Quantium Internship -Task 1

Bolanle Ogunlola

2024-10-15

Loading Require library

```
library(data.table)
library(ggplot2)
library(readr)
library(readxl)
library(dplyr)
```

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

## The following objects are masked from 'package:data.table':
##
##     between, first, last

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

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

```
library(ggmosaic)
```

Importing Data

```
QVI_purchase_behaviour <- read_csv("QVI_purchase_behaviour.csv")
```

```
## Rows: 72637 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): LIFESTAGE, PREMIUM_CUSTOMER
## dbl (1): LYLTY_CARD_NBR
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
View(QVI_purchase_behaviour)
QVI_transaction_data <- read_excel("QVI_transaction_data.xlsx")
View(QVI_transaction_data)
```

Rename data

Renaming the data for easy access.

```
transactionData <- rename(QVI_transaction_data)
customerData <- rename(QVI_purchase_behaviour)
```

Explore data

```
str(transactionData)
```

```
## tibble [264,836 x 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
##  $ 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 ...
```

```
head(transactionData)
```

```
## # A tibble: 6 x 8
##    DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR PROD_NAME    PROD_QTY TOT_SALES
##   <dbl>     <dbl>          <dbl>  <dbl>    <dbl> <chr>           <dbl>     <dbl>
## 1 43390         1           1000      1        5 Natural Chi~        2       6
## 2 43599         1           1307    348       66 CCs Nacho C~        3       6.3
## 3 43605         1           1343    383       61 Smiths Crin~        2       2.9
## 4 43329         2           2373    974       69 Smiths Chip~        5      15
## 5 43330         2           2426   1038      108 Kettle Tort~        3      13.8
## 6 43604         4           4074   2982       57 Old El Paso~        1       5.1
```

```
glimpse(transactionData)
```

```
## Rows: 264,836
## Columns: 8
## $ DATE          <dbl> 43390, 43599, 43605, 43329, 43330, 43604, 43601, 43601,~
## $ STORE_NBR     <dbl> 1, 1, 1, 2, 2, 4, 4, 4, 5, 7, 7, 8, 9, 13, 19, 20, 20, ~
## $ LYLTY_CARD_NBR <dbl> 1000, 1307, 1343, 2373, 2426, 4074, 4149, 4196, 5026, 7~
## $ TXN_ID        <dbl> 1, 348, 383, 974, 1038, 2982, 3333, 3539, 4525, 6900, 7~
## $ PROD_NBR      <dbl> 5, 66, 61, 69, 108, 57, 16, 24, 42, 52, 16, 114, 15, 92~
## $ PROD_NAME     <chr> "Natural Chip        Compny SeaSalt175g", "CCs Nacho Ch~
## $ PROD_QTY      <dbl> 2, 3, 2, 5, 3, 1, 1, 1, 1, 2, 1, 5, 2, 1, 1, 1, 4, 1, 1~
## $ TOT_SALES     <dbl> 6.0, 6.3, 2.9, 15.0, 13.8, 5.1, 5.7, 3.6, 3.9, 7.2, 5.7~
```

Convert Date to date format

The date currently appears in a number format. Using the as.Date syntax to convert it to the correct format.

```
transactionData$DATE <- as.Date(transactionData$DATE,origin = "1899-12-30")
head(transactionData)
```

```
## # A tibble: 6 x 8
##   DATE       STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR PROD_NAME         PROD_QTY
##   <date>         <dbl>          <dbl>  <dbl>    <dbl> <chr>                <dbl>
## 1 2018-10-17         1           1000      1        5 Natural Chip    ~        2
## 2 2019-05-14         1           1307    348       66 CCs Nacho Cheese~        3
## 3 2019-05-20         1           1343    383       61 Smiths Crinkle C~        2
## 4 2018-08-17         2           2373    974       69 Smiths Chip Thin~        5
## 5 2018-08-18         2           2426   1038      108 Kettle Tortilla ~        3
## 6 2019-05-19         4           4074   2982       57 Old El Paso Sals~        1
## # i 1 more variable: TOT_SALES <dbl>
```

Summary of PRODUCT NAME

```
transactionData.summary <- transactionData
head.matrix(transactionData.summary)
```

```
## # A tibble: 6 x 8
##   DATE       STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR PROD_NAME         PROD_QTY
##   <date>         <dbl>          <dbl>  <dbl>    <dbl> <chr>                <dbl>
## 1 2018-10-17         1           1000      1        5 Natural Chip    ~        2
## 2 2019-05-14         1           1307    348       66 CCs Nacho Cheese~        3
## 3 2019-05-20         1           1343    383       61 Smiths Crinkle C~        2
## 4 2018-08-17         2           2373    974       69 Smiths Chip Thin~        5
## 5 2018-08-18         2           2426   1038      108 Kettle Tortilla ~        3
## 6 2019-05-19         4           4074   2982       57 Old El Paso Sals~        1
## # i 1 more variable: TOT_SALES <dbl>
```

Further examaine product name

```
unique_products <- unique(transactionData$PROD_NAME)
split_products <- unlist(strsplit(unique_products, ","))
productWords <- data.table(split_products)
setnames(productWords, "split_products", "words")
View(productWords)
head(productWords)
```

```
##                                     words
##                                    <char>
## 1:   Natural Chip        Compny SeaSalt175g
## 2:                 CCs Nacho Cheese    175g
## 3:   Smiths Crinkle Cut  Chips Chicken 170g
## 4:   Smiths Chip Thinly  S/Cream&Onion 175g
## 5: Kettle Tortilla ChpsHny&Jlpno Chili 150g
## 6: Old El Paso Salsa   Dip Tomato Mild 300g
```

