

QuantumDATask2

Santosh Reddy Edulapalle

2023-03-27

Load required libraries and datasets

```
filePath <- "/Users/santosh/Documents/QuantumDA/quantumDA/"
df <- fread(paste0(filePath,"QVI_data.csv"))
#### Set themes for plots
theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = 0.5))
```

```
head(df)
```

```
##      LYLTY_CARD_NBR      DATE STORE_NBR TXN_ID PROD_NBR
## 1:      1000 2018-10-17      1      1      5
## 2:      1002 2018-09-16      1      2     58
## 3:      1003 2019-03-07      1      3     52
## 4:      1003 2019-03-08      1      4    106
## 5:      1004 2018-11-02      1      5     96
## 6:      1005 2018-12-28      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: Grain Waves Sour      Cream&Chives 210G      1      3.6      210
## 4: Natural ChipCo      Hony Soy Chckn175g      1      3.0      175
## 5:      WW Original Stacked Chips 160g      1      1.9      160
## 6:      Cheetos Puffs 165g      1      2.8      165
##
##      BRAND      LIFESTAGE PREMIUM_CUSTOMER
## 1: NATURAL YOUNG SINGLES/COUPLES      Premium
## 2: RRD YOUNG SINGLES/COUPLES      Mainstream
## 3: GRNWVES YOUNG FAMILIES      Budget
## 4: NATURAL YOUNG FAMILIES      Budget
## 5: WOOLWORTHS OLDER SINGLES/COUPLES      Mainstream
## 6: CHEETOS MIDAGE SINGLES/COUPLES      Mainstream
```

Select control stores

The client has selected store numbers 77, 86 and 88 as trial stores and want control stores to be established stores that are operational for the entire observation period. We would want to match trial stores to control stores that are similar to the trial store prior to the trial period of Feb 2019 in terms of : - Monthly overall sales revenue - Monthly number of customers - Monthly number of transactions per customer Let's first create the metrics of interest and filter to stores that are present throughout the pre-trial period.

```
#### Calculate these measures over time for each store
#### Add a new month ID column in the data with the format yyyyymm.
```

```
df <- df[, YEARMONTH := (format(as.Date(df$DATE,format = "%Y/%m/%d"),paste("%Y","%m",sep = ""))) ]

#### Next, we define the measure calculations to use during the analysis.

# Over to you! For each store and month calculate total sales, number of customers, transactions per cus
## Hint: you can use uniqueN() to count distinct values in a column
measureOverTime <- df[, .(totSales = sum(TOT_SALES),
                          nCustomers = uniqueN(LYLTY_CARD_NBR),
                          nTxnPerCust = length(unique(TXN_ID))/uniqueN(LYLTY_CARD_NBR),
                          nChipsPerTxn = sum(PROD_QTY)/length(unique(TXN_ID)),
                          avgPricePerUnit = (sum(TOT_SALES)/sum(PROD_QTY))
                        ),
                      by = .(STORE_NBR, YEARMONTH) ][order(STORE_NBR, YEARMONTH)]

#### Filter to the pre-trial period and stores with full observation periods
storesWithFullObs <- unique(measureOverTime[, .N, STORE_NBR][N == 12, STORE_NBR])
preTrialMeasures <- measureOverTime[YEARMONTH < 201902 & STORE_NBR %in% storesWithFullObs, ]
```

Now we need to work out a way of ranking how similar each potential control store is to the trial store. We can calculate how correlated the performance of each store is to the trial store. Let's write a function for this so that we don't have to calculate this for each trial store and control store pair.

Over to you! Create a function to calculate correlation for a measure, looping through each contro
 #### Let's define inputTable as a metric table with potential comparison stores, metricCol as the store

```
calculateCorrelation <- function(inputTable, metricCol, storeComparison) {
  calcCorrTable = data.table(Store1 = numeric(), Store2 = numeric(), corr_measure = numeric())
  storeNumbers <-

  for (i in storeNumbers) {
    calculatedMeasure = data.table("Store1" = , "Store2" = , "corr_measure" = )

    calcCorrTable <- rbind(calcCorrTable, calculatedMeasure)
  }
  return(calcCorrTable)
}
```