

Quantium Virtual Internship- Retail Strategy and Analytics- Task 2

2025-12-04

```
# Load required Libraries
library(data.table)

## Warning: package 'data.table' was built under R version 4.4.3

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.4.2

library(lubridate)

## Warning: package 'lubridate' was built under R version 4.4.2

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':
##
##      hour, isoweek, mday, minute, month, quarter, second, wday, week,
##      yday, year

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

# Load dataset
data <- fread("QVI_data.csv")

# Create Month ID
data[, YEARMONTH := year(Date)*100 + month(Date)]

# Calculate measures for each store
measureOverTime <- data[, .(
  totSales = sum(TOT_SALES),
  nCustomers = uniqueN(LYLT_CARD_NBR),
  nTxnPerCust = sum(TOT_SALES)/uniqueN(LYLT_CARD_NBR),
  avgPricePerUnit = sum(TOT_SALES)/sum(PROD_QTY)
), by = .(STORE_NBR, YEARMONTH)][order(STORE_NBR, YEARMONTH)]

# Filter stores with full pre-trial period (Jan 2018 to Jan 2019)
storesWithFullObs <- measureOverTime[, .N, by = STORE_NBR][N == 12,
STORE_NBR]
preTrialMeasures <- measureOverTime[YEARMONTH < 201902 & STORE_NBR %in%
storesWithFullObs]
```

```

# Function to calculate correlation between trial store and potential
controls
calculateCorrelation <- function(inputTable, metricCol, storeComparison){
  storeNumbers <- unique(inputTable$STORE_NBR)
  calcCorrTable <- rbindlist(lapply(storeNumbers, function(i){
    data.table(
      Store1 = storeComparison,
      Store2 = i,
      corr_measure = cor(inputTable[STORE_NBR == storeComparison,
eval(metricCol)],
                        inputTable[STORE_NBR == i, eval(metricCol)])
    )
  })))
  return(calcCorrTable)
}

# Function to calculate magnitude distance
calculateMagnitudeDistance <- function(inputTable, metricCol,
storeComparison){
  storeNumbers <- unique(inputTable$STORE_NBR)
  calcDistTable <- rbindlist(lapply(storeNumbers, function(i){
    tmp <- data.table(
      Store1 = storeComparison,
      Store2 = i,
      YEARMONTH = inputTable[STORE_NBR == storeComparison, YEARMONTH],
      measure = abs(inputTable[STORE_NBR == storeComparison, eval(metricCol)]
-
                        inputTable[STORE_NBR == i, eval(metricCol)])
    )
    tmp
  })))

  # Standardize magnitude between 0 and 1
  minMaxDist <- calcDistTable[, .(minDist = min(measure), maxDist =
max(measure)), by = .(Store1, YEARMONTH)]
  distTable <- merge(calcDistTable, minMaxDist, by = c("Store1",
"YEARMONTH"))
  distTable[, magnitudeMeasure := 1 - (measure - minDist)/(maxDist -
minDist)]

  finalDistTable <- distTable[, .(mag_measure = mean(magnitudeMeasure)), by =
.(Store1, Store2)]
  return(finalDistTable)
}

# Function to select control store for a given trial store
selectControlStore <- function(trial_store, preTrialMeasures){
  # Correlation

```

```

    corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales),
trial_store)
    corr_nCustomers <- calculateCorrelation(preTrialMeasures,
quote(nCustomers), trial_store)

    # Magnitude
    magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures,
quote(totSales), trial_store)
    magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures,
quote(nCustomers), trial_store)

    # Combine scores
    corr_weight <- 0.5
    score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1",
"Store2"))
    score_nSales[, scoreNSales := corr_measure * corr_weight + mag_measure * (1
- corr_weight)]

    score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by =
c("Store1", "Store2"))
    score_nCustomers[, scoreNCust := corr_measure * corr_weight + mag_measure *
(1 - corr_weight)]

    score_Control <- merge(score_nSales, score_nCustomers, by = c("Store1",
"Store2"))
    score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]

    # Select control store (second highest score, not itself)
    control_store <- score_Control[Store1 == trial_store][order(-
finalControlScore)][2, Store2]
    return(control_store)
}

# Example: select control stores for trial stores 77, 86, 88
trial_stores <- c(77, 86, 88)
control_stores <- sapply(trial_stores, selectControlStore, preTrialMeasures =
preTrialMeasures)
control_stores
## [1] 233 155 237

# Visualization of sales trends (example for trial store 77)
trial_store <- 77
control_store <- control_stores["77"]

measureOverTimeSales <- measureOverTime
measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store,
"Trial",
                                     ifelse(STORE_NBR ==
control_store, "Control", "Other stores"))]

```

```
measureOverTimeSales[, TransactionMonth := as.Date(paste(YEARMONTH %% 100,
YEARMONTH %% 100, 1, sep = "-"))]
```

```
ggplot(measureOverTimeSales[YEARMONTH < 201903], aes(TransactionMonth,
totSales, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Total sales", title = paste("Total
sales by month - Trial store", trial_store))
```

