Quantium Virtual Internship - Retail Strategy and Analytics - Task 2

Lizbeth Santiago

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

Load required libraries

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

Read in data from previous Task

```
filePath <- "C:/Users/Liz/Documents/Virtual_Internship/"
data <- fread(paste0(filePath,"QVI_data.csv"))
```

Set themes for plots

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

Select control stores

Calculate measures over time for each store

Add a new month ID column in the data with the format yyyymm

```
data[, YEARMONTH := format(DATE,'%Y%m')]
```

Measure calculations for each store and month

```
measureOverTime <- data[, .(totSales = sum(TOT_SALES),
                            nCustomers = uniqueN(LYLTY_CARD_NBR),
                            nTxnPerCust = .N/uniqueN(LYLTY_CARD_NBR),
                            nChipsPerTxn = sum(PROD_QTY)/uniqueN(LYLTY_CARD_NBR),
                            avgPricePerUnit = sum(TOT_SALES)/sum(PROD_QTY))
                        , .(YEARMONTH , STORE_NBR)][order(YEARMONTH)]
```

Filter to the stores with full observation periods and the pre-trial period

```
storesWithFullObs <- unique(measureOverTime[, .N, STORE_NBR][N == 12, STORE_NBR])
preTrialMeasures <- measureOverTime[YEARMONTH < 201902 & STORE_NBR %in% storesWithFullObs, ]
```

Functions

Function to calculate correlation

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.

```
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 == i, ..metricCol],
                                                          inputTable[STORE_NBR == storeComparison, ..me
      )
      calcCorrTable <- rbind(calcCorrTable, calculatedMeasure, use.names = FALSE)
    }
  return(calcCorrTable)
}
```

Function to calculate a standardised magnitude distance for a measure

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.

```
calculateMagnitudeDistance <- function(inputTable, metricCol, storeComparison) {

  calcDistTable = data.table(Store1 = numeric(), Store2 = numeric(), YEARMONTH = numeric(), measure = nu
  storeNumbers <- unique(inputTable[, STORE_NBR])

  for (i in storeNumbers) {
    calculatedMeasure = data.table("Store1" = storeComparison
```

```
                                   , "Store2" = i
                                   , "YEARMONTH" = inputTable[STORE_NBR == storeComparison, YEARMONTH]
                                   , "measure" = abs(inputTable[STORE_NBR == storeComparison, eval(metri
                                                    - 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("Store1", "YEA
  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)
}
```

Store 77

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.

Calculate correlations against store 77 using total sales and number of customers.

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

Calculating magnitude.

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

Combine all the scores calculated to create a composite score to rank on.

```
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by = c('Store1','Store2'))[
                , scoreNSales := corr_weight*corr_measure + corr_weight*mag_measure]


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

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

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

Control store

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

```r
controlStores <- score_Control[finalControlScore > 0.85, .(Store2, finalControlScore)]
controlStores
```