Remove special character and digit

```
Cleaned_Product_name <- gsub ("[^a-z,A-Z]", "", transactionData$PROD_NAME)
Clean_product <- data.table(Cleaned_Product_name)
setnames(productWords, "words")
View(productWords)
head(productWords)
```

```
##                                     words
##                                    <char>
## 1:   Natural Chip        Compny SeaSalt175g
## 2:                 CCs Nacho Cheese    175g
## 3:   Smiths Crinkle Cut  Chips Chicken 170g
## 4:   Smiths Chip Thinly  S/Cream&Onion 175g
## 5: Kettle Tortilla ChpsHny&Jlpno Chili 150g
## 6: Old El Paso Salsa   Dip Tomato Mild 300g
```

```
transactionData <- as.data.table(transactionData)
transactionData[, SALSA := grepl("salsa", tolower(PROD_NAME))]
transactionData <- transactionData[SALSA == FALSE, ][, SALSA := NULL]
View(transactionData)
```

Remove salsa products

```
summary(transactionData)
```

Summarise the data to check for nulls and possible outliers

```
##       DATE              STORE_NBR     LYLTY_CARD_NBR        TXN_ID
##  Min.   :2018-07-01   Min.   :  1.0   Min.   :   1000   Min.   :      1
##  1st Qu.:2018-09-30   1st Qu.: 70.0   1st Qu.:  70015   1st Qu.:  67569
##  Median :2018-12-30   Median :130.0   Median : 130367   Median : 135183
##  Mean   :2018-12-30   Mean   :135.1   Mean   : 135531   Mean   : 135131
##  3rd Qu.:2019-03-31   3rd Qu.:203.0   3rd Qu.: 203084   3rd Qu.: 202654
##  Max.   :2019-06-30   Max.   :272.0   Max.   :2373711   Max.   :2415841
##     PROD_NBR       PROD_NAME            PROD_QTY         TOT_SALES
##  Min.   :  1.00   Length:246742      Min.   :  1.000   Min.   :  1.700
##  1st Qu.: 26.00   Class :character   1st Qu.:  2.000   1st Qu.:  5.800
##  Median : 53.00   Mode  :character   Median :  2.000   Median :  7.400
##  Mean   : 56.35                      Mean   :  1.908   Mean   :  7.321
##  3rd Qu.: 87.00                      3rd Qu.:  2.000   3rd Qu.:  8.800
##  Max.   :114.00                      Max.   :200.000   Max.   :650.000
```

Filter the dataset to find the outlier

```
filtered_transactionData <- transactionData %>%
  filter(PROD_QTY == 200)
head(filtered_transactionData)
```

```
##          DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
##        <Date>     <num>          <num>  <num>    <num>
## 1: 2018-08-19       226         226000 226201        4
## 2: 2019-05-20       226         226000 226210        4
##                           PROD_NAME PROD_QTY TOT_SALES
##                              <char>    <num>     <num>
## 1: Dorito Corn Chp     Supreme 380g      200       650
## 2: Dorito Corn Chp     Supreme 380g      200       650
```

```
filtered_transactionData <- transactionData %>%
  filter(LYLTY_CARD_NBR == 226000)
head(filtered_transactionData)
```

Let's see if the customer has had other transactions

```
##          DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
##        <Date>     <num>          <num>  <num>    <num>
## 1: 2018-08-19       226         226000 226201        4
## 2: 2019-05-20       226         226000 226210        4
##                           PROD_NAME PROD_QTY TOT_SALES
##                              <char>    <num>     <num>
## 1: Dorito Corn Chp     Supreme 380g      200       650
## 2: Dorito Corn Chp     Supreme 380g      200       650
```

```
transactionData <- transactionData[LYLTY_CARD_NBR !=226000]
```

Filter out the customer based on the loyalty card number

```
summary(transactionData)
```

Re-examine transaction data

```
##       DATE              STORE_NBR     LYLTY_CARD_NBR        TXN_ID
##  Min.   :2018-07-01   Min.   :  1.0   Min.   :   1000   Min.   :      1
##  1st Qu.:2018-09-30   1st Qu.: 70.0   1st Qu.:  70015   1st Qu.:  67569
##  Median :2018-12-30   Median :130.0   Median : 130367   Median : 135182
##  Mean   :2018-12-30   Mean   :135.1   Mean   : 135530   Mean   : 135130
##  3rd Qu.:2019-03-31   3rd Qu.:203.0   3rd Qu.: 203083   3rd Qu.: 202652
##  Max.   :2019-06-30   Max.   :272.0   Max.   :2373711   Max.   :2415841
```

```
##     PROD_NBR       PROD_NAME            PROD_QTY       TOT_SALES
##  Min.   :  1.00   Length:246740      Min.   :1.000   Min.   : 1.700
##  1st Qu.: 26.00   Class :character   1st Qu.:2.000   1st Qu.: 5.800
##  Median : 53.00   Mode  :character   Median :2.000   Median : 7.400
##  Mean   : 56.35                      Mean   :1.906   Mean   : 7.316
##  3rd Qu.: 87.00                      3rd Qu.:2.000   3rd Qu.: 8.800
##  Max.   :114.00                      Max.   :5.000   Max.   :29.500
```

```
transactionData %>%
  count(DATE, sort = FALSE)
```

```
##            DATE     n
##          <Date> <int>
##   1: 2018-07-01   663
##   2: 2018-07-02   650
##   3: 2018-07-03   674
##   4: 2018-07-04   669
##   5: 2018-07-05   660
##  ---
## 360: 2019-06-26   657
## 361: 2019-06-27   669
## 362: 2019-06-28   673
## 363: 2019-06-29   703
## 364: 2019-06-30   704
```

