Retail Analytics and Strategy Task 2

Krithika HJ

18 July 2021

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Load libraries and datasets

```
library(readxl)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

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

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

Let us use the data set from task 1.

```
data <- fread(paste0("QVI_data.csv"))
```

Set themes for plots

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

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

Create a month ID

```
 # !diagnostics off

data[, YEARMONTH := year(DATE)*100 + month(DATE)]
glimpse(data)
```

```
## Rows: 264,836
## Columns: 11
## $ LYLTY_CARD_NBR   <int> 1000, 1002, 1003, 1003, 1004, 1005, 1007, 1007, 1009,~
## $ LIFESTAGE        <chr> "YOUNG SINGLES/COUPLES", "YOUNG SINGLES/COUPLES", "YO~
## $ PREMIUM_CUSTOMER <chr> "Premium", "Mainstream", "Budget", "Budget", "Mainstr~
## $ DATE             <date> 2018-10-17, 2018-09-16, 2019-03-08, 2019-03-07, 2018~
## $ STORE_NBR        <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ TXN_ID           <int> 1, 2, 4, 3, 5, 6, 7, 8, 9, 11, 10, 14, 15, 13, 12, 16~
## $ PROD_NBR         <int> 5, 58, 106, 52, 96, 86, 49, 10, 20, 59, 51, 49, 1, 59~
## $ PROD_NAME        <chr> "Natural Chip        Compny SeaSalt175g", "Red Rock D~
## $ PROD_QTY         <int> 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 2,~
## $ TOT_SALES        <dbl> 6.0, 2.7, 3.0, 3.6, 1.9, 2.8, 3.8, 2.7, 5.7, 5.1, 8.8~
## $ YEARMONTH        <dbl> 201810, 201809, 201903, 201903, 201811, 201812, 20181~
```

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

```
measureOverTime <- data[, .(totSales = sum(TOT_SALES),
nCustomers = uniqueN(LYLTY_CARD_NBR),
nTxnPerCust = uniqueN(TXN_ID)/uniqueN(LYLTY_CARD_NBR),
nChipsPerTxn = sum(PROD_QTY)/uniqueN(TXN_ID),
avgPricePerUnit = sum(TOT_SALES)/sum(PROD_QTY))
, by = c("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. Function:

```
calculateCorrelation <- function(inputTable, metricCol, storeComparison) {
  calcCorrTable = data.table(Store1 = numeric(), Store2 = numeric(), corr_measure = numeric())
storeNumbers <- unique(inputTable[, STORE_NBR])
for (i in storeNumbers) {
  calculatedMeasure = data.table("Store1" = storeComparison, "Store2" = i,
                                 "corr_measure" = cor(inputTable[STORE_NBR == storeComparison,
                                                                 eval(metricCol)], inputTable[ST
ORE_NBR == i,
                                                                                              ev
al(metricCol)]))
  calcCorrTable <- rbind(calcCorrTable, calculatedMeasure)
}
return(calcCorrTable)
}
```

Apart from correlation, we can also calculate a standardised metric based on the absolute difference between the trial store's performance and each control store's performance. Function:

```
#### Create a function to calculate a standardised magnitude distance for a measure, looping thr
ough each control store
calculateMagnitudeDistance <- function(inputTable, metricCol, storeComparison)
{
  calcDistTable = data.table(Store1 = numeric(), Store2 = numeric(), YEARMONTH = numeric(), meas
ure = numeric())
  storeNumbers <- unique(inputTable[, STORE_NBR])
  for (i in storeNumbers) {
    calculatedMeasure = data.table("Store1" = storeComparison,
                                   "Store2" = i,
                                   "YEARMONTH" = inputTable[STORE_NBR ==storeComparison, YEARMON
TH],
                                   "measure" = abs(inputTable[STORE_NBR == storeComparison, eval
(metricCol)]-inputTable[STORE_NBR == i, eval(metricCol)])
                                   )
    calcDistTable <- rbind(calcDistTable, calculatedMeasure)
  }

#### Standardise the magnitude distance so that the measure ranges from 0 to 1
  minMaxDist <- calcDistTable[, .(minDist = min(measure), maxDist = max(measure)), by = c("Store
1", "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)
}
```

