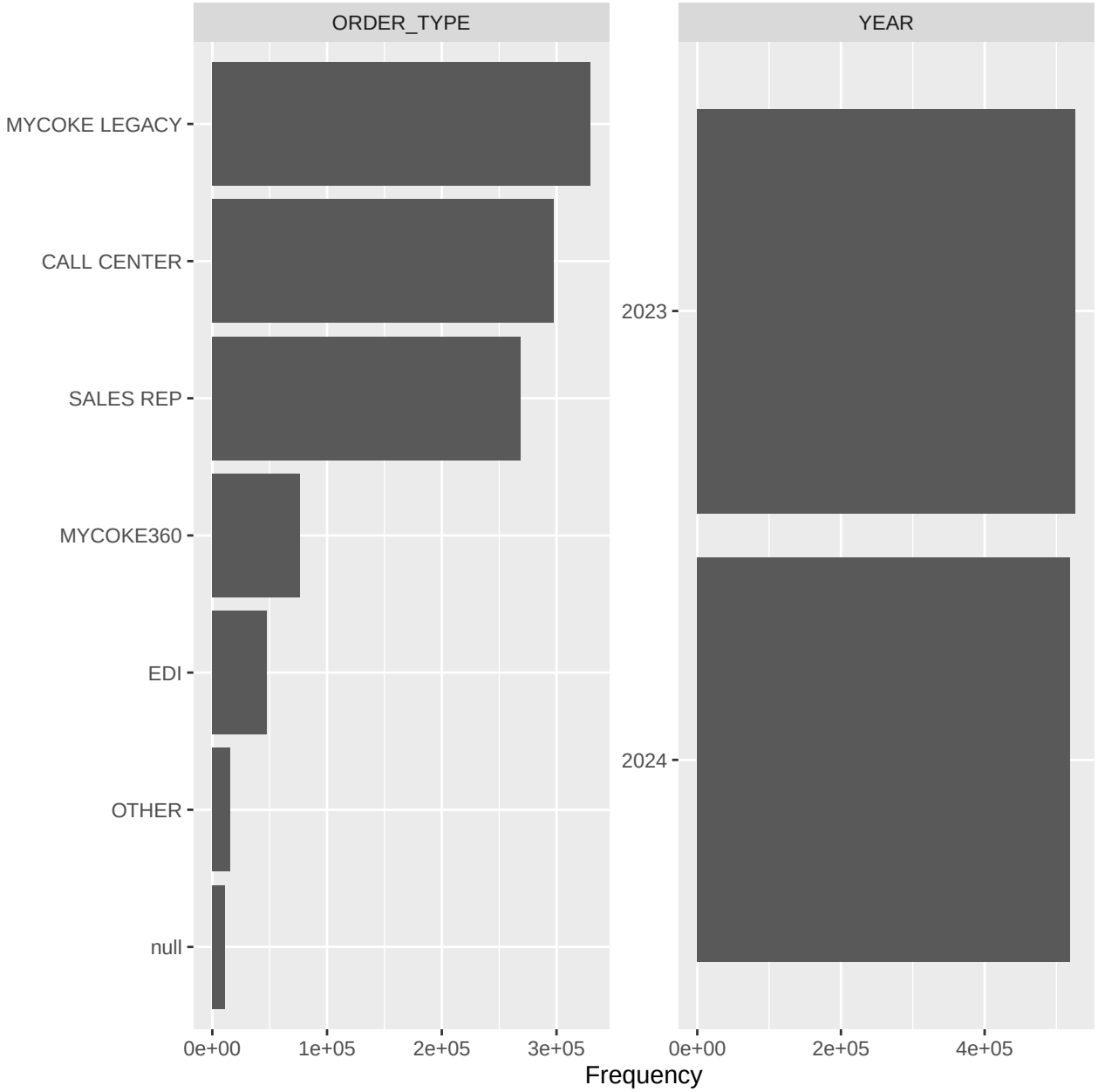
Visualizations

Data summaries based on your questions: plots and summary tables. If, as you are doing EDA, more questions occur to you, then go back and put them into your question list. Make sure that you include plot titles and axis labels.

Bar Plots

```
full_data[0:10] |>
  plot_bar()
```

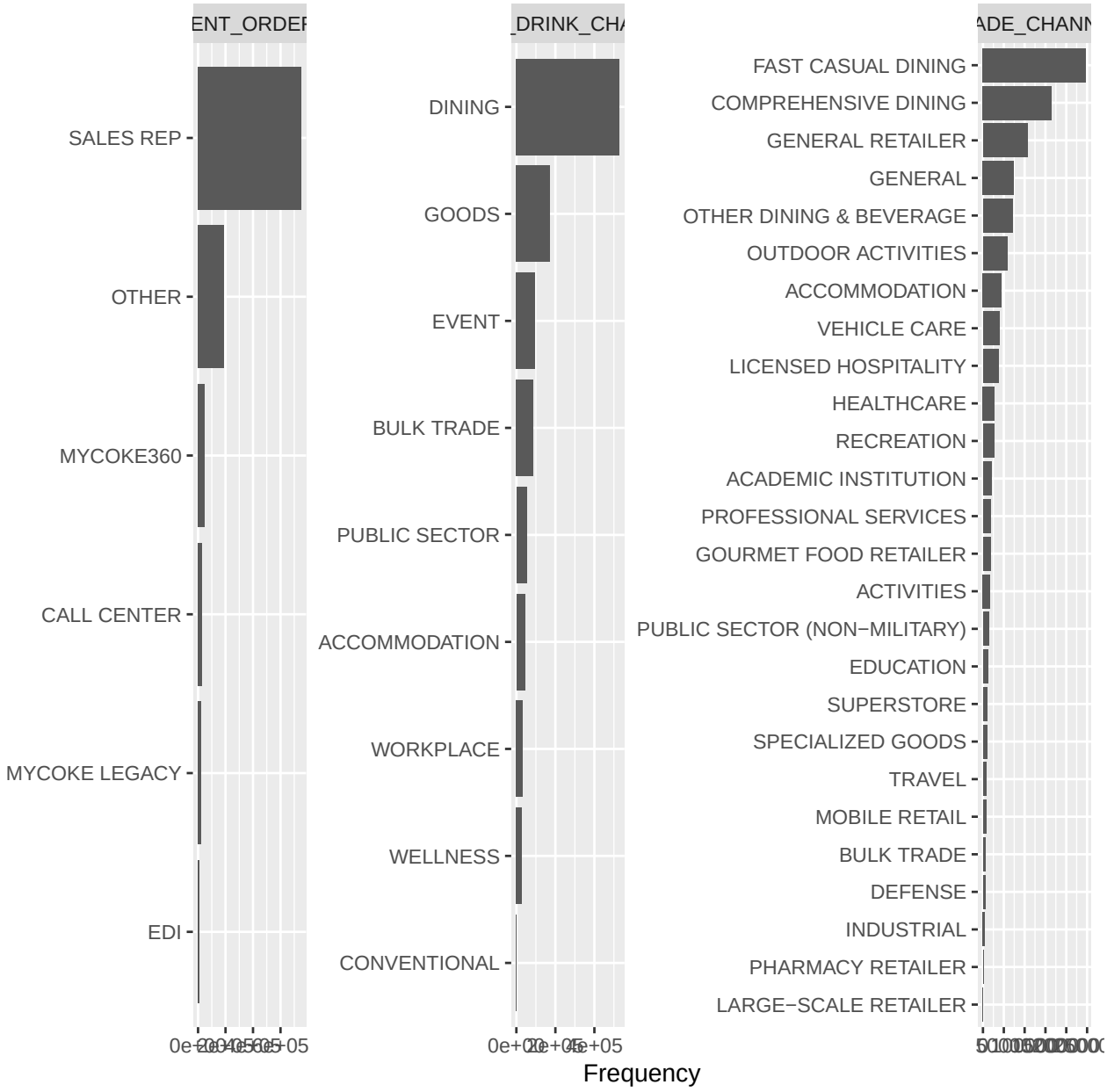
```
## 2 columns ignored with more than 50 categories.
## TRANSACTION_DATE: 723 categories
## CUSTOMER_NUMBER: 30322 categories
```