Create a sequence of dates and join this the count of transactions by date

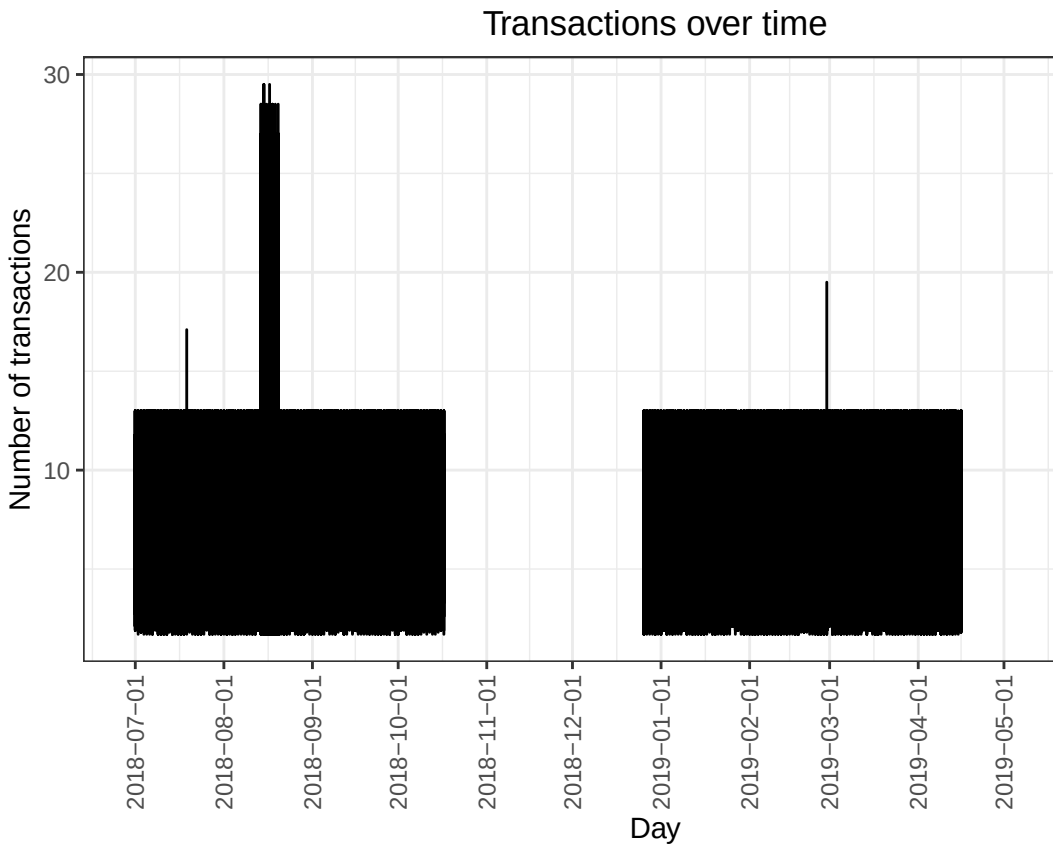
```
date_seq <- data.table(DATE = seq (as.Date("2018-07-01"), as.Date("2019-06-30"), by = "day"))
full_transactionData <- merge(date_seq, transactionData, by = "DATE", all.x = TRUE)
head(full_transactionData)
```

```
## Key: <DATE>
##          DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
##        <Date>     <num>          <num>  <num>    <num>
## 1: 2018-07-01        47          47142  42540       14
## 2: 2018-07-01        55          55073  48884       99
## 3: 2018-07-01        55          55073  48884       91
## 4: 2018-07-01        58          58351  54374      102
## 5: 2018-07-01        68          68193  65598       44
## 6: 2018-07-01        69          69207  67156       49
##                                  PROD_NAME PROD_QTY TOT_SALES
##                                     <char>    <num>     <num>
## 1:   Smiths Crnkle Chip  Orgnl Big Bag 380g        2      11.8
## 2:         Pringles Sthrn FriedChicken 134g        2       7.4
## 3:                 CCs Tasty Cheese    175g        2       4.2
## 4:   Kettle Mozzarella   Basil & Pesto 175g        2      10.8
## 5:           Thins Chips Light&  Tangy 175g        2       6.6
## 6: Infuzions SourCream&Herbs Veg Strws 110g        2       7.6
```

```
theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = 0.5))
```