Now let's use the functions to find the control stores! We'll select control stores based on how similar monthly total sales in dollar amounts and monthly number of customers are to the trial stores. So we will need to use our functions to get four scores, two for each of total sales and total customers.

```
trial_store <- 77
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales),trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers), trial_store)

magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures,quote(totSales), trial_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures,quote(nCustomers), trial_sto
re)
```

We'll need to combine the all the scores calculated using our function to create a composite score to rank on. Let's take a simple average of the correlation and magnitude scores for each driver.

```
#### Create a combined score composed of correlation and magnitude
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1", "Store2"))[, scoreNSales :
= corr_measure * corr_weight + mag_measure * (1-corr_weight)]

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

Now we have a score for each of total number of sales and number of customers. Let's combine the two via a simple average.

```
#### Combine scores across the drivers
score_Control <- merge(score_nSales, score_nCustomers, by = c("Store1","Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]
```

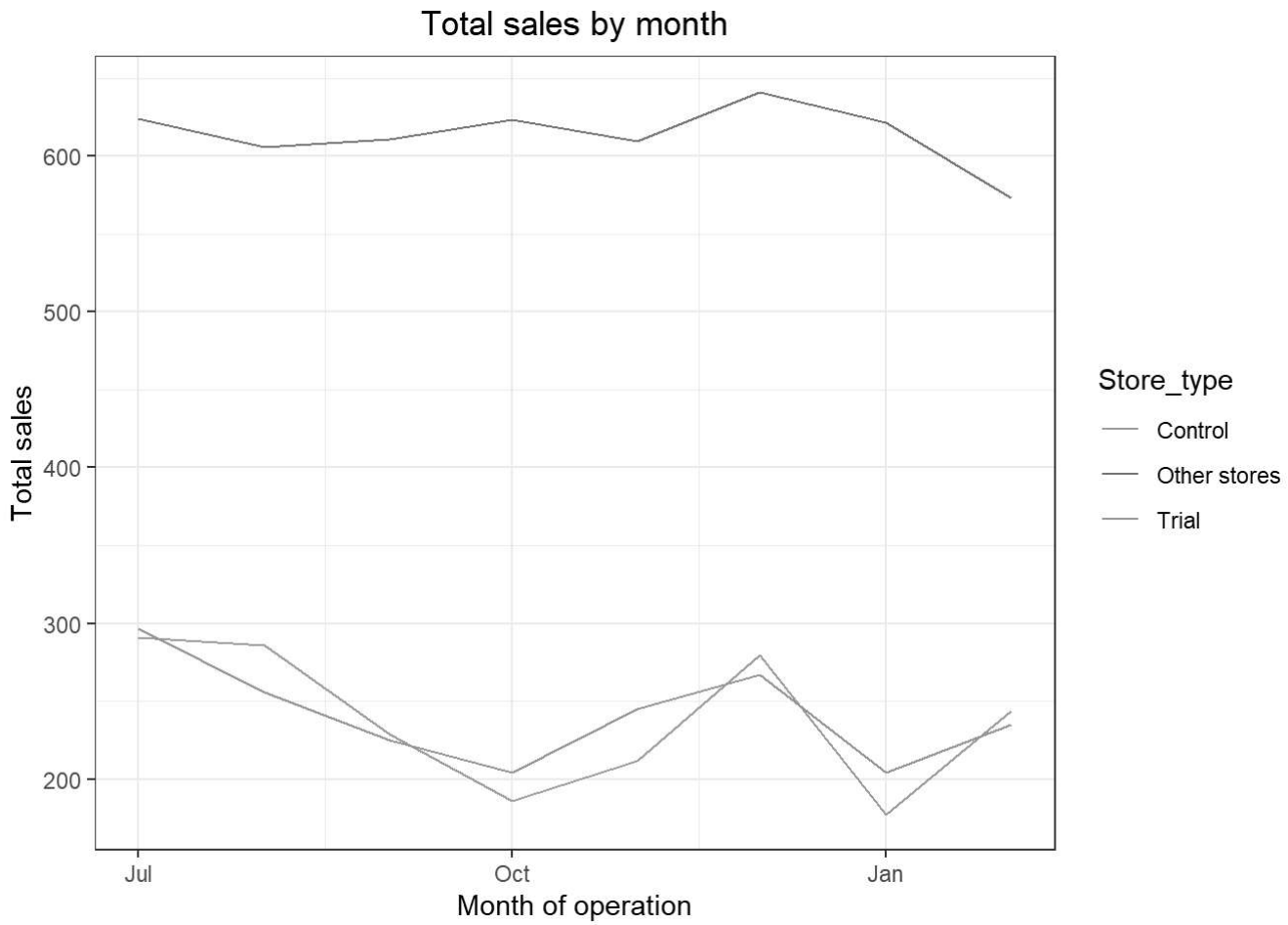
The store with the highest score is then selected as the control store since it is most similar to the trial store.

