Untitled

Rachit Biswas

2023-06-12

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.2.2
```

```
purchase_data=read.csv("E:\\PROJECTS\\FORAGE R\\QVI_purchase_behaviour
(1).csv")
print("Purchase Data")
```

```
## [1] "Purchase Data"
```

```
head(purchase_data)
```

```
##   LYLTY_CARD_NBR               LIFESTAGE PREMIUM_CUSTOMER
## 1           1000  YOUNG SINGLES/COUPLES          Premium
## 2           1002  YOUNG SINGLES/COUPLES       Mainstream
## 3           1003         YOUNG FAMILIES           Budget
## 4           1004  OLDER SINGLES/COUPLES       Mainstream
## 5           1005 MIDAGE SINGLES/COUPLES       Mainstream
## 6           1007  YOUNG SINGLES/COUPLES           Budget
```

```
transcation_data=read_xlsx("E:\\PROJECTS\\FORAGE R\\QVI_transaction_data
(1).xlsx")
print("Transaction Data")
```

```
## [1] "Transaction Data"
```

```
head(transcation_data)
```

```
## # A tibble: 6 × 8
##    DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR PROD_NAME       PROD_…¹
```

```
TOT_S…²
##   <dbl>     <dbl>          <dbl>  <dbl>    <dbl> <chr>              <dbl>
<dbl>
## 1 43390         1           1000      1        5 Natural Chip  …       2
6
## 2 43599         1           1307    348       66 CCs Nacho Chee…       3
6.3
## 3 43605         1           1343    383       61 Smiths Crinkle…       2
2.9
## 4 43329         2           2373    974       69 Smiths Chip Th…       5
15
## 5 43330         2           2426   1038      108 Kettle Tortill…       3
13.8
## 6 43604         4           4074   2982       57 Old El Paso Sa…       1
5.1
## # … with abbreviated variable names ¹PROD_QTY, ²TOT_SALES
```

```
print("NA value Check")
```

```
## [1] "NA value Check"
```

```
sum(is.na(transcation_data))
```

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

```
sum(is.na(purchase_data))
```

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

```
summary_stats <- purchase_data %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>%
  summarise(count = n())
```

```
## `summarise()` has grouped output by 'LIFESTAGE'. You can override using
the
## `.groups` argument.
```

```
# Print the summary statistics
print(summary_stats)
```

```
## # A tibble: 21 × 3
## # Groups:   LIFESTAGE [7]
##    LIFESTAGE              PREMIUM_CUSTOMER count
##    <chr>                  <chr>            <int>
##  1 MIDAGE SINGLES/COUPLES Budget            1504
##  2 MIDAGE SINGLES/COUPLES Mainstream        3340
##  3 MIDAGE SINGLES/COUPLES Premium           2431
##  4 NEW FAMILIES           Budget            1112
##  5 NEW FAMILIES           Mainstream         849
##  6 NEW FAMILIES           Premium            588
##  7 OLDER FAMILIES         Budget            4675
##  8 OLDER FAMILIES         Mainstream        2831
```

```
##  9 OLDER FAMILIES         Premium           2274
## 10 OLDER SINGLES/COUPLES  Budget            4929
## # … with 11 more rows
```

```
#transaction Datat
print("Transaction Data")
```

```
## [1] "Transaction Data"
```

```
str(transcation_data)
```

```
## tibble [264,836 × 8] (S3: tbl_df/tbl/data.frame)
##  $ DATE          : num [1:264836] 43390 43599 43605 43329 43330 ...
##  $ STORE_NBR     : num [1:264836] 1 1 1 2 2 4 4 4 5 7 ...
##  $ LYLTY_CARD_NBR: num [1:264836] 1000 1307 1343 2373 2426 ...
##  $ TXN_ID        : num [1:264836] 1 348 383 974 1038 ...
##  $ PROD_NBR      : num [1:264836] 5 66 61 69 108 57 16 24 42 52 ...
##  $ PROD_NAME     : chr [1:264836] "Natural Chip        Compny SeaSalt175g"
"CCs Nacho Cheese    175g" "Smiths Crinkle Cut  Chips Chicken 170g" "Smiths
Chip Thinly  S/Cream&Onion 175g" ...
##  $ PROD_QTY      : num [1:264836] 2 3 2 5 3 1 1 1 1 2 ...
##  $ TOT_SALES     : num [1:264836] 6 6.3 2.9 15 13.8 5.1 5.7 3.6 3.9 7.2
...
```