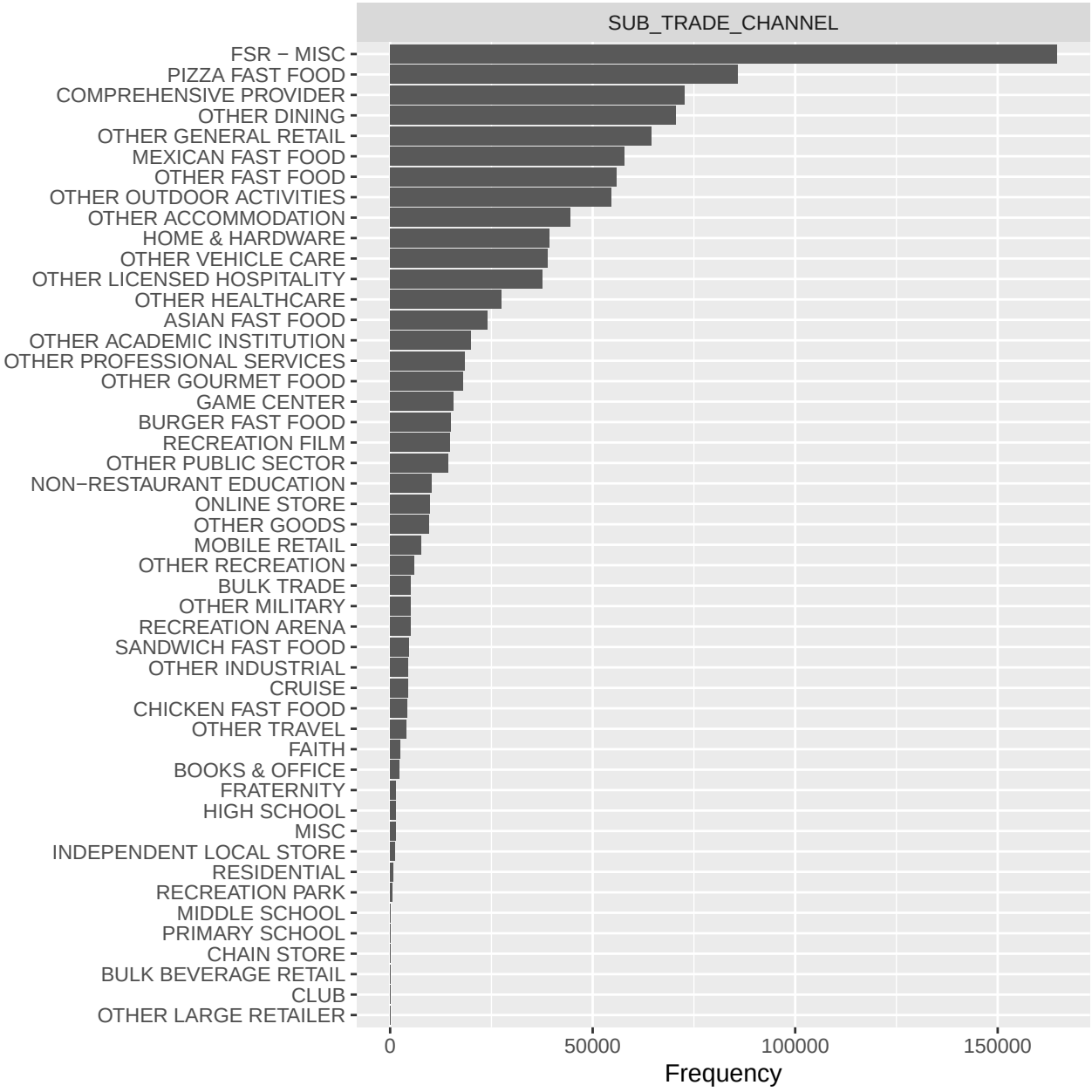

```
full_data[11:17] |>
  plot_bar()
```

```
## 3 columns ignored with more than 50 categories.
## PRIMARY_GROUP_NUMBER: 1021 categories
## FIRST_DELIVERY_DATE: 2401 categories
```

```
## ON_BOARDING_DATE: 6476 categories
```