```
#### Select control stores based on the highest matching store (closest to 1 but not the store i
tself, i.e. the second ranked highest store)
#### Select control store for trial store 77
control_store <- score_Control[Store1 == trial_store,][order(-finalControlScore)][2, Store2]
control_store
```

```
## [1] 233
```

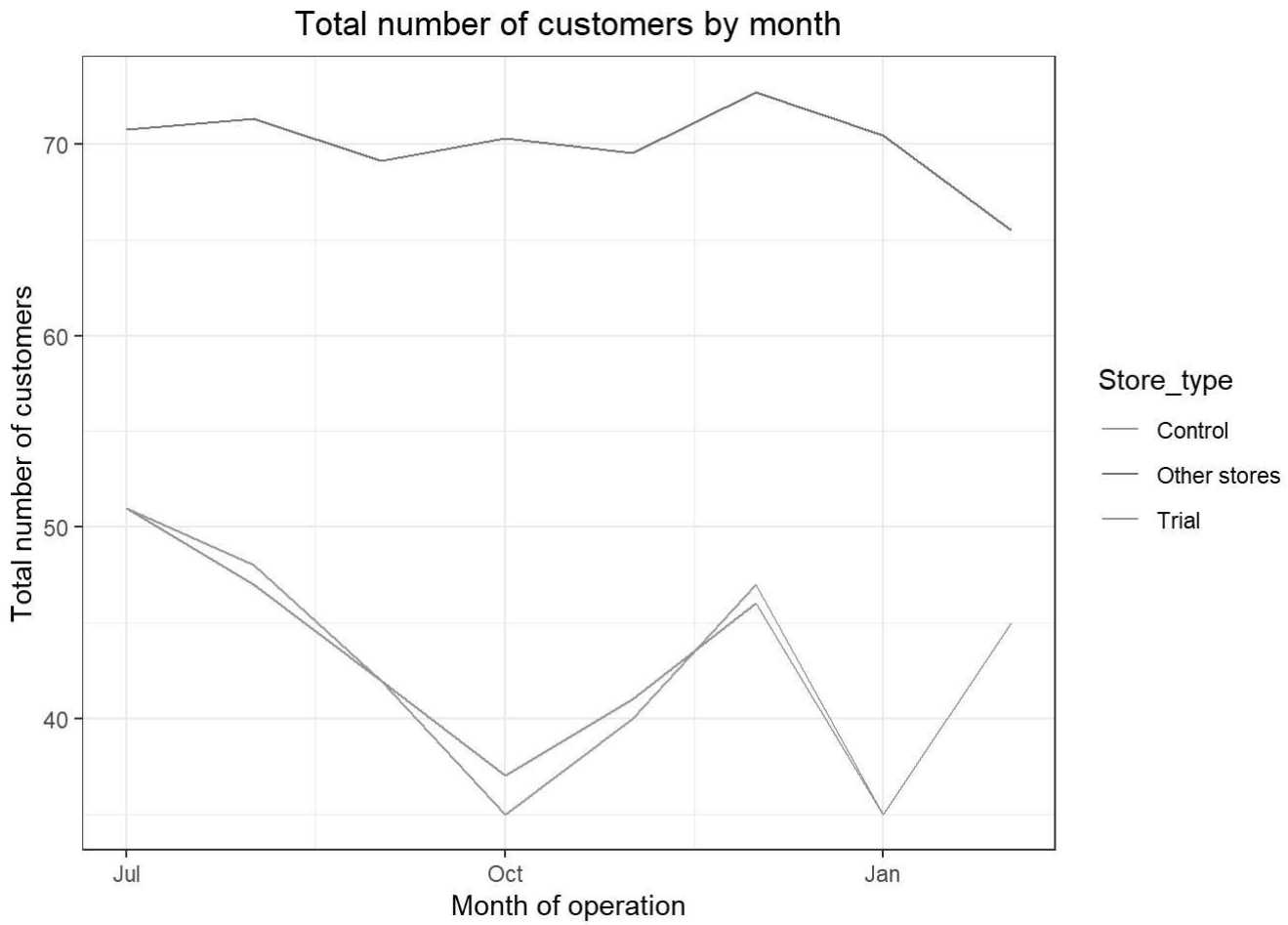
Now that we have found a control store, let's check visually if the drivers are indeed similar in the period before the trial. We'll look at total sales first.