Setting plot themes to format graphs

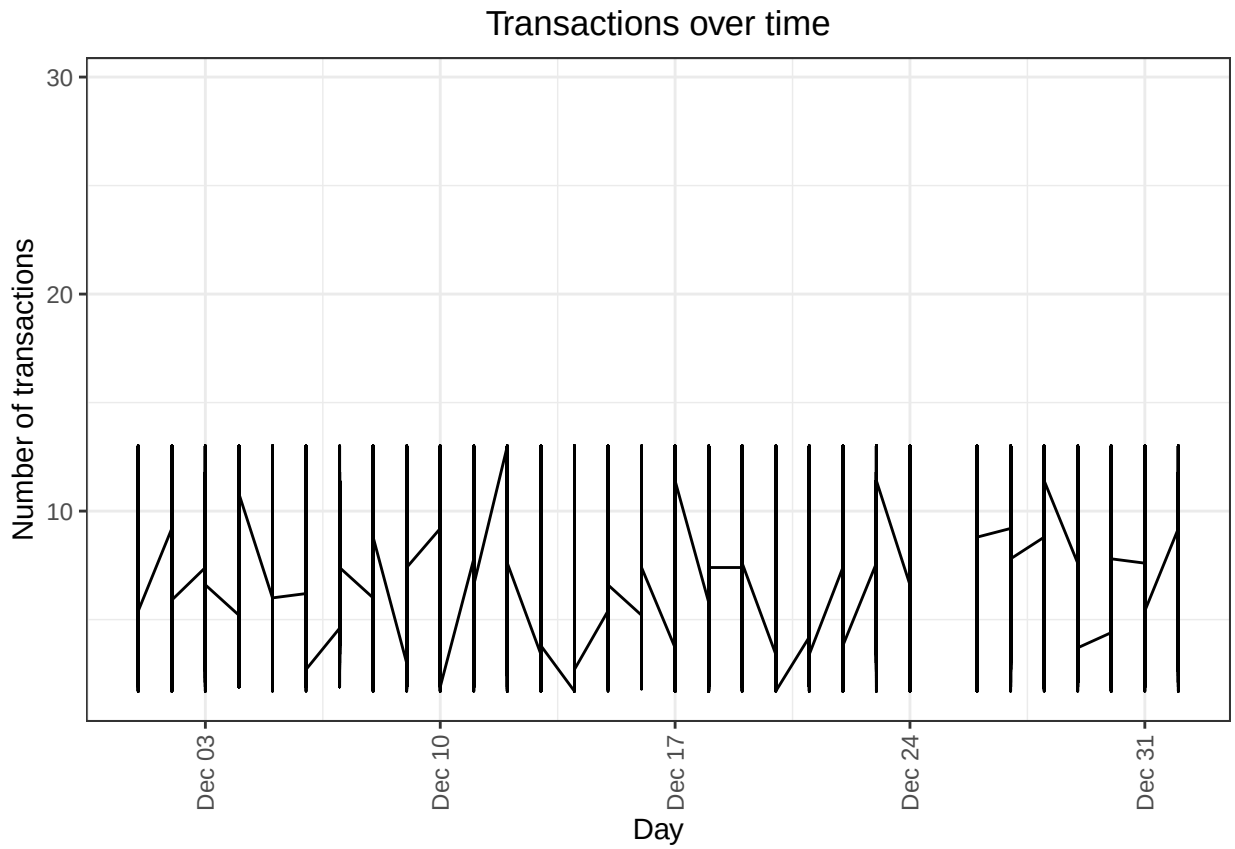
```
ggplot(full_transactionData, aes(x = DATE, y = TOT_SALES)) +
 geom_line() +
 labs(x = "Day", y = "Number of transactions", title = "Transactions over time") +
 scale_x_date(breaks = "1 month") +
 theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



Plot transactions over time

```
ggplot(full_transactionData, aes(x = DATE, y = TOT_SALES)) +
 geom_line() +
 labs(x = "Day", y = "Number of transactions", title = "Transactions over time") +
 scale_x_date(limits = as.Date(c("2018-12-01", "2019-01-01"))) +
 theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```

```
## Warning: Removed 224881 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
transactionData[, PACK_SIZE := parse_number(PROD_NAME)]
transactionData[, .N, PACK_SIZE][order(PACK_SIZE)]
```