```
##    Store2 finalControlScore
## 1:     17         0.8582899
## 2:     41         0.8918077
## 3:     77         1.0000000
## 4:    233         0.9680425
```

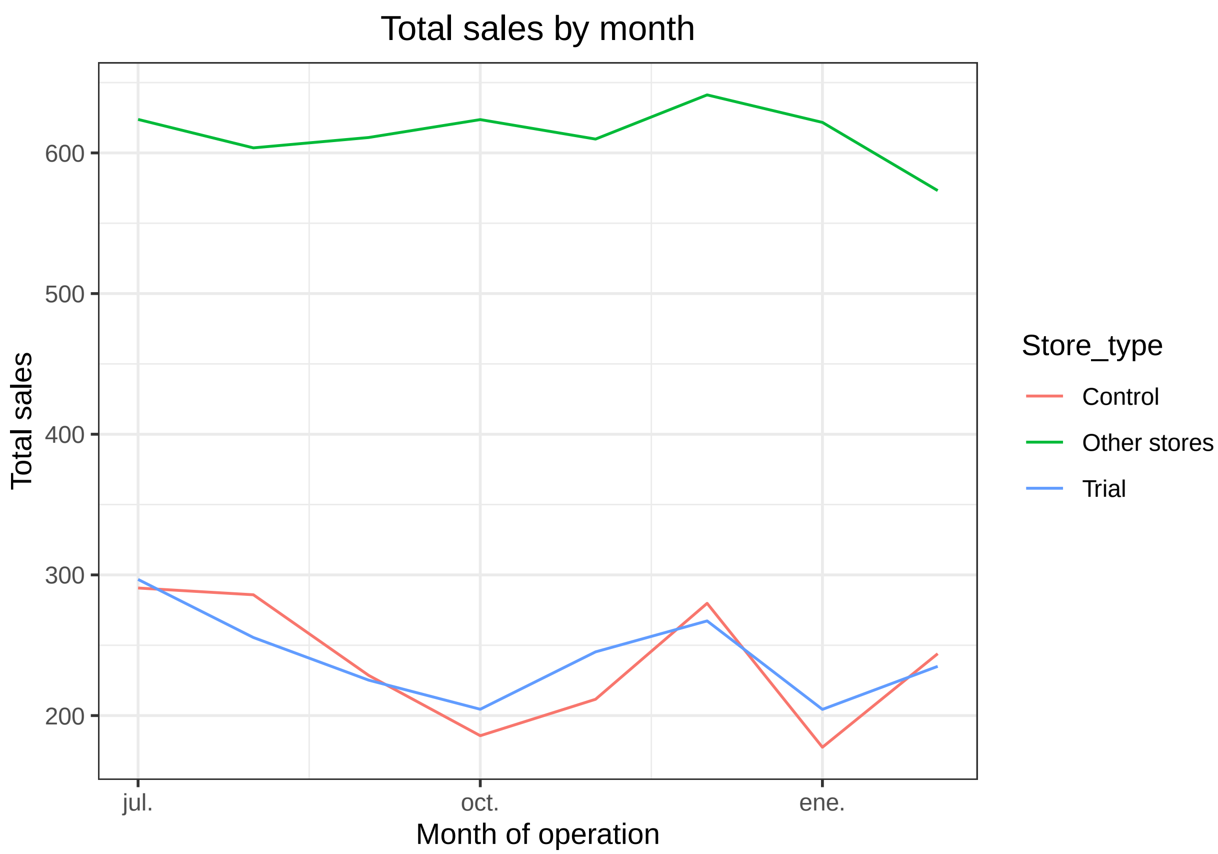
```r
control_store <- 233
```

Now that we have found a control store, let's check visually if the drivers are indeed similar in the period before the trial.

Visual checks on trends based on the drivers before de trial

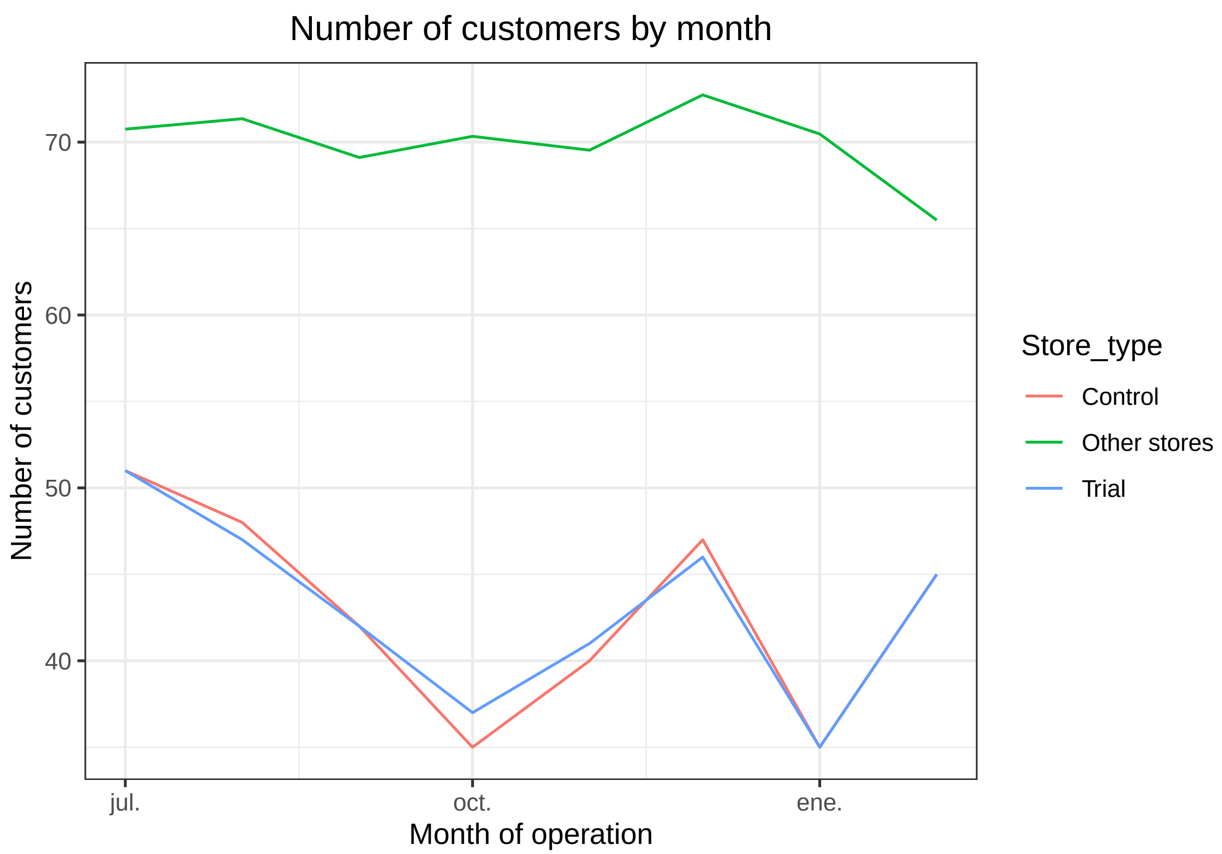
Total sales

```r
measureOverTimeSales <- copy(measureOverTime)
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                  ifelse(STORE_NBR == control_store, "Control", "Other 
  , totalSales := mean(totSales), by = c("YEARMONTH","Store_type")][
  , TransactionMonth := as.Date(paste(as.numeric(YEARMONTH)%/%100, as.numeric(YEARMONTH)%%100, 1, sep = 
    as.numeric(YEARMONTH) < 201903 , ]

ggplot(pastSales, aes(TransactionMonth, totalSales, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by month")
```



Number of customers.