```
#### Visual checks on trends based on the drivers
measureOverTimeSales <- measureOverTime

pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial", ifel
se(STORE_NBR == control_store, "Control", "Other stores"))][, totSales := mean(totSales), by = c
("YEARMONTH", "Store_type")][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %%
100, 1, sep = "-"), "%Y-%m-%d")][YEARMONTH < 201903, ]
ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
geom_line() +
labs(x = "Month of operation", y = "Total sales", title = "Total sales by month")
```



Next, number of customers.

```
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
 ifelse(STORE_NBR == control_store, "Control", "Other stores"))][, numberCustomers := mean(nCust
omers), by = c("YEARMONTH", "Store_type")][, TransactionMonth := as.Date(paste(YEARMONTH %/%100,
YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")][YEARMONTH < 201903 , ]
ggplot(pastCustomers, aes(TransactionMonth, numberCustomers, color =Store_type)) +geom_line() +
 labs(x = "Month of operation", y = "Total number of customers", title = "Total number of custom
ers by month")
```



Assessment of trial

The trial period goes from the start of March 2019 to June 2019. We now want to see if there has been an uplift in overall chip sales. We'll start with scaling the control store's sales to a level similar to control for any differences between the two stores outside of the trial period.

```
#### Scale pre-trial control sales to match pre-trial trial store sales
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902,
 sum(totSales)]/preTrialMeasures[STORE_NBR == control_store & YEARMONTH < 201902, sum(totSales)]

#### Apply the scaling factor
measureOverTimeSales <- measureOverTime
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][ ,controlSales := totSa
les * scalingFactorForControlSales]
```

Now that we have comparable sales figures for the control store, we can calculate the percentage difference between the scaled control sales and the trial store's sales during the trial period.

```
#### Calculate the percentage difference between scaled control sales and trial sales
percentageDiff <- merge(scaledControlSales[, c("YEARMONTH", "controlSales")], measureOverTime[ST
ORE_NBR == trial_store, c("totSales","YEARMONTH")], by = "YEARMONTH" )[, percentageDiff := abs(c
ontrolSales-totSales)/controlSales]
```

Let's see if the difference is significant!

```
#### As our null hypothesis is that the trial period is the same as the pre-trial period, let's
 take the standard deviation based on the scaled percentage difference in the pre-trial period

stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])