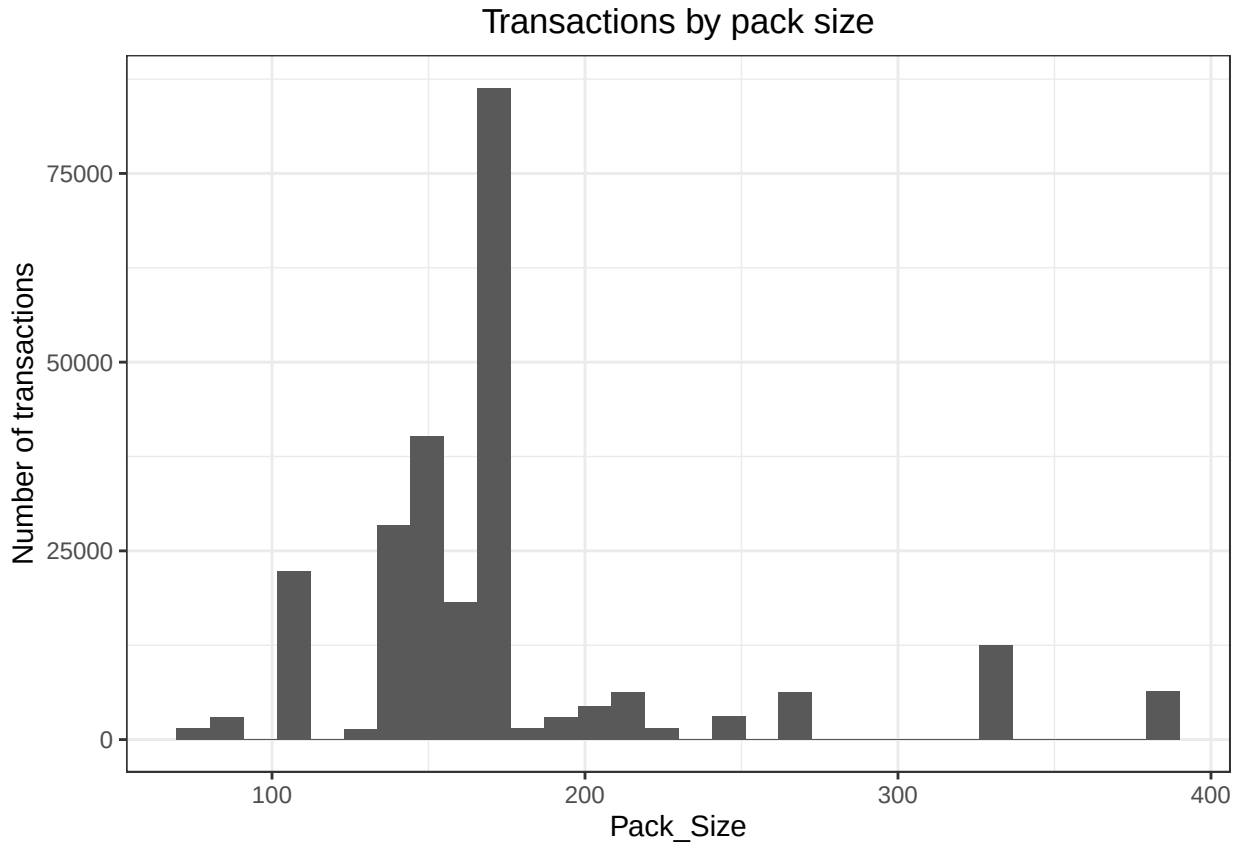
Pack size

```
##     PACK_SIZE     N
##         <num> <int>
##  1:        70  1507
##  2:        90  3008
##  3:       110 22387
##  4:       125  1454
##  5:       134 25102
##  6:       135  3257
##  7:       150 40203
##  8:       160  2970
##  9:       165 15297
## 10:       170 19983
## 11:       175 66390
## 12:       180  1468
## 13:       190  2995
## 14:       200  4473
## 15:       210  6272
## 16:       220  1564
```

```
## 17:       250  3169
## 18:       270  6285
## 19:       330 12540
## 20:       380  6416
##     PACK_SIZE     N
```

```
ggplot(transactionData, aes(PACK_SIZE))+
  geom_histogram()+
  labs(x = "Pack_Size", y = "Number of transactions", title = "Transactions by pack size")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Brand Name

```
transactionData[, BRAND_NAME := parse_character(PROD_NAME)]
transactionData[, .N, BRAND_NAME][order(BRAND_NAME)]
```

```
##                               BRAND_NAME     N
##                                   <char> <int>
##   1:                    Burger Rings 220g  1564
##   2:               CCs Nacho Cheese    175g  1498
##   3:                      CCs Original 175g  1514
##   4:               CCs Tasty Cheese    175g  1539
```

```
##   5:         Cheetos Chs & Bacon Balls 190g  1479
##  ---
## 101:         WW Original Corn    Chips 200g  1495
## 102:         WW Original Stacked Chips 160g  1487
## 103: WW Sour Cream &OnionStacked Chips 160g  1483
## 104:    WW Supreme Cheese   Corn Chips 200g  1509
## 105:         Woolworths Cheese   Rings 190g  1516
```

```
transactionData$BRAND_NAME[transactionData$BRAND_NAME == "RED"] <- "RRD"
```

Clean brand names

```
summary(customerData)
```

Examining customer data

```
##  LYLTY_CARD_NBR     LIFESTAGE         PREMIUM_CUSTOMER
##  Min.   :   1000   Length:72637       Length:72637
##  1st Qu.:  66202   Class :character   Class :character
##  Median : 134040   Mode  :character   Mode  :character
##  Mean   : 136186
##  3rd Qu.: 203375
##  Max.   :2373711
```

```
head(customerData)
```

```
## # A tibble: 6 x 3
##   LYLTY_CARD_NBR LIFESTAGE              PREMIUM_CUSTOMER
##            <dbl> <chr>                  <chr>
## 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
```

```
str(customerData)
```

```
## spc_tbl_ [72,637 x 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ LYLTY_CARD_NBR  : num [1:72637] 1000 1002 1003 1004 1005 ...
##  $ LIFESTAGE       : chr [1:72637] "YOUNG SINGLES/COUPLES" "YOUNG SINGLES/COUPLES" "YOUNG FAMILIES" "
##  $ PREMIUM_CUSTOMER: chr [1:72637] "Premium" "Mainstream" "Budget" "Mainstream" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   LYLTY_CARD_NBR = col_double(),
##   ..   LIFESTAGE = col_character(),
##   ..   PREMIUM_CUSTOMER = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Merge both Transaction and Customer Data into a single file

```
data <- merge(transactionData, customerData, all.x = TRUE)
```