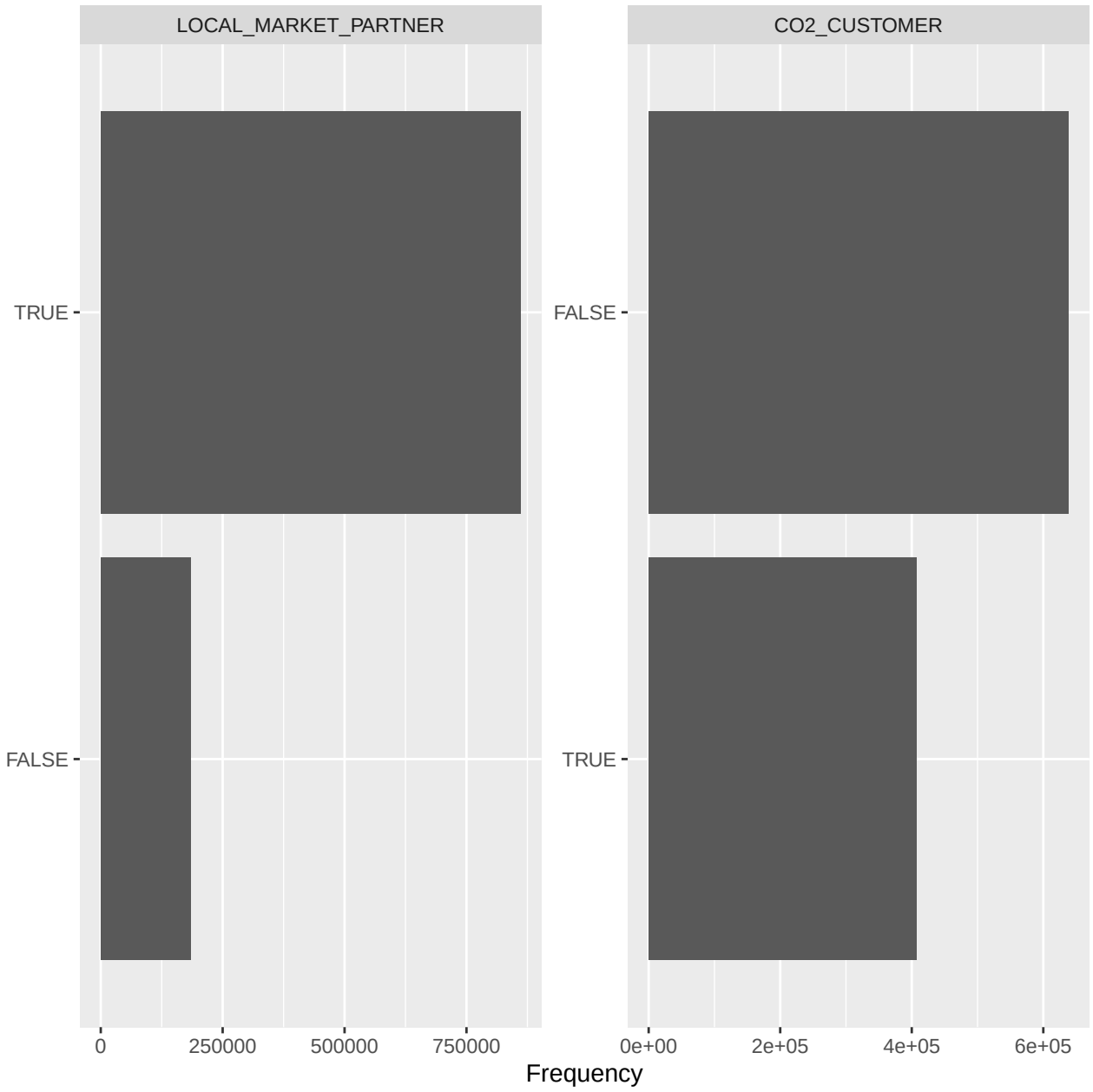



```
full_data[18] |>
  plot_bar()
```




```
full_data[19:22] |>
  plot_bar()
```

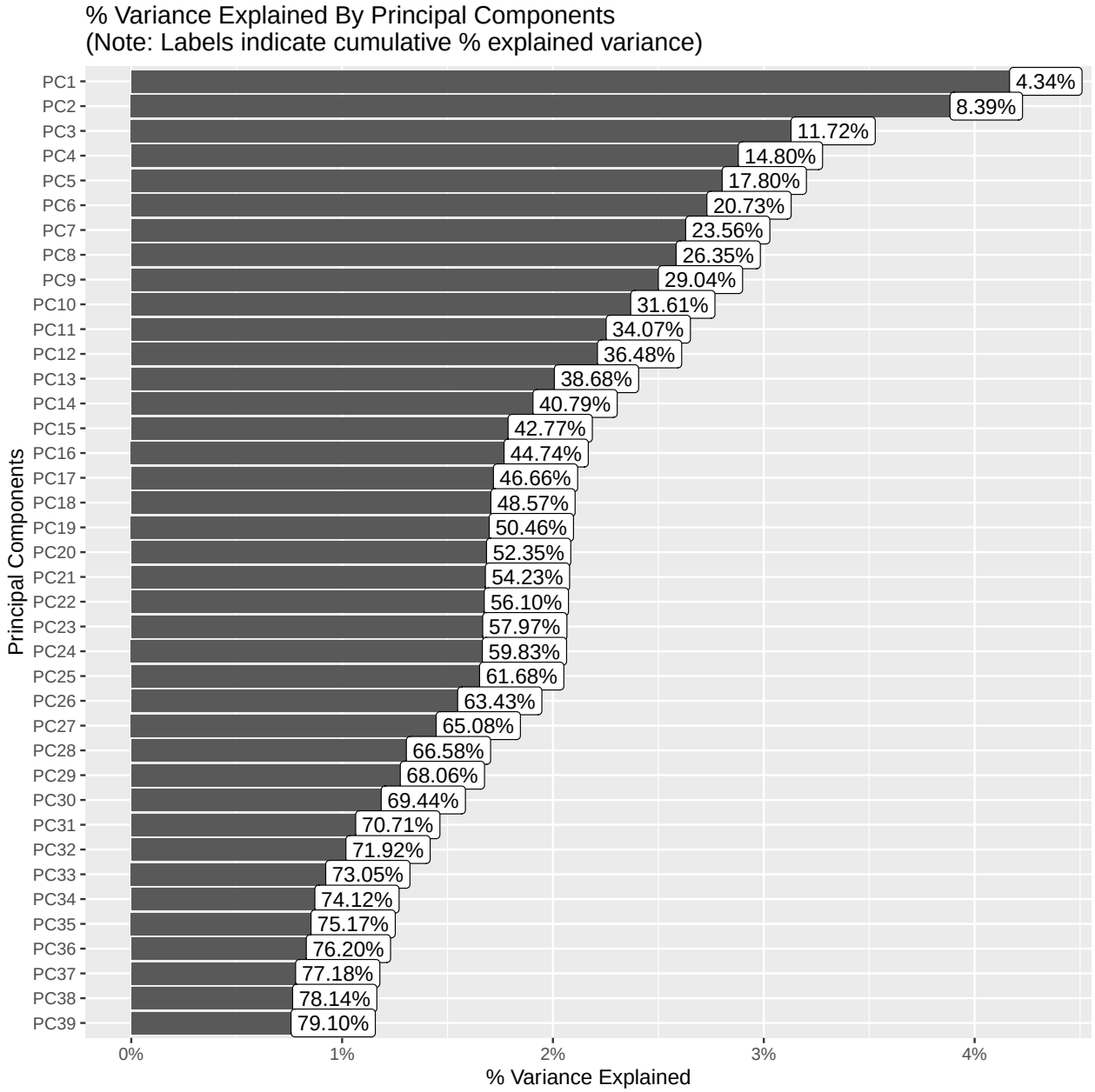
```
## 2 columns ignored with more than 50 categories.
## ZIP_CODE: 1799 categories
## full.address: 1799 categories
```