```
summary(transcation_data)
```

```
##       DATE         STORE_NBR     LYLTY_CARD_NBR        TXN_ID
##  Min.   :43282   Min.   :  1.0   Min.   :   1000   Min.   :      1
##  1st Qu.:43373   1st Qu.: 70.0   1st Qu.:  70021   1st Qu.:  67602
##  Median :43464   Median :130.0   Median : 130358   Median : 135138
##  Mean   :43464   Mean   :135.1   Mean   : 135550   Mean   : 135158
##  3rd Qu.:43555   3rd Qu.:203.0   3rd Qu.: 203094   3rd Qu.: 202701
##  Max.   :43646   Max.   :272.0   Max.   :2373711   Max.   :2415841
##     PROD_NBR       PROD_NAME            PROD_QTY         TOT_SALES
##  Min.   :  1.00   Length:264836      Min.   :  1.000   Min.   :  1.500
##  1st Qu.: 28.00   Class :character   1st Qu.:  2.000   1st Qu.:  5.400
##  Median : 56.00   Mode  :character   Median :  2.000   Median :  7.400
##  Mean   : 56.58                      Mean   :  1.907   Mean   :  7.304
##  3rd Qu.: 85.00                      3rd Qu.:  2.000   3rd Qu.:  9.200
##  Max.   :114.00                      Max.   :200.000   Max.   :650.000
```

```
summary_stats2 <- transcation_data %>%
  summarise(
    total_sales = sum(TOT_SALES),
    total_quantity = sum(PROD_QTY),
    num_transactions = n(),
    avg_sales_per_transaction = mean(TOT_SALES),
    top_selling_products = paste(unique(PROD_NAME), collapse = ", ")
  )
```

```r
# Print the summary statistics
print(summary_stats2)
```

```
## # A tibble: 1 × 5
##   total_sales total_quantity num_transactions avg_sales_per_transaction
top_se…¹
##         <dbl>          <dbl>            <int>                      <dbl>
<chr>
## 1     1934415         505124           264836                       7.30
Natural…
## # … with abbreviated variable name ¹top_selling_products
```

```r
print("outlier detection")
```

```
## [1] "outlier detection"
```

```r
# Identify outliers using the IQR method
outliers <- transcation_data %>%
  filter(TOT_SALES > quantile(TOT_SALES, 0.75) + 1.5 * IQR(TOT_SALES) |
           TOT_SALES < quantile(TOT_SALES, 0.25) - 1.5 * IQR(TOT_SALES))

# Print the outliers
print(outliers)
```

```
## # A tibble: 578 × 8
##     DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR PROD_NAME      PROD_…¹
TOT_S…²
##    <dbl>     <dbl>          <dbl>  <dbl>    <dbl> <chr>            <dbl>
<dbl>
##  1 43329         2           2373    974       69 Smiths Chip T…       5
15
##  2 43332         8           8294   8221      114 Kettle Sensat…       5
23
##  3 43601        74          74336  73182       84 GrnWves Plus …       5
15.5
##  4 43331        96          96203  96025        7 Smiths Crinkl…       5
28.5
##  5 43605       130         130108 134125        2 Cobs Popd Sou…       5
19
##  6 43600       133         133250 137666       30 Doritos Corn …       4
17.6
##  7 43602       168         168219 170719       33 Cobs Popd Swt…       4
15.2
##  8 43602       222         222209 222693       40 Thins Chips S…       5
16.5
##  9 43329       257         257258 257308      114 Kettle Sensat…       4
18.4
## 10 43331       262         262126 262025      108 Kettle Tortil…       4
18.4
## # … with 568 more rows, and abbreviated variable names ¹PROD_QTY, ²
TOT_SALES
```

```r
library(tinytex)
```

```
## Warning: package 'tinytex' was built under R version 4.2.2
```

```r
print("removing")
```

```
## [1] "removing"
```

```r
q1 <- quantile(transcation_data$TOT_SALES, 0.25)
q3 <- quantile(transcation_data$TOT_SALES, 0.75)
iqr <- q3 - q1
lower_threshold <- q1 - 1.5 * iqr
upper_threshold <- q3 + 1.5 * iqr

# Remove outliers
transaction_data <- transcation_data %>%
  filter(TOT_SALES >= lower_threshold, TOT_SALES <= upper_threshold)

View(transaction_data)

#MUTATE

transaction_data <- transaction_data %>%
  mutate(
    PACK_SIZE = as.numeric(gsub("[^0-9]", "", PROD_NAME)),
    BRAND_NAME = gsub("[0-9]", "", PROD_NAME)
  )