Save file for future use

```
fwrite (data,"QVI_data.csv")
```

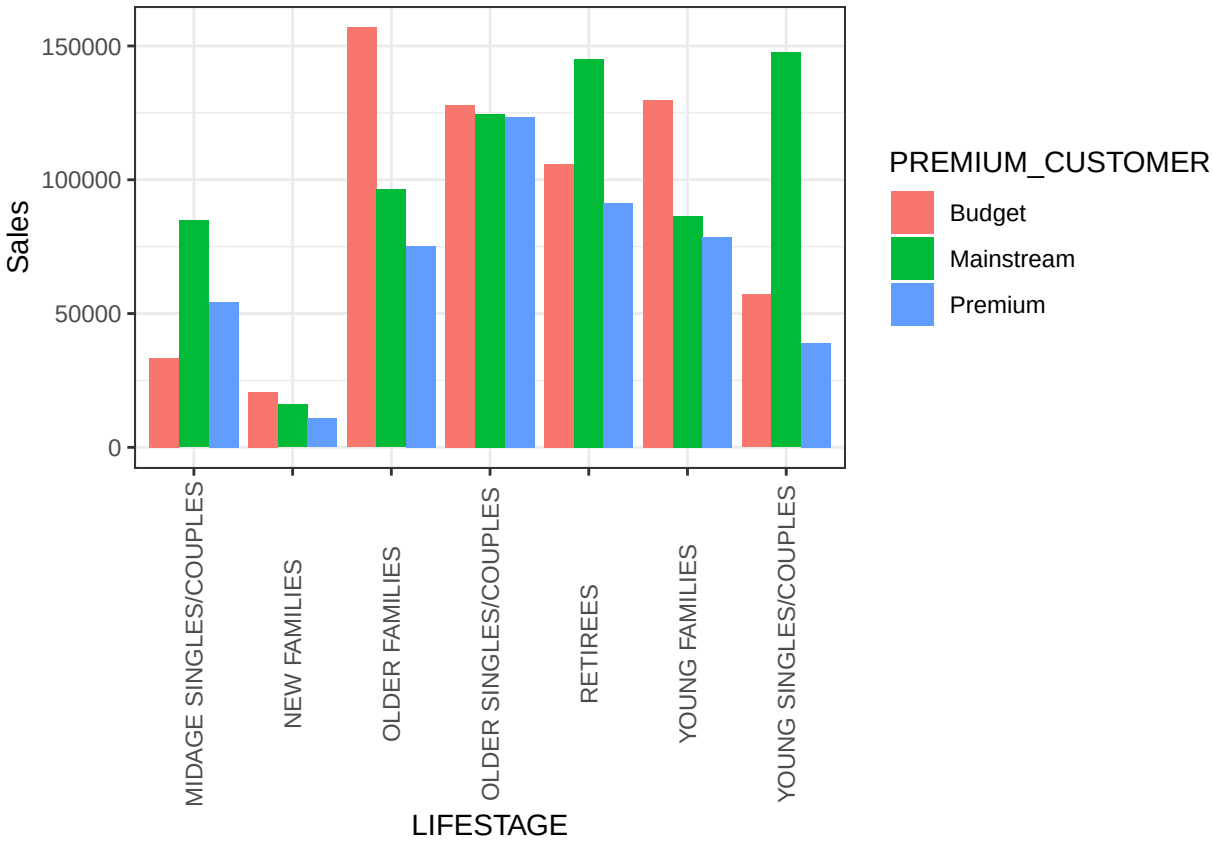
Data analysis on customer segments

```
Total_sales <- data %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER)%>%
  summarise(Sales = sum(TOT_SALES))%>%
  ungroup()
```