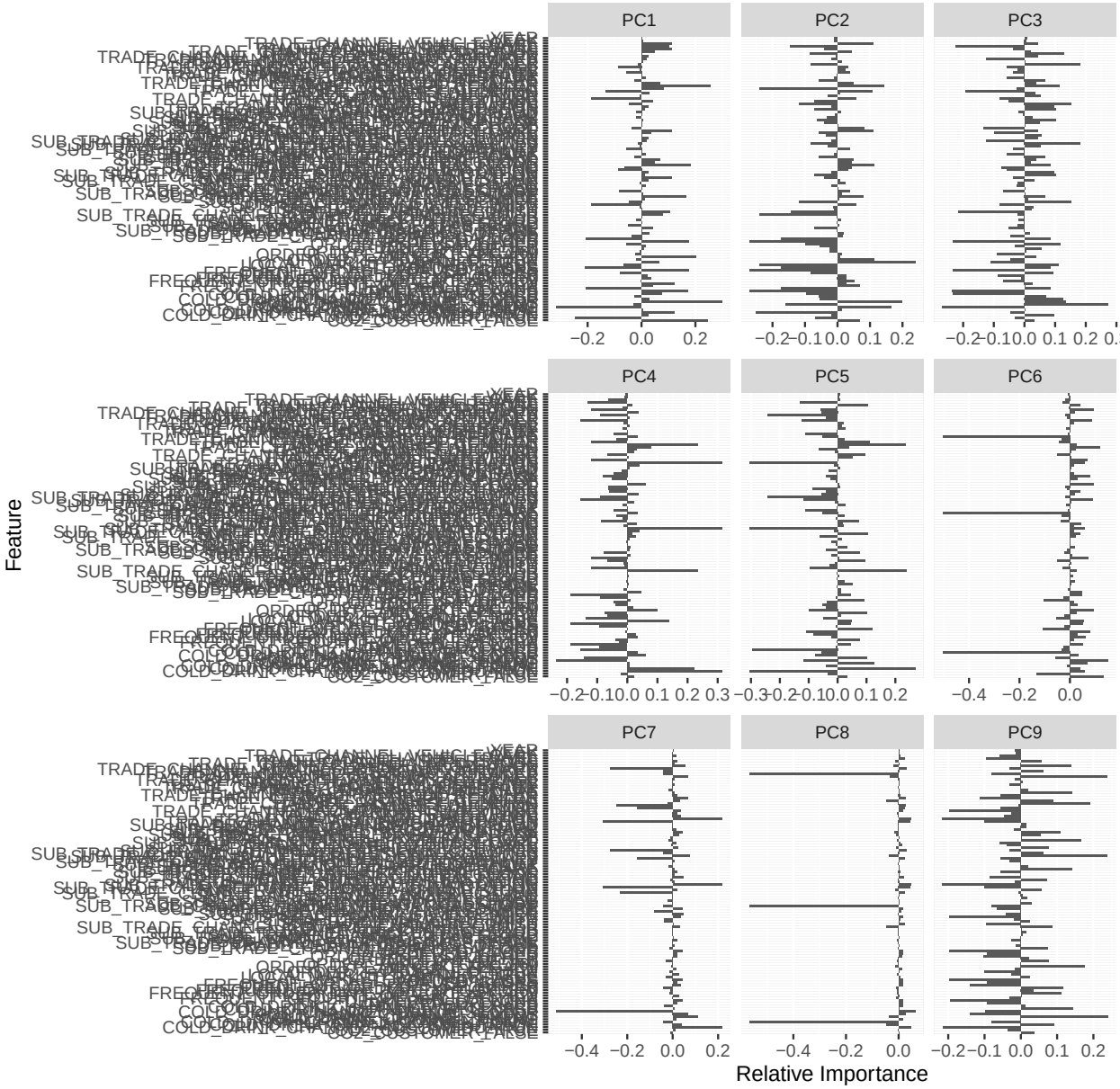


Principal Component Analysis

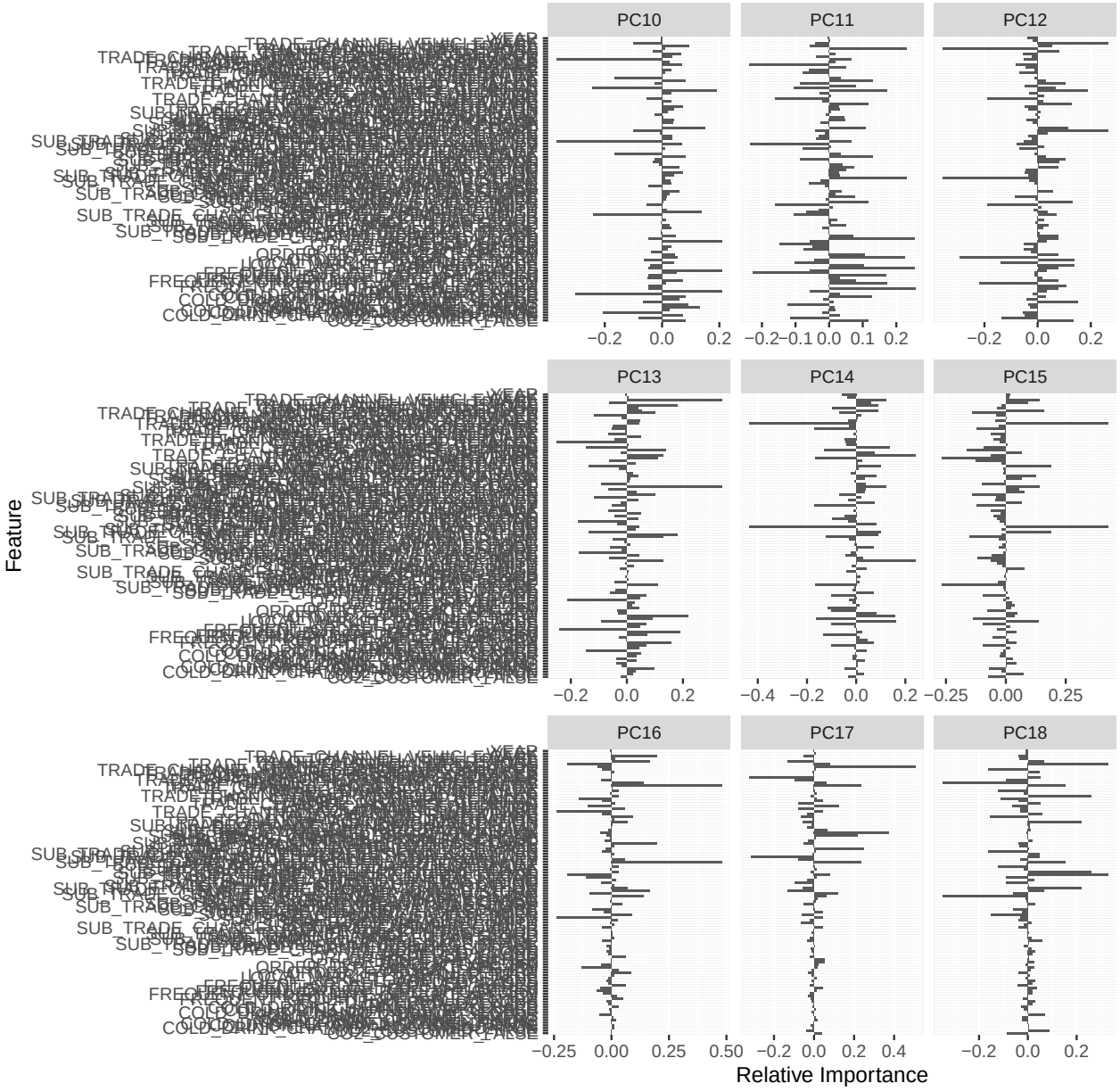
```
# Capture the PCA plots
pca_plots <- plot_prcomp(full_data)
```

```
## 7 features with more than 50 categories ignored!
## TRANSACTION_DATE: 723 categories
## CUSTOMER_NUMBER: 30322 categories
## PRIMARY_GROUP_NUMBER: 1021 categories
## FIRST_DELIVERY_DATE: 2401 categories
## ON_BOARDING_DATE: 6476 categories
## ZIP_CODE: 1799 categories
## full.address: 1799 categories
```