# Print the updated dataset
head(transaction_data)
```

```
## # A tibble: 6 × 10
##    DATE STORE_NBR LYLTY…¹ TXN_ID PROD_…² PROD_…³ PROD_…⁴ TOT_S…⁵ PACK_…⁶
BRAND…⁷
##   <dbl>     <dbl>   <dbl>  <dbl>   <dbl> <chr>     <dbl>   <dbl>   <dbl>
<chr>
## 1 43390         1    1000      1       5 Natura…       2     6       175
Natura…
## 2 43599         1    1307    348      66 CCs Na…       3     6.3     175
CCs Na…
## 3 43605         1    1343    383      61 Smiths…       2     2.9     170
Smiths…
## 4 43330         2    2426   1038     108 Kettle…       3    13.8     150
Kettle…
## 5 43604         4    4074   2982      57 Old El…       1     5.1     300
Old El…
## 6 43601         4    4149   3333      16 Smiths…       1     5.7     330
Smiths…
## # … with abbreviated variable names ¹LYLTY_CARD_NBR, ²PROD_NBR, ³
```

```
PROD_NAME,
## #   ⁴PROD_QTY, ⁵TOT_SALES, ⁶PACK_SIZE, ⁷BRAND_NAME

transaction_data <- merge(transaction_data, purchase_data, by =
"LYLTY_CARD_NBR")
head(transaction_data)

##   LYLTY_CARD_NBR  DATE STORE_NBR TXN_ID PROD_NBR
## 1           1000 43390         1      1        5
## 2           1002 43359         1      2       58
## 3           1003 43532         1      4      106
## 4           1003 43531         1      3       52
## 5           1004 43406         1      5       96
## 6           1005 43462         1      6       86
##                               PROD_NAME PROD_QTY TOT_SALES PACK_SIZE
## 1 Natural Chip        Compny SeaSalt175g        2       6.0       175
## 2  Red Rock Deli Chikn&Garlic Aioli 150g        1       2.7       150
## 3 Natural ChipCo      Hony Soy Chckn175g        1       3.0       175
## 4  Grain Waves Sour    Cream&Chives 210G        1       3.6       210
## 5         WW Original Stacked Chips 160g        1       1.9       160
## 6                     Cheetos Puffs 165g        1       2.8       165
##                          BRAND_NAME              LIFESTAGE
PREMIUM_CUSTOMER
## 1 Natural Chip        Compny SeaSaltg  YOUNG SINGLES/COUPLES
Premium
## 2  Red Rock Deli Chikn&Garlic Aioli g  YOUNG SINGLES/COUPLES
Mainstream
## 3 Natural ChipCo      Hony Soy Chckng         YOUNG FAMILIES
Budget
## 4  Grain Waves Sour    Cream&Chives G         YOUNG FAMILIES
Budget
## 5         WW Original Stacked Chips g  OLDER SINGLES/COUPLES
Mainstream
## 6                     Cheetos Puffs g MIDAGE SINGLES/COUPLES
Mainstream

metrics <- transaction_data %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>%
  summarise(
    total_spending = sum(TOT_SALES),
    average_spending = mean(TOT_SALES),
    total_quantity = sum(PROD_QTY),
    average_price_per_chip = sum(TOT_SALES) / sum(PROD_QTY),
    purchase_frequency = n(),
    top_brand = names(which.max(table(BRAND_NAME))),
    top_pack_size = names(which.max(table(PACK_SIZE)))
  )

## `summarise()` has grouped output by 'LIFESTAGE'. You can override using
the
## `.groups` argument.
```

```r
# Print the metrics
print(metrics)
```

```
## # A tibble: 21 × 9
## # Groups:   LIFESTAGE [7]
##    LIFESTAGE     PREMI…¹ total…² avera…³ total…⁴ avera…⁵ purch…⁶ top_b…⁷
top_p…⁸
##    <chr>         <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <int> <chr>
<chr>
##  1 MIDAGE SINGL… Budget   35309.    7.05    9445    3.74    5009 Infzns…
175
##  2 MIDAGE SINGL… Mainst…  90178.    7.61   22561    4.00   11843 Smiths…
175
##  3 MIDAGE SINGL… Premium  58096.    7.09   15449    3.76    8199 Pringl…
175
##  4 NEW FAMILIES  Budget   21862.    7.28    5558    3.93    3002 Kettle…
175
##  5 NEW FAMILIES  Mainst…  16940.    7.30    4301    3.94    2321 Kettle…
175
##  6 NEW FAMILIES  Premium  11450.    7.22    2948    3.88    1587 Grain …
175
##  7 OLDER FAMILI… Budget  167214.    7.24   44816    3.73   23104 Smiths…
175
##  8 OLDER FAMILI… Mainst… 102669.    7.23   27576    3.72   14204 Old El…
175
##  9 OLDER FAMILI… Premium  80062.    7.18   21626    3.70   11158 Infuzi…
175
## 10 OLDER SINGLE… Budget  135859.    7.40   35022    3.88   18361 Cobs P…
175
## # … with 11 more rows, and abbreviated variable names ¹PREMIUM_CUSTOMER,
## #   ²total_spending, ³average_spending, ⁴total_quantity,
## #   ⁵average_price_per_chip, ⁶purchase_frequency, ⁷top_brand, ⁸
top_pack_size
```