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

Plot graph

```
 ggplot(Total_sales, aes(x = LIFESTAGE, y = Sales, fill = PREMIUM_CUSTOMER)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Total Sales by Segment",
       x = "LIFESTAGE",
       y = "Total Sales",
       fill = "Premium Customer")%>%
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



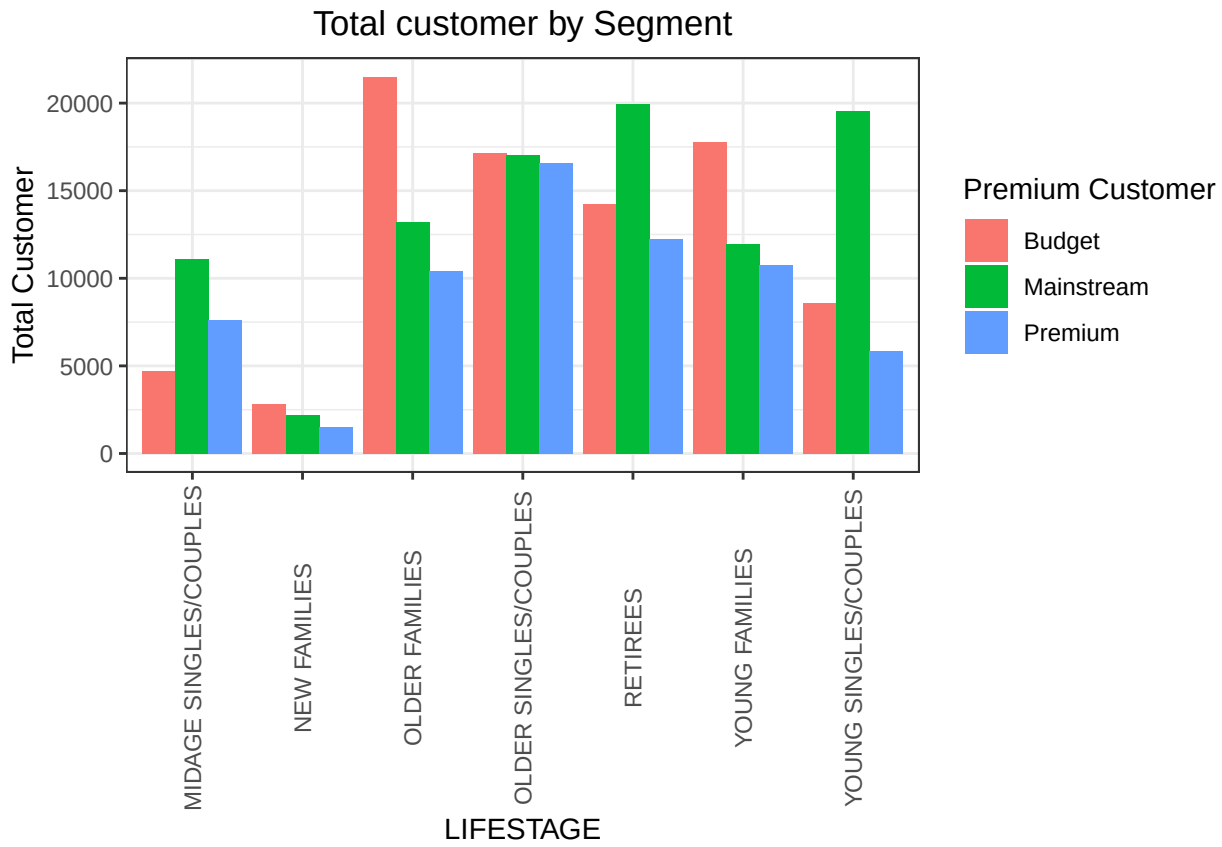
Number of customers by LIFESTAGE and PREMIUM_CUSTOMER

```
Total_customers <- data %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER)%>%
  summarise(Customer = n())
```

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

##Plot graph

```
ggplot(Total_customers, aes(x = LIFESTAGE, y = Customer, fill = PREMIUM_CUSTOMER)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Total customer by Segment", x = "LIFESTAGE", y = "Total Customer", fill = "Premium Custo
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



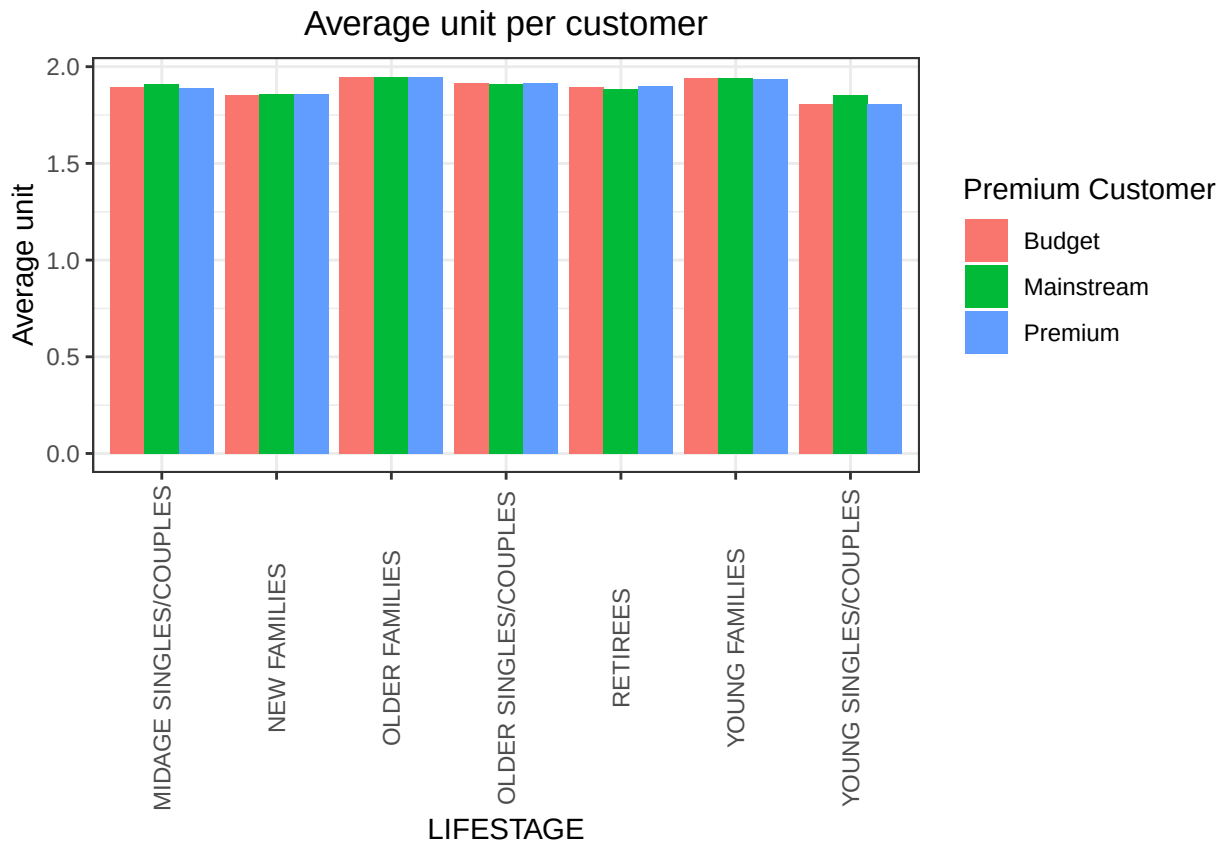
Average number of units per customer by LIFESTAGE and PREMIUM_CUSTOMER

```
Average_unit <- data %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER)%>%
  summarise(Average_unit = mean(PROD_QTY))
```