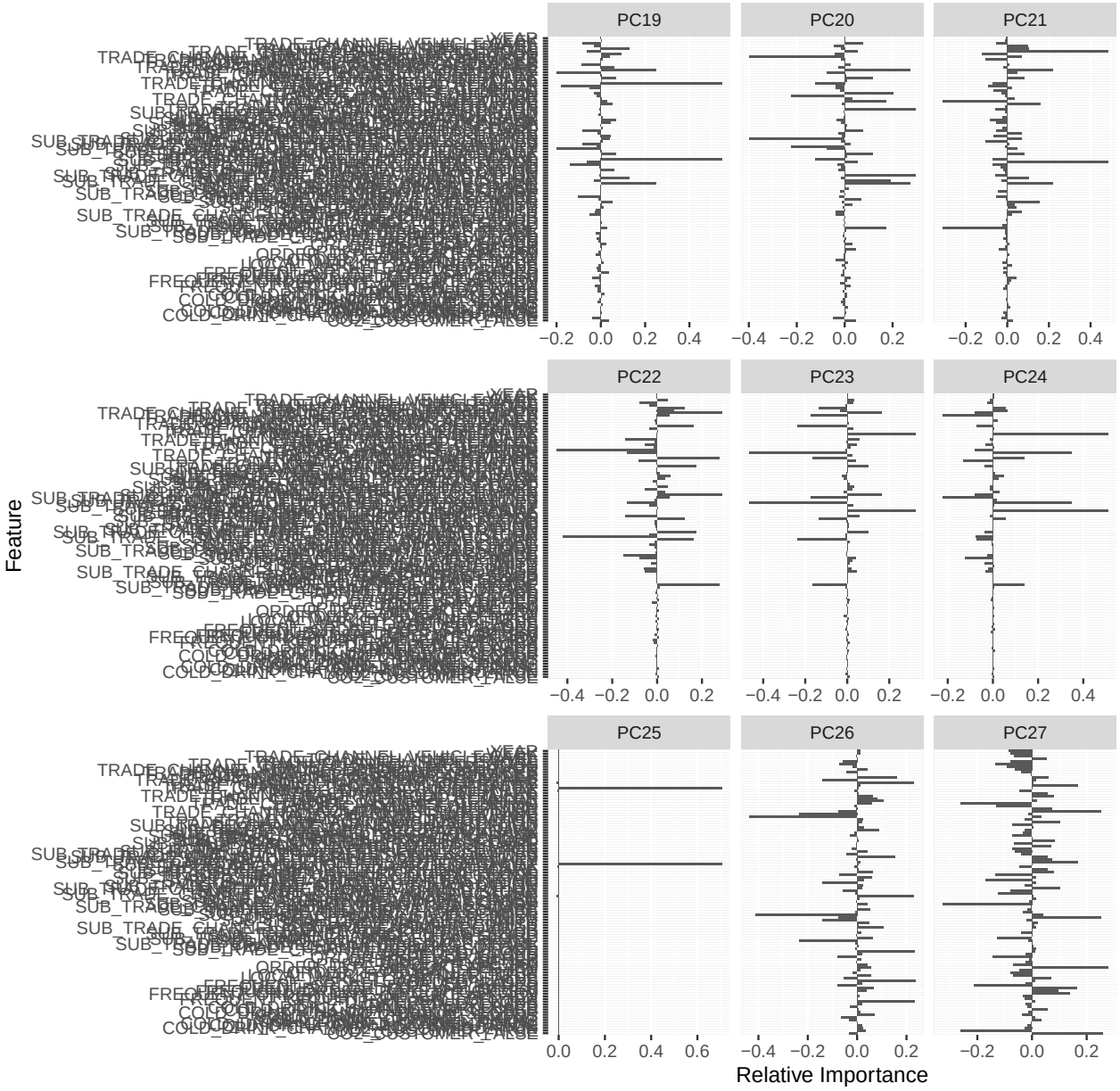

% Variance Explained By Principal Components
(Note: Labels indicate cumulative % explained variance)
Principal Components
PC1
PC2
PC3
PC4
PC5
PC6
PC7
PC8
PC9
PC10
PC11
PC12
PC13
PC14
PC15
PC16
PC17
PC18
PC19
PC20
PC21
PC22
PC23
PC24
PC25
PC26
PC27
PC28
PC29
PC30
PC31
PC32
PC33
PC34
PC35
PC36
PC37
PC38
PC39
4.34%
8.39%
11.72%
14.80%
17.80%
20.73%
23.56%
26.35%
29.04%
31.61%
34.07%
36.48%
38.68%
40.79%
42.77%
44.74%
46.66%
48.57%
50.46%
52.35%
54.23%
56.10%
57.97%
59.83%
61.68%
63.43%
65.08%
66.58%
68.06%
69.44%
70.71%
71.92%
73.05%
74.12%
75.17%
76.20%
77.18%
78.14%
79.10%
0%
1%
2%
3%
4%
% Variance Explained

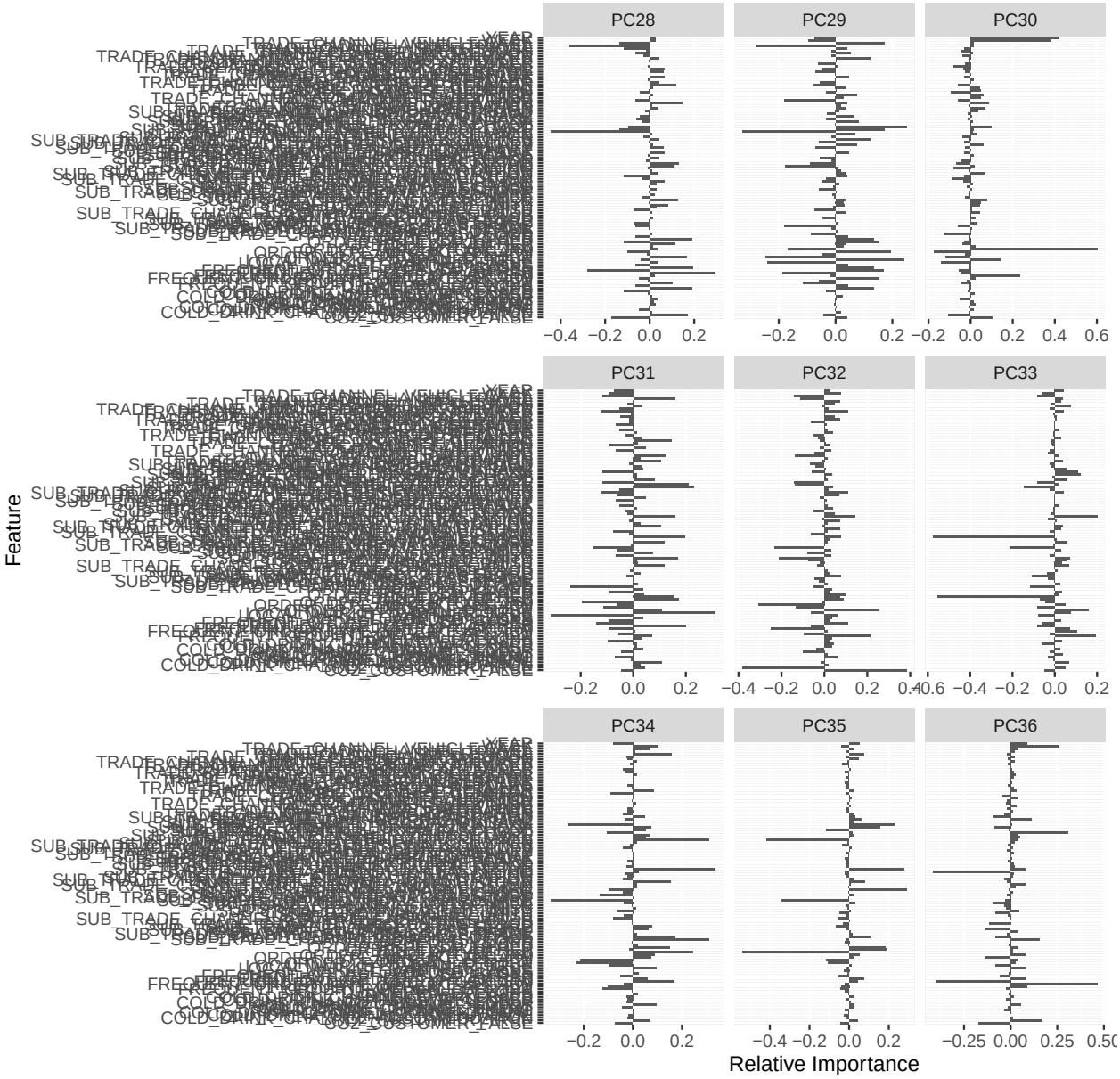

PC1
PC2
PC3
PC4
PC5
PC6
PC7
PC8
PC9
Feature
Relative Importance

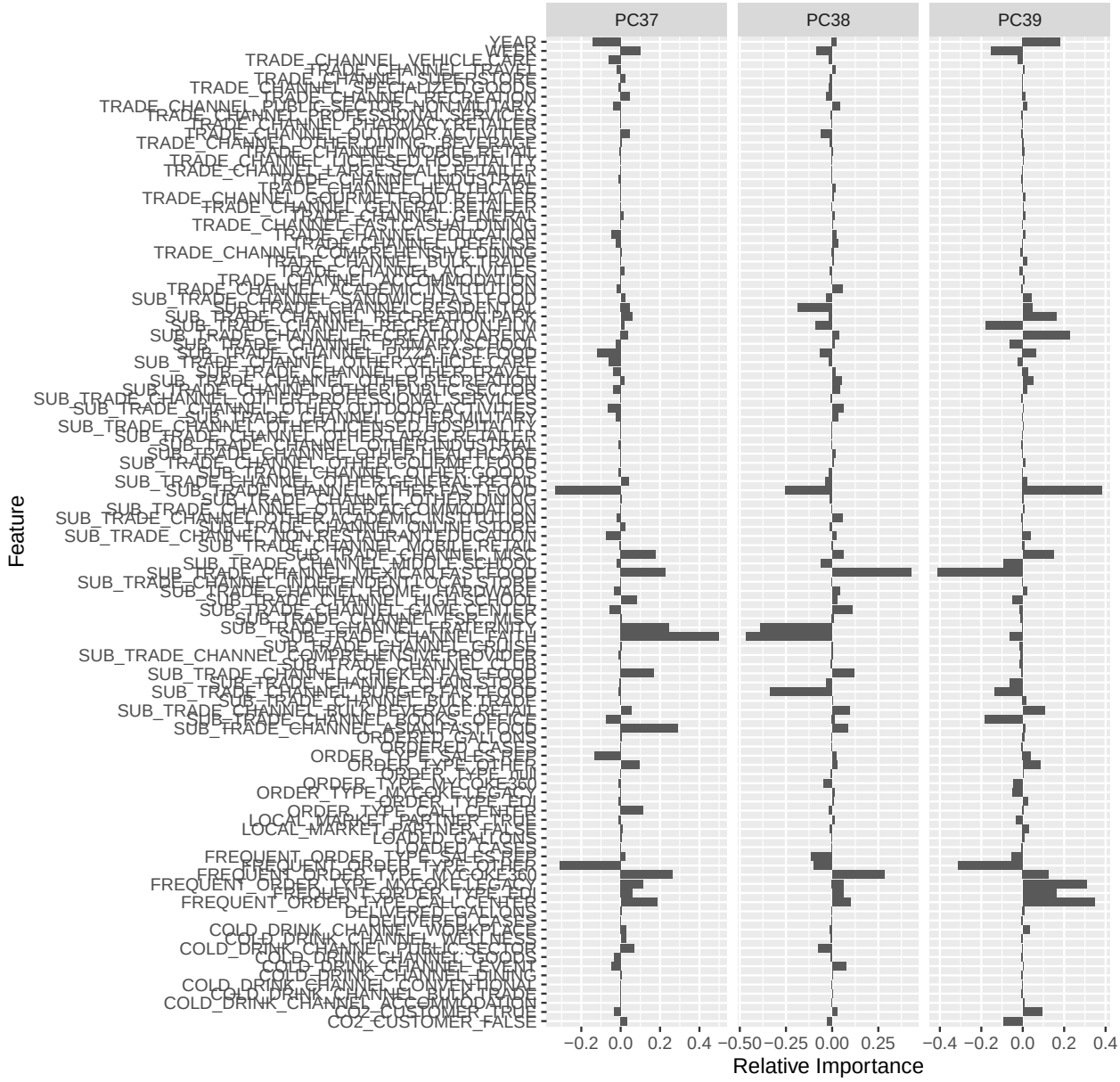
Page 1


PC10
PC11
PC12
PC13
PC14
PC15
PC16
PC17
PC18
Feature
Relative Importance
−0.2 0.0 0.2
−0.2 −0.1 0.0 0.1 0.2
−0.4 −0.2 0.0 0.2
−0.25 0.00 0.25
−0.25 0.00 0.25 0.50
−0.2 0.0 0.2 0.4