#### note that there are 8 months in the pre-trial period hence 8 - 1 = 7 degrees of freedom

degreesOfFreedom <- 7
#### We will test with a null hypothesis of there being 0 difference between trial and control s
tores
percentageDiff[, tValue := (percentageDiff - 0)/stdDev][, TransactionMonth := as.Date(paste(YEAR
MONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")][YEARMONTH < 201905 & YEARMONTH > 20
1901, .(TransactionMonth,tValue)]
```

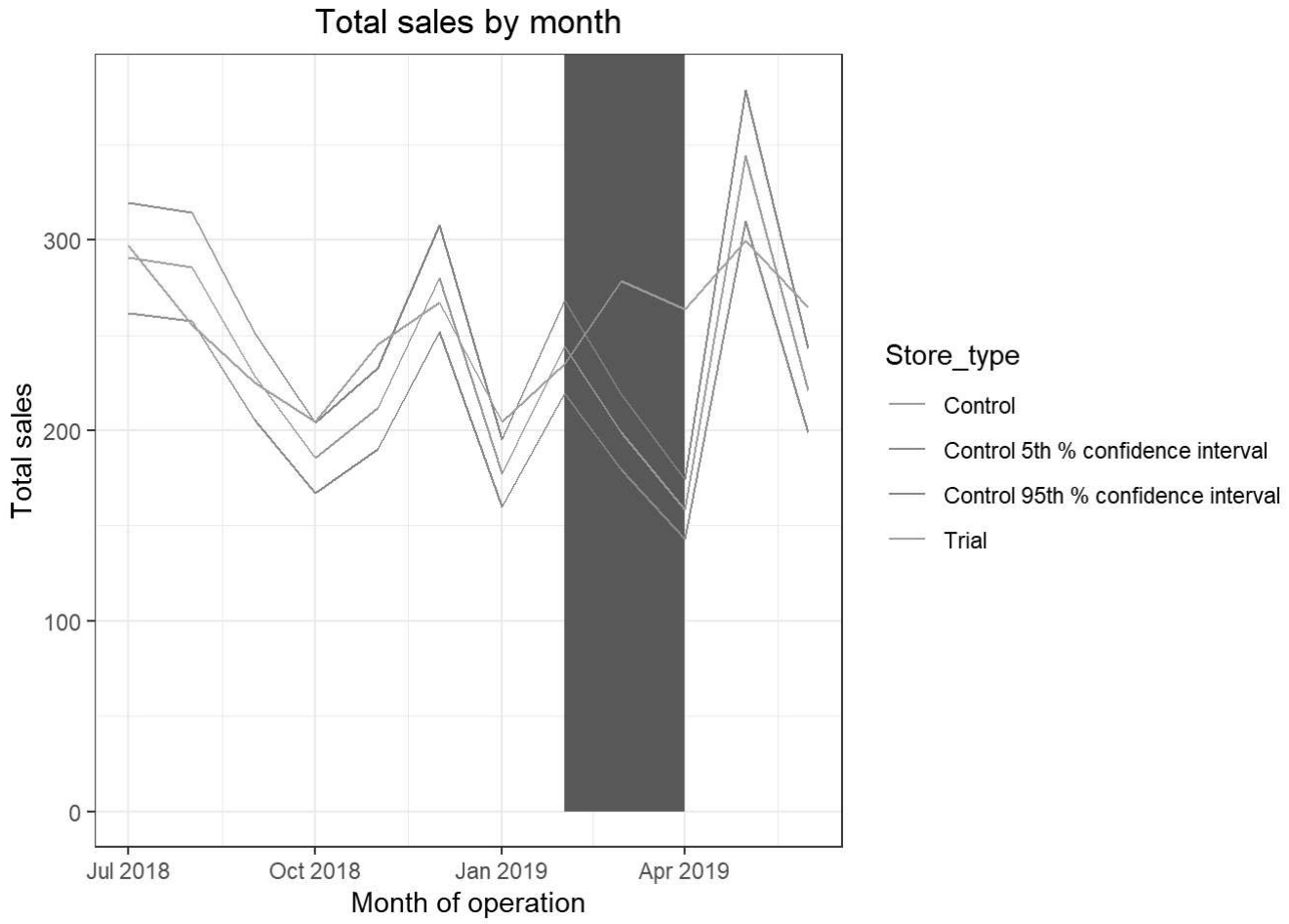
```
##    TransactionMonth    tValue
## 1:       2019-02-01  1.183534
## 2:       2019-03-01  7.339116
## 3:       2019-04-01 12.476373
```

```
#### Find the 95th percentile of the t distribution with the appropriate degrees of freedom to c
ompare against
qt(0.95, df = degreesOfFreedom)
```

```
## [1] 1.894579
```

We can observe that the t-value is much larger than the 95th percentile value of the t-distribution for March and April - i.e. the increase in sales in the trial store in March and April is statistically greater than in the control store. Let's create a more visual version of this by plotting the sales of the control store, the sales of the trial stores and the 95th percentile value of sales of the control store.