```
measureOverTimeCusts <- copy(measureOverTime)
pastCustomers <- measureOverTimeCusts[,Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                      ifelse(STORE_NBR == control_store, "Control", "Othe
  , totalCustomers := mean(nCustomers), by = c('YEARMONTH','Store_type')][
  , TransactionMonth := as.Date(paste(as.numeric(YEARMONTH)%/%100, as.numeric(YEARMONTH)%%100, 1, sep =
      as.numeric(YEARMONTH) < 201903 , ]

ggplot(pastCustomers, aes(TransactionMonth, totalCustomers, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Number of customers", title = "Number of customers by month")
```



Assessment of trial store 77

The trial period goes from the start of February 2019 to April 2019.

Assessing for sales

We now want to see if there has been an uplift in overall chip sales.

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

```
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][ ,
                    controlSales := totSales * 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

```
trialSales <- measureOverTimeSales[STORE_NBR == trial_store,]

percentageDiff <- merge(scaledControlSales[,c('YEARMONTH', 'controlSales')],
                        trialSales[,c('YEARMONTH','totSales')],
                        by = 'YEARMONTH')[, percentageDiff := (abs(controlSales-totSales)/
                                          ((controlSales+totSales)/2))]
```

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 stores. ### Calculate the t-values for the trial months.

```
percentageDiff[, tValue := abs(controlSales-totSales)/stdDev][
    , TransactionMonth := as.Date(paste(as.numeric(YEARMONTH)%/%100, as.numeric(YEARMONTH)%%100, 1,
                                        sep = "-"),"%Y-%m-%d")]
percentil95 <-  qt(0.95, df = degreesOfFreedom)
```

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.

Trial and control store total sales

```
pSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                              ifelse(STORE_NBR == control_store, "Control", "Other stor
    , totalSales := mean(totSales), by = c("YEARMONTH","Store_type")][
    , TransactionMonth := as.Date(paste(as.numeric(YEARMONTH)%/%100, as.numeric(YEARMONTH)%%100, 1, sep
      Store_type %in% c("Trial", "Control"), ]
```

Control store 95th percentile

```
pSales_Controls95 <- pSales[Store_type == "Control",
][, totalSales := totalSales * (1 + stdDev * 2)
][, Store_type := "Control 95th % confidence
interval"]
```

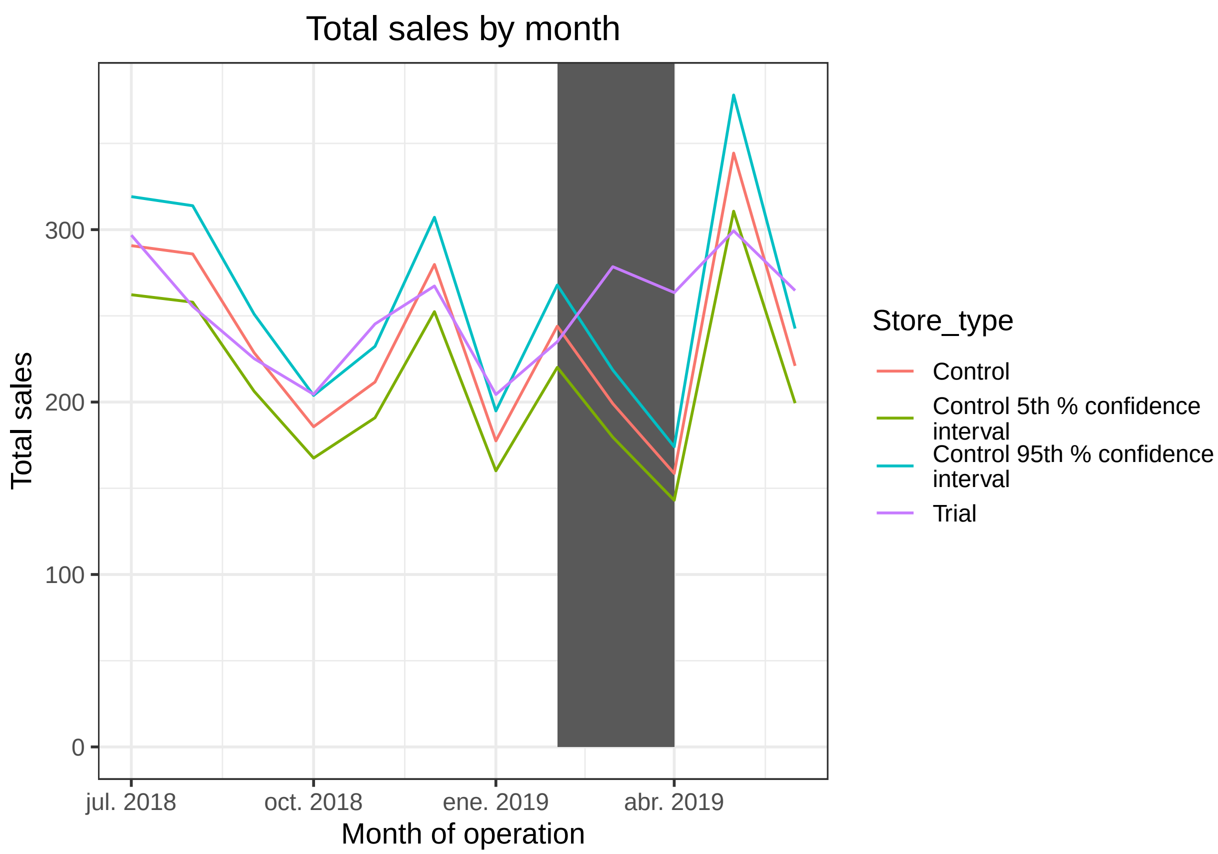
Control store 5th percentile

```r
pSales_Controls5 <- pSales[Store_type == "Control",
][, totalSales := totalSales * (1 - stdDev * 2)
][, Store_type := "Control 5th % confidence
interval"]
```

```r
trialAssessment <- rbind(pSales, pSales_Controls95, pSales_Controls5)
```

Plotting these in one nice graph

```r
ggplot(trialAssessment, aes(TransactionMonth, totalSales, 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")
```



Trial in store 77 is significantly different to its control store in the trial period in two of the three trial months.