Page 2

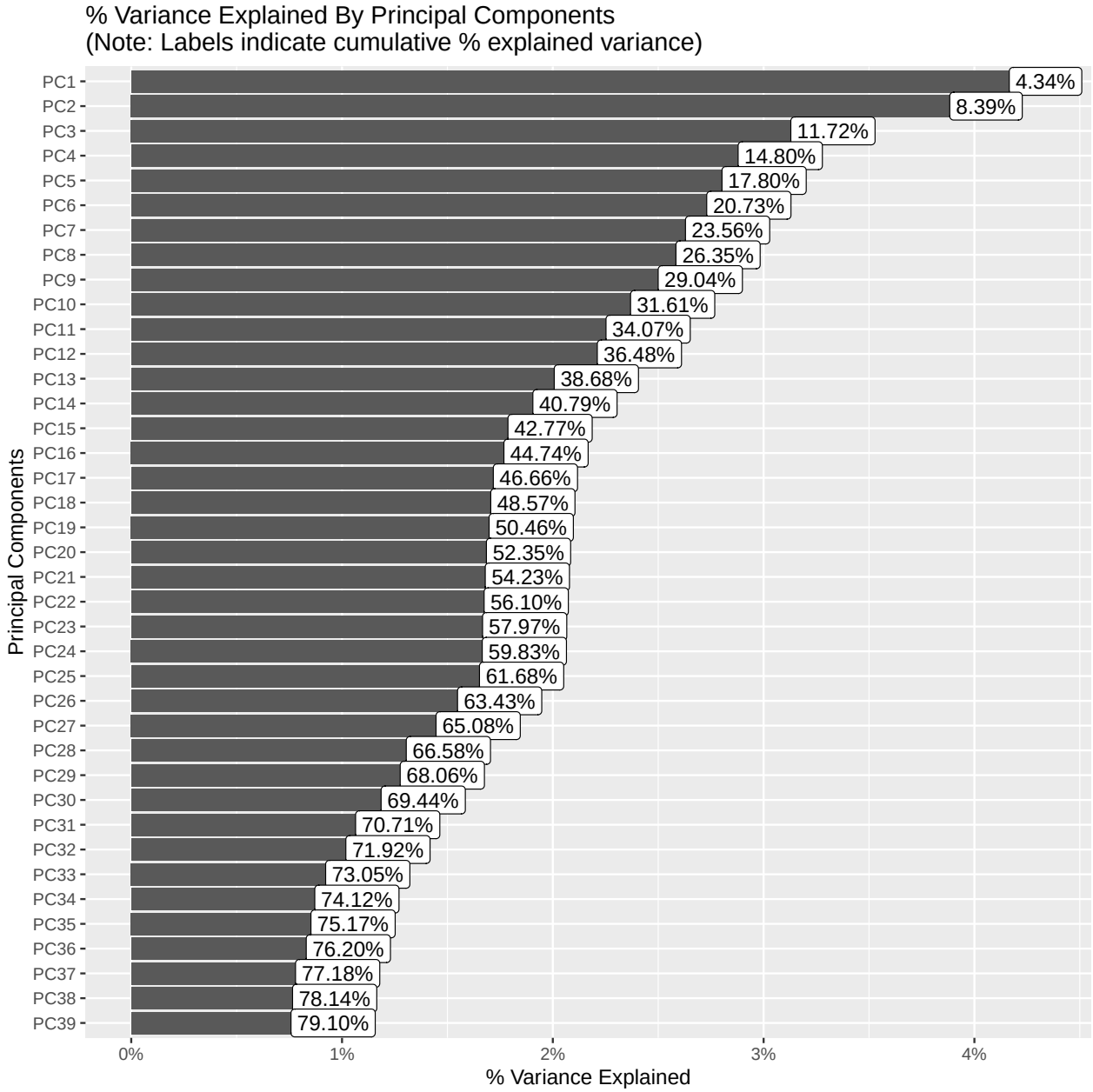

PC19
PC20
PC21
PC22
PC23
PC24
PC25
PC26
PC27
Feature
Relative Importance


PC28
PC29
PC30
PC31
PC32
PC33
PC34
PC35
PC36
Feature
Relative Importance



Page 5

```
# Print only the first PCA plot
print(pca_plots[[1]])
```

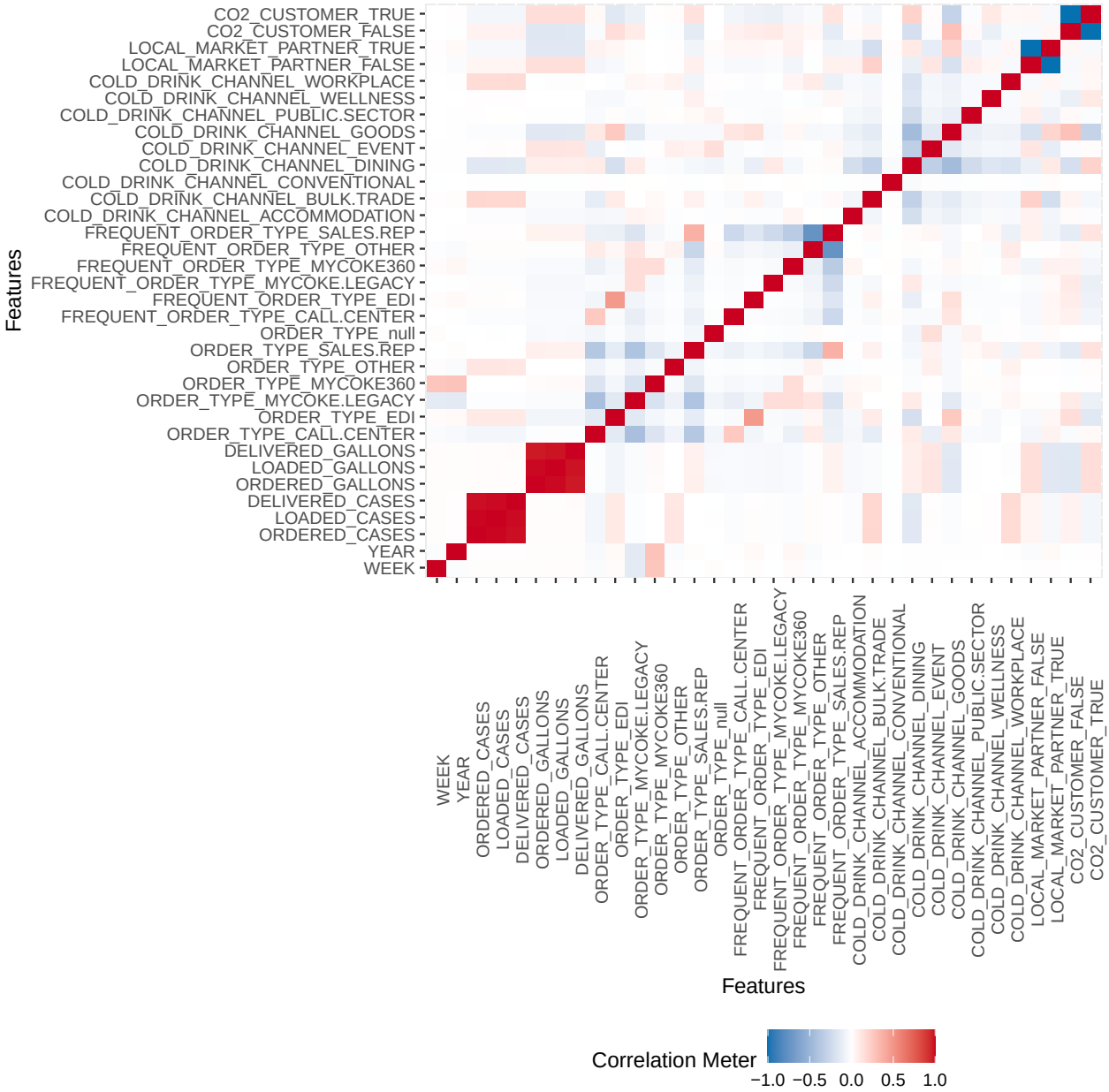


Correlation Analysis

```
full_data |>
  plot_correlation()
```

```
## 9 features with more than 20 categories ignored!
## TRANSACTION_DATE: 723 categories
## CUSTOMER_NUMBER: 30322 categories
## PRIMARY_GROUP_NUMBER: 1021 categories
## FIRST_DELIVERY_DATE: 2401 categories
## ON_BOARDING_DATE: 6476 categories
## TRADE_CHANNEL: 26 categories
## SUB_TRADE_CHANNEL: 48 categories
## ZIP_CODE: 1799 categories
```

```
## full.address: 1799 categories
```



Feature Engineering

Should Volume Range be Set Up as a Binning Variable?

I believe it should be set as a binning variable

```
# Checking max amount of ordered/loaded/delivered gallons/cases.
# summary(full_data)

# Will make sure it is set up as a factor
delivery_data <- delivery_cost |>
  mutate(`Vol Range` = as.factor(`Vol Range`))

# Drop Cost Type
```

```r
delivery_data <- delivery_data |>
  dplyr::select(-`Cost Type`)
```

Add Total Delivery Costs Calculation

If a customer in the "AT WORK" channel purchases Fountain drinks and Bottles/Cans, the total delivery cost is calculated as follows: Determine the volume range for each category (Fountain and Bottles/Cans). Retrieve the median delivery cost per unit from the dataset. Multiply the cost per unit by the quantity purchased for each category. Sum the delivery costs to get the total.

A customer orders: Fountain: 600 gallons annually (Volume Range: 600 - 749) → Cost: $1.18 per gallon Bottles and Cans: 400 cases annually (Volume Range: 300 - 449) → Cost: $2.99 per case