```
measureOverTimeSales <- measureOverTime
#### Trial and control store total sales
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",ifels
e(STORE_NBR == control_store, "Control", "Other stores"))][, totSales := mean(totSales), by = c(
"YEARMONTH","Store_type")][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 1
00, 1, sep = "-"), "%Y-%m-%d")][Store_type %in% c("Trial", "Control"), ]

#### Control store 95th percentile
pastSales_Controls95 <- pastSales[Store_type == "Control",][, totSales := totSales * (1 + stdDev
* 2)][, Store_type := "Control 95th % confidence interval"]

#### Control store 5th percentile
pastSales_Controls5 <- pastSales[Store_type == "Control", ][, totSales := totSales * (1 - stdDev
* 2)][, Store_type := "Control 5th % confidence interval"]

trialAssessment <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)

#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, totSales, color = Store_type)) + geom_rect(data =
 trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,], aes(xmin = min(TransactionMonth),
 xmax = max(TransactionMonth), ymin = 0 , ymax = Inf, color = NULL), show.legend = FALSE) + geom
_line() + labs(x = "Month of operation", y = "Total sales", title = "Total sales by month")
```

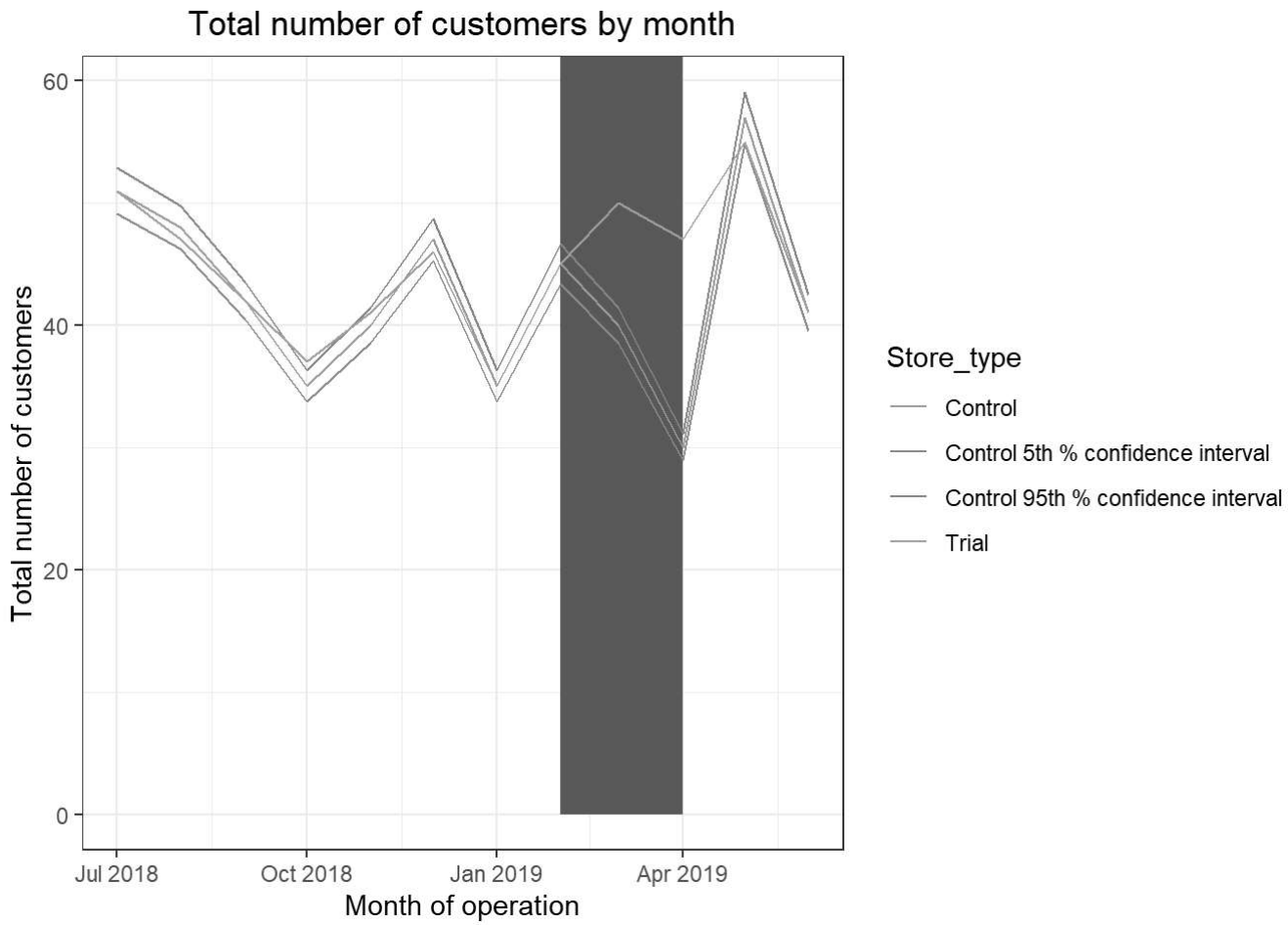


The results show that the trial in store 77 is significantly different to its control store in the trial period as the trial store performance lies outside the 5% to 95% confidence interval of the control store in two of the three trial months. Let's have a look at assessing this for number of customers as well.