Assessing for number of customers

Scaling Factor

```
scalingFactorForControlCustomers <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum
                                    preTrialMeasures[STORE_NBR == control_store & YEARMONTH < 201902, su
```

Apply the scaling factor

```
scaledControlCusts <- measureOverTimeCusts[STORE_NBR == control_store,
                                           ][ ,controlCustomers := nCustomers * scalingFactorForControl
```

Calculate the percentage difference between scaled control customers and trial customers

```
trialCusts <- measureOverTimeCusts[STORE_NBR == trial_store,]

percentageDiffCusts <- merge(scaledControlCusts[,c('YEARMONTH', 'controlCustomers')],
                        trialCusts[,c('YEARMONTH','totalCustomers')],
                        by = 'YEARMONTH')[, percentageDiff := (abs(controlCustomers-totalCustomers)/
                                                          ((controlCustomers+totalCustomers)/2))]
```

Let's see if the difference is significant

```
stdDevCust <- sd(percentageDiffCusts[YEARMONTH < 201902 , percentageDiff])
```

Calculate the t-values for the trial months.

```
percentageDiffCusts[, tValue := abs(controlCustomers-totalCustomers)/stdDevCust
                    ][, TransactionMonth := as.Date(paste(as.numeric(YEARMONTH)%/%100, as.numeric(YEARM
                                  sep = "-"),"%Y-%m-%d")]
```

Trial and control store number of customers

```
pCusts <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                ifelse(STORE_NBR == control_store, "Control", "Ot
       ][, totalCustomers2 := mean(totalCustomers), by = c("YEARMONTH","Store_type")
       ][, TransactionMonth := as.Date(paste(as.numeric(YEARMONTH)%/%100, as.numeric(YEARMONTH)%%100, 1
       ][Store_type %in% c("Trial", "Control"), ]
```

Control store 95th percentile

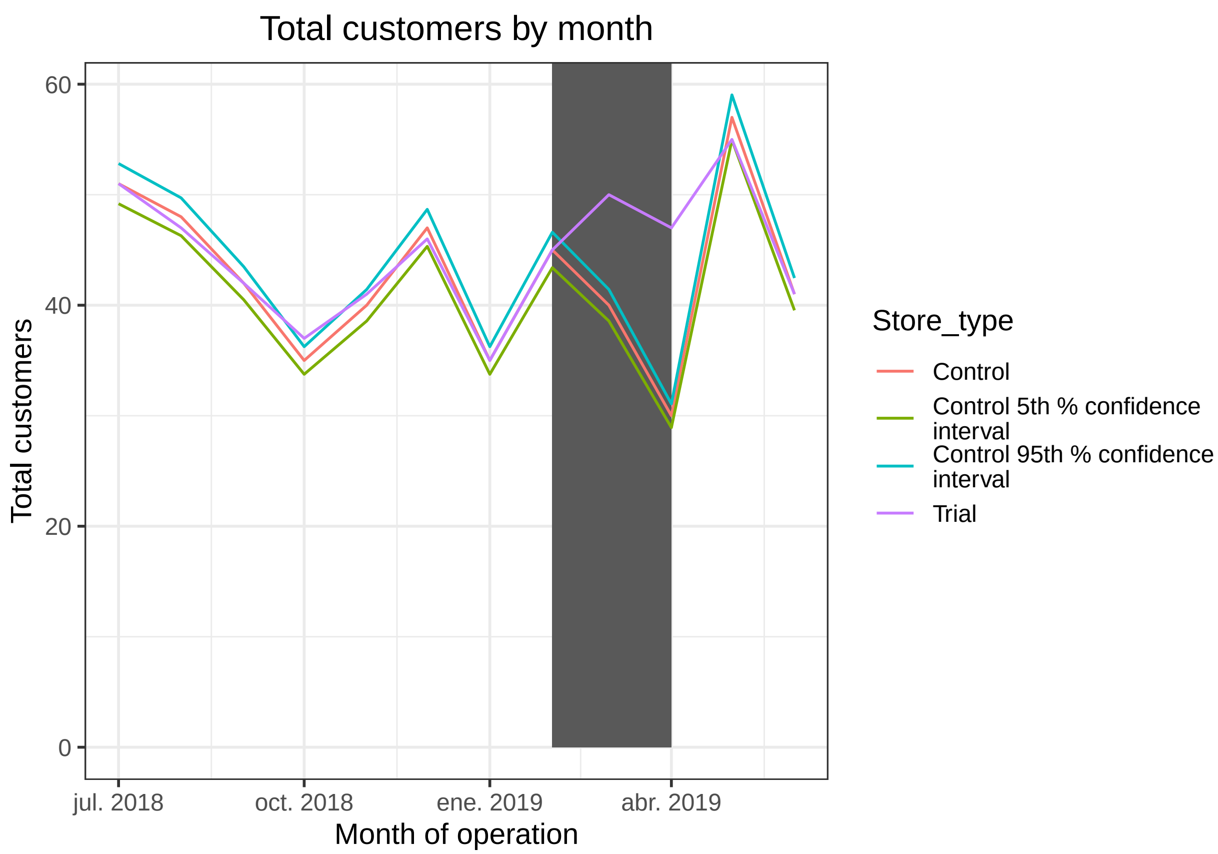
```
pCusts_Controls95 <- pCusts[Store_type == "Control",
][, totalCustomers2 := totalCustomers2 * (1 + stdDevCust * 2)
][, Store_type := "Control 95th % confidence
interval"]
```

Control store 5th percentile

```
pCusts_Controls5 <- pCusts[Store_type == "Control",
][, totalCustomers2 := totalCustomers2 * (1 - stdDevCust * 2)
][, Store_type := "Control 5th % confidence
interval"]

trialAssessmentCusts <- rbind(pCusts, pCusts_Controls95, pCusts_Controls5)
```

Plotting these in one nice graph

```
ggplot(trialAssessmentCusts, aes(TransactionMonth, totalCustomers2, color = Store_type)) +
  geom_rect(data = trialAssessmentCusts[ 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 customers", title = "Total customers by month")
```


Total customers by month
60
40
20
0
Total customers
jul. 2018
oct. 2018
ene. 2019
abr. 2019
Month of operation
Store_type
Control
Control 5th % confidence interval
Control 95th % confidence interval
Trial

Store 86

calculate correlations and magnitude against store 86 using total sales and number of customers.

```
trial_store <- 86
corr_nSales <- calculateCorrelation(preTrialMeasures, 'totSales', trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures, 'nCustomers', trial_store)

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

### combine all the scores calculated to create a composite score to rank on.
corr_weight <- 0.5

score_nSales <- merge(corr_nSales, magnitude_nSales, by = c('Store1','Store2'))[
  , scoreNSales := corr_weight*corr_measure + corr_weight*mag_measure]

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

### 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]
```

Control Store

```
controlStores <- score_Control[finalControlScore > 0.85, .(Store2, finalControlScore)]
controlStores
```

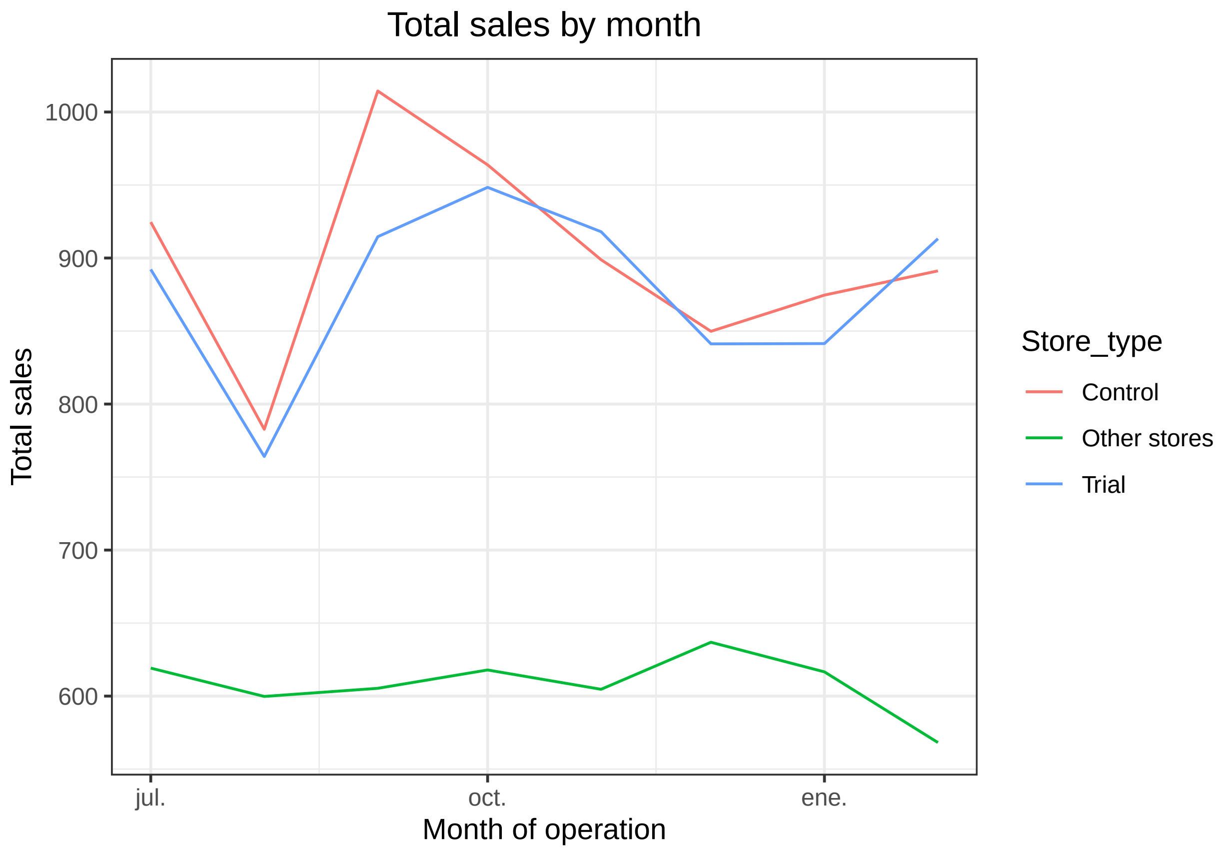
```
##    Store2 finalControlScore
## 1:     86         1.0000000
## 2:    109         0.8717505
## 3:    114         0.8611359
## 4:    155         0.9421896
```

```
control_store <- 155
```

Visual checks on trends based on the drivers

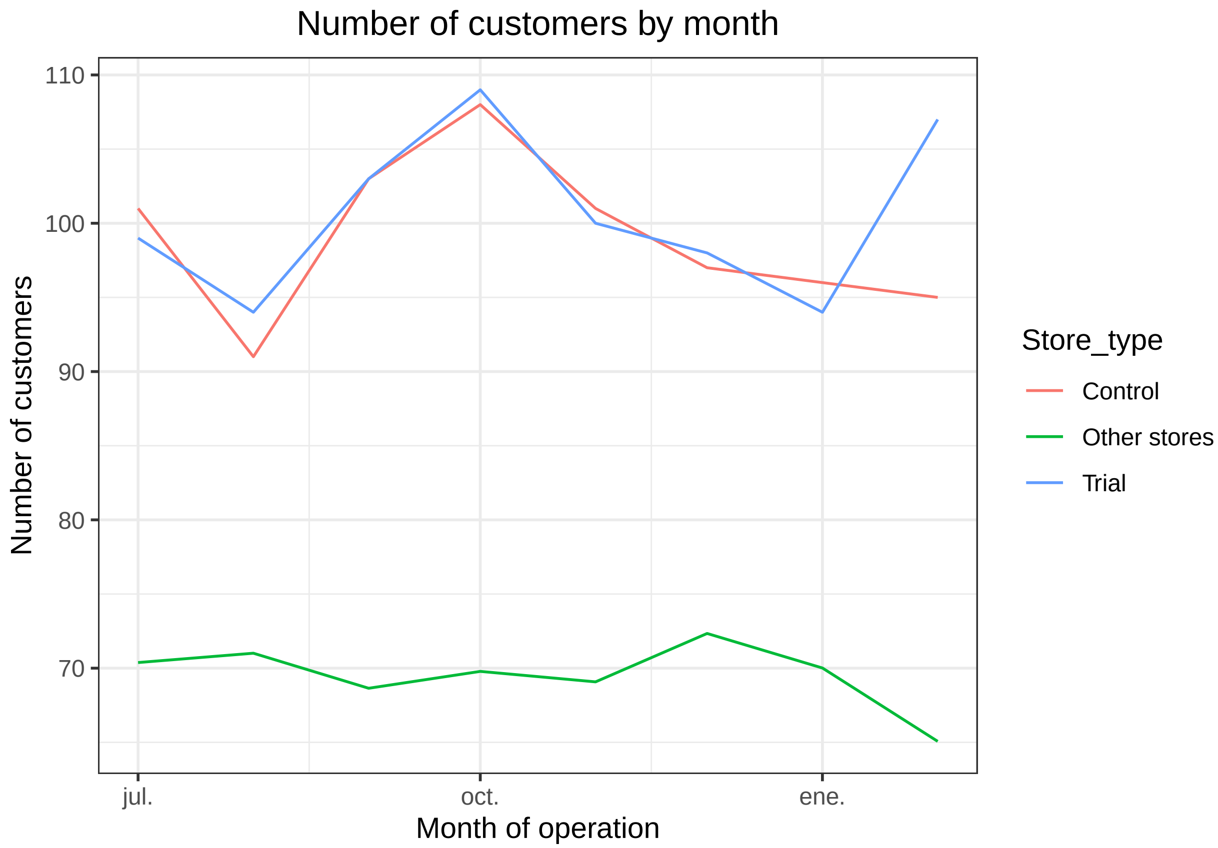
Sales

```
measureOverTimeSales <- copy(measureOverTime)
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                 ifelse(STORE_NBR == control_store, "Control",
                                                   , totalSales := mean(totSales), by = c("YEARM
                                                     , TransactionMonth := as.Date(paste(as.nume
                                                       as.numeric(YEARMONTH) < 201903 , ]

ggplot(pastSales, aes(TransactionMonth, totalSales, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by month")
```



Number of customers.

```
measureOverTimeCusts <- copy(measureOverTime)
pastCustomers <- measureOverTimeCusts[,Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                     ifelse(STORE_NBR == control_store, "Control
                                                       , totalCustomers := mean(nCustomers), by =
                                                         , TransactionMonth := as.Date(paste(as.n
                                                           as.numeric(YEARMONTH) < 201903 , ]

ggplot(pastCustomers, aes(TransactionMonth, totalCustomers, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Number of customers", title = "Number of customers by month")
```