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

Plot a graph to show the trend

```
ggplot(Average_unit, aes(x = LIFESTAGE, y = Average_unit, fill = PREMIUM_CUSTOMER)) +
 geom_bar(stat = "identity", position = "dodge") +
 labs(title = "Average unit per customer", x = "LIFESTAGE", y = "Average unit", fill = "Premium Custome
 theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



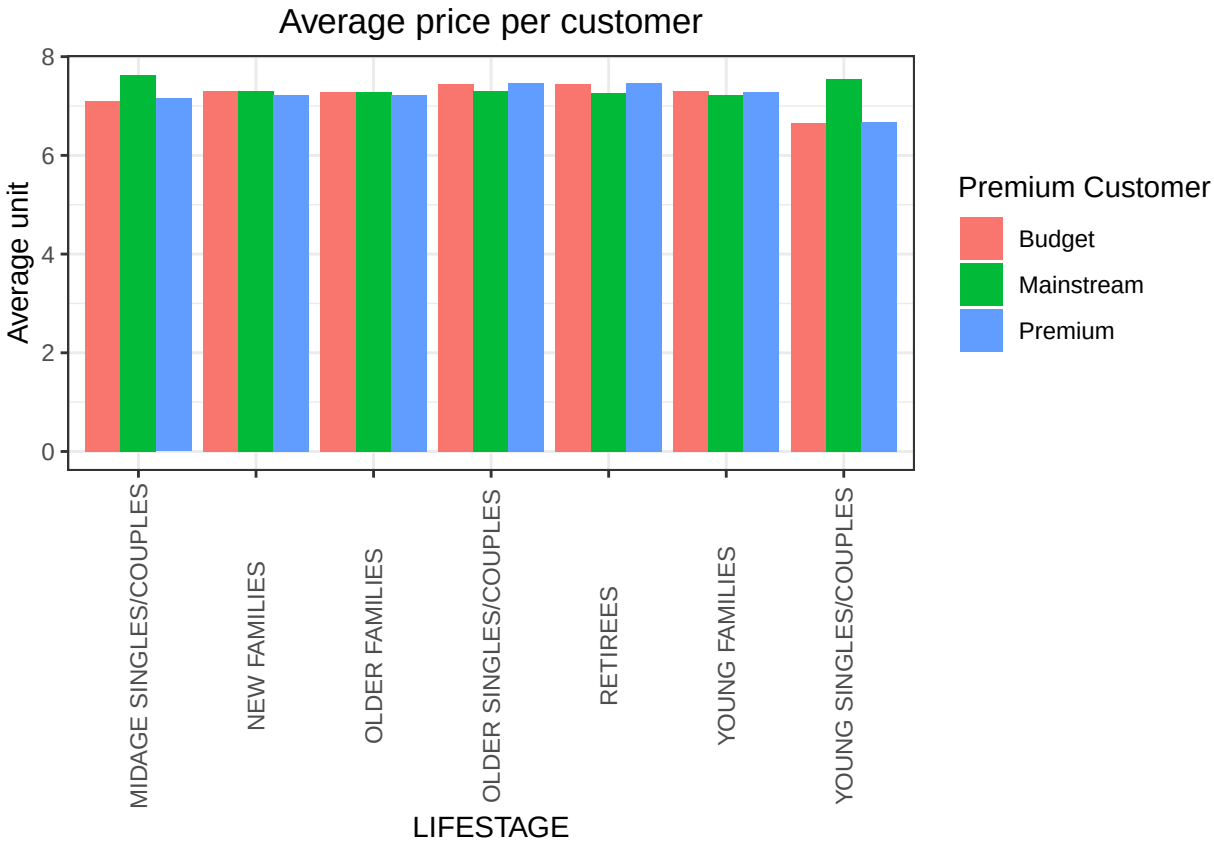
Average price per unit

```
Average_price <- data %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER)%>%
  summarise(Average_unit = mean(TOT_SALES))
```

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

Plot a graph to show the trend

```
ggplot(Average_price, aes(x = LIFESTAGE, y = Average_unit, fill = PREMIUM_CUSTOMER)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average price per customer", x = "LIFESTAGE", y = "Average unit", fill = "Premium Custo
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```


Average price per customer
Average unit
8
6
4
2
0
MIDAGE SINGLES/COUPLES
NEW FAMILIES
OLDER FAMILIES
OLDER SINGLES/COUPLES
RETIREES
YOUNG FAMILIES
YOUNG SINGLES/COUPLES
LIFESTAGE
Premium Customer
Budget
Mainstream
Premium