```
#### This would be a repeat of the steps before for total sales
#### Scale pre-trial control customers to match pre-trial trial store customers
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, s
um(nCustomers)]/preTrialMeasures[STORE_NBR == control_store & YEARMONTH < 201902, sum(nCustomer
s)]

#### Apply the scaling factor
measureOverTimeCusts <- measureOverTime
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store,][ , controlCustomers
 := nCustomers * scalingFactorForControlCust][, Store_type := ifelse(STORE_NBR == trial_store,
"Trial", ifelse(STORE_NBR == control_store, "Control", "Other stores"))]
#### Calculate the percentage difference between scaled control sales and trial sales
percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH","controlCustomers")], measureOver
TimeCusts[STORE_NBR == trial_store, c("nCustomers", "YEARMONTH")], by = "YEARMONTH")[, percentag
eDiff := abs(controlCustomers-nCustomers)/controlCustomers]
```

Let's again see if the difference is significant visually!

```
#### As our null hypothesis is that the trial period is the same as the pre-trial period, let's
 take the standard deviation based on the scaled percentage difference in the pre-trial period

stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7

#### Trial and control store number of customers
pastCustomers <- measureOverTimeCusts[, nCusts := mean(nCustomers), by = c("YEARMONTH", "Store_t
ype")][Store_type %in% c("Trial", "Control"), ]

#### Control store 95th percentile
pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",][, nCusts := nCusts * (1 + st
dDev * 2)][, Store_type := "Control 95th % confidence interval"]
#### Control store 5th percentile
pastCustomers_Controls5 <- pastCustomers[Store_type == "Control",][, nCusts := nCusts * (1 - std
Dev * 2)][, Store_type := "Control 5th % confidence interval"]

trialAssessment <- rbind(pastCustomers, pastCustomers_Controls95, pastCustomers_Controls5)

#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, nCusts, color = Store_type)) + geom_rect(data = tr
ialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,], aes(xmin = min(TransactionMonth), xma
x = max(TransactionMonth), ymin = 0 , ymax = Inf, color = NULL), show.legend = FALSE) + geom_lin
e() +
labs(x = "Month of operation", y = "Total number of customers", title = "Total number of custome
rs by month")
```


Total number of customers by month
Total number of customers
60
40
20
0
Jul 2018
Oct 2018
Jan 2019
Apr 2019
Month of operation
Store_type
Control
Control 5th % confidence interval
Control 95th % confidence interval
Trial