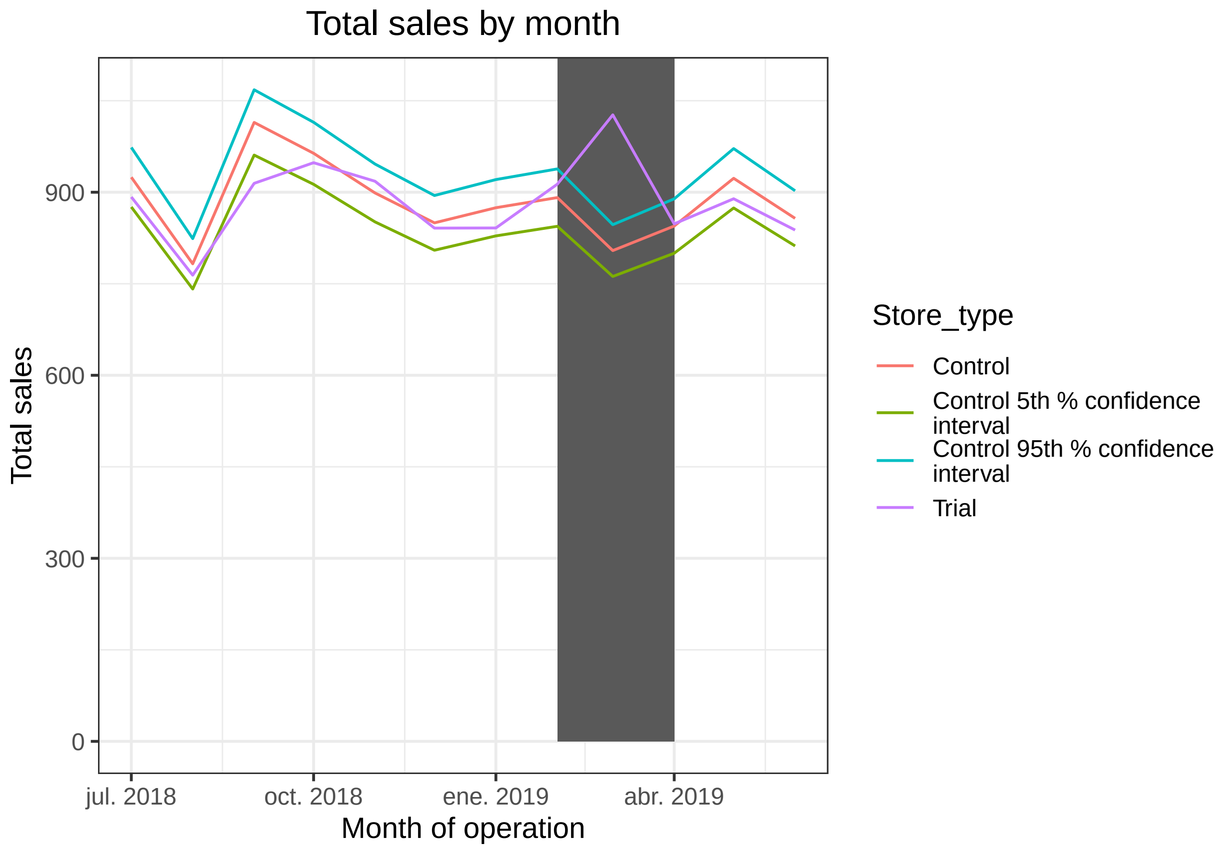
Assessment of trial Store 86

Assessing for sales

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


### Apply the scaling factor
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][ ,
                                                                    controlSales := totSales * sca


### Calculate the percentage difference between scaled control sales and trial sales
trialSales <- measureOverTimeSales[STORE_NBR == trial_store,]

percentageDiff <- merge(scaledControlSales[,c('YEARMONTH', 'controlSales')],
                        trialSales[,c('YEARMONTH','totSales')],
                        by = 'YEARMONTH')[, percentageDiff := (abs(controlSales-totSales)/
                                                          ((controlSales+totSales)/2))]

###Let's see if the difference is significant!
```

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

# Calculate the t-values for the trial months.
percentageDiff[, tValue := abs(controlSales-totSales)/stdDev][
  , TransactionMonth := as.Date(paste(as.numeric(YEARMONTH)%/%100, as.numeric(YEARMONTH)%%100, 1,
                                      sep = "-"),"%Y-%m-%d")]
percentil95 <-  qt(0.95, df = degreesOfFreedom)
```

```
### Trial and control store total sales
pSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                      ifelse(STORE_NBR == control_store, "Ot
                                                        , totalSales := mean(totSales), by = c("YEARMON
                                                          , TransactionMonth := as.Date(paste(as.numeri
                                                            Store_type %in% c("Trial", "Control"), ]

#### Control store 95th percentile
pSales_Controls95 <- pSales[Store_type == "Control",
][, totalSales := totalSales * (1 + stdDev * 2)
][, Store_type := "Control 95th % confidence
interval"]

#### Control store 5th percentile
pSales_Controls5 <- pSales[Store_type == "Control",
][, totalSales := totalSales * (1 - stdDev * 2)
][, Store_type := "Control 5th % confidence
interval"]

trialAssessment <- rbind(pSales, pSales_Controls95, pSales_Controls5)

#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, totalSales, 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")
```



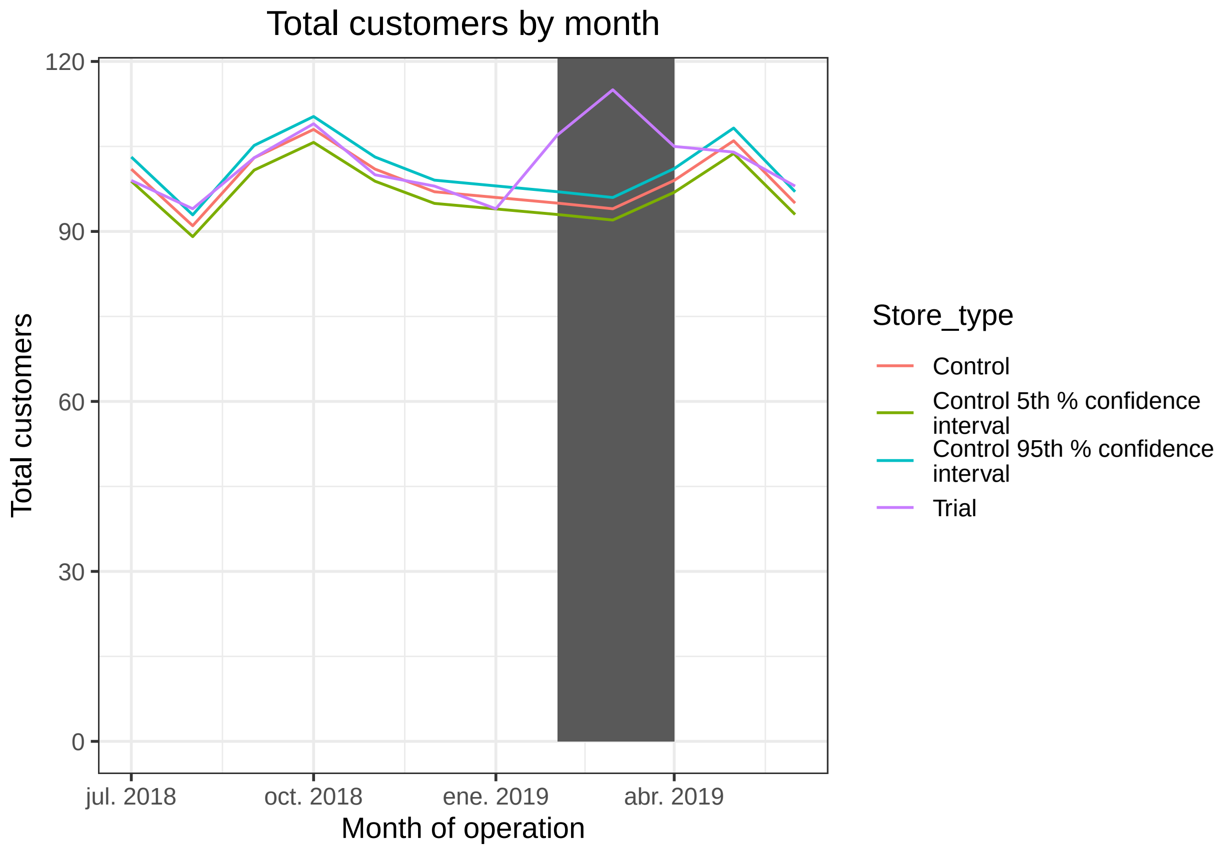
The results show that the trial in store 86 is not significantly different to its control store in the trial period as the trial store performance lies inside the 5% to 95% confidence interval of the control store in two of the three trial months.

Assessing for number of customers

```
scalingFactorForControlCustomers <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum
  preTrialMeasures[STORE_NBR == control_store & YEARMONTH < 201902, sum(nCustomers)]

### Apply the scaling factor
scaledControlCusts <- measureOverTimeCusts[STORE_NBR == control_store,
][ ,controlCustomers := nCustomers * scalingFactorForControlCustomers]


### Calculate the percentage difference between scaled control customers and trial customers
trialCusts <- measureOverTimeCusts[STORE_NBR == trial_store,]

percentageDiffCusts <- merge(scaledControlCusts[,c('YEARMONTH', 'controlCustomers')],
                             trialCusts[,c('YEARMONTH','totalCustomers')],
                             by = 'YEARMONTH')[, percentageDiff := (abs(controlCustomers-totalCustomers
                                                                   ((controlCustomers+totalCustomers



###Let's see if the difference is significant
```

```r
stdDevCust <- sd(percentageDiffCusts[YEARMONTH < 201902 , percentageDiff])

### Calculate the t-values for the trial months.

percentageDiffCusts[, tValue := abs(controlCustomers-totalCustomers)/stdDevCust
][, TransactionMonth := as.Date(paste(as.numeric(YEARMONTH)%/%100, as.numeric(YEARMONTH)%%100, 1,
                                      sep = "-"),"%Y-%m-%d")]
```

```r
### Trial and control store number of customers
pCusts <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                      ifelse(STORE_NBR == control_store, "Control", "Ot
][, totalCustomers2 := mean(totalCustomers), by = c("YEARMONTH","Store_type")
][, TransactionMonth := as.Date(paste(as.numeric(YEARMONTH)%/%100, as.numeric(YEARMONTH)%%100, 1, sep =
][Store_type %in% c("Trial", "Control"), ]

#### Control store 95th percentile
pCusts_Controls95 <- pCusts[Store_type == "Control",
][, totalCustomers2 := totalCustomers2 * (1 + stdDevCust * 2)
][, Store_type := "Control 95th % confidence
interval"]

#### Control store 5th percentile
pCusts_Controls5 <- pCusts[Store_type == "Control",
][, totalCustomers2 := totalCustomers2 * (1 - stdDevCust * 2)
][, Store_type := "Control 5th % confidence
interval"]

trialAssessmentCusts <- rbind(pCusts, pCusts_Controls95, pCusts_Controls5)

#### Plotting these in one nice graph
ggplot(trialAssessmentCusts, aes(TransactionMonth, totalCustomers2, color = Store_type)) +
  geom_rect(data = trialAssessmentCusts[ 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 customers", title = "Total customers by month")
```



It looks like the number of customers is significantly higher in all of the three months. This seems to suggest that the trial had a significant impact on increasing the number of customers in trial store 86.

Store 88

calculate correlations and magnitude against store 86 using total sales and number of customers.

```
trial_store <- 88
corr_nSales <- calculateCorrelation(preTrialMeasures, 'totSales', trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures, 'nCustomers', trial_store)

### Then, use the functions for calculating magnitude.
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales), trial_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures, quote(nCustomers), trial_store)

### combine all the scores calculated to create a composite score to rank on.
corr_weight <- 0.5

score_nSales <- merge(corr_nSales, magnitude_nSales, by = c('Store1','Store2'))[
  , scoreNSales := corr_weight*corr_measure + corr_weight*mag_measure]

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

### 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]
```

Control Store

```
controlStores <- score_Control[finalControlScore > 0.75, .(Store2, finalControlScore)]
controlStores
```