The total delivery cost is calculated as follows: Fountain: $600 \times 1.18 = 708.00$ Bottles and Cans: $400 \times 2.99 = 1{,}196.00$

Total Delivery Cost: $708.00 + $1,196.00 = $1,904.00

Thus, the total annual delivery cost for this customer in the "AT WORK" channel is $1,904.00

How exactly should this be calculated? If we have negatives, do we just not calculate the cost?

Should this only be calculated by ordered or delivered? Cause I thought delivered but now I am unsure?

```r
# I don't think it is necessary to merge the datasets, just want to mutate????
# Going to calculate by delivered gallons + cases

# full_data$Total_Annual_Cost <- full_data |>
 # group_by(COLD_DRINK_CHANNEL, DELIVERED_CASES, DELIVERED_GALLONS) |>
 # summarise(total_annual_cost = sum(delivery_cost, na.rm = TRUE))

# Match up based on cold drink channel
# Compare by ordered_cases and ordered_gallons
```

Finalize Data Cleaning

Are there outliers?

What is the Distribution of the Target Variable?

How to Approach NAs?

Results

- deliveries with negative values are returns
- when there is a return it costs for the company slightly
- use returns as a metric for tiering the customers
- scenarios for returning coke: 1. maybe coke bottles broke during transit, so not full case

2. lots in warehouse where they are close to expiration date 3.

- caveat: customer can be newley on boarded
- Swire coca cola ordering gallon then they are ordering a fountain drink
- every customer number is an outlet,
- if there is a primary group number it is part of a chain

Future Plans

linear regression, predictive modeling, extracting featured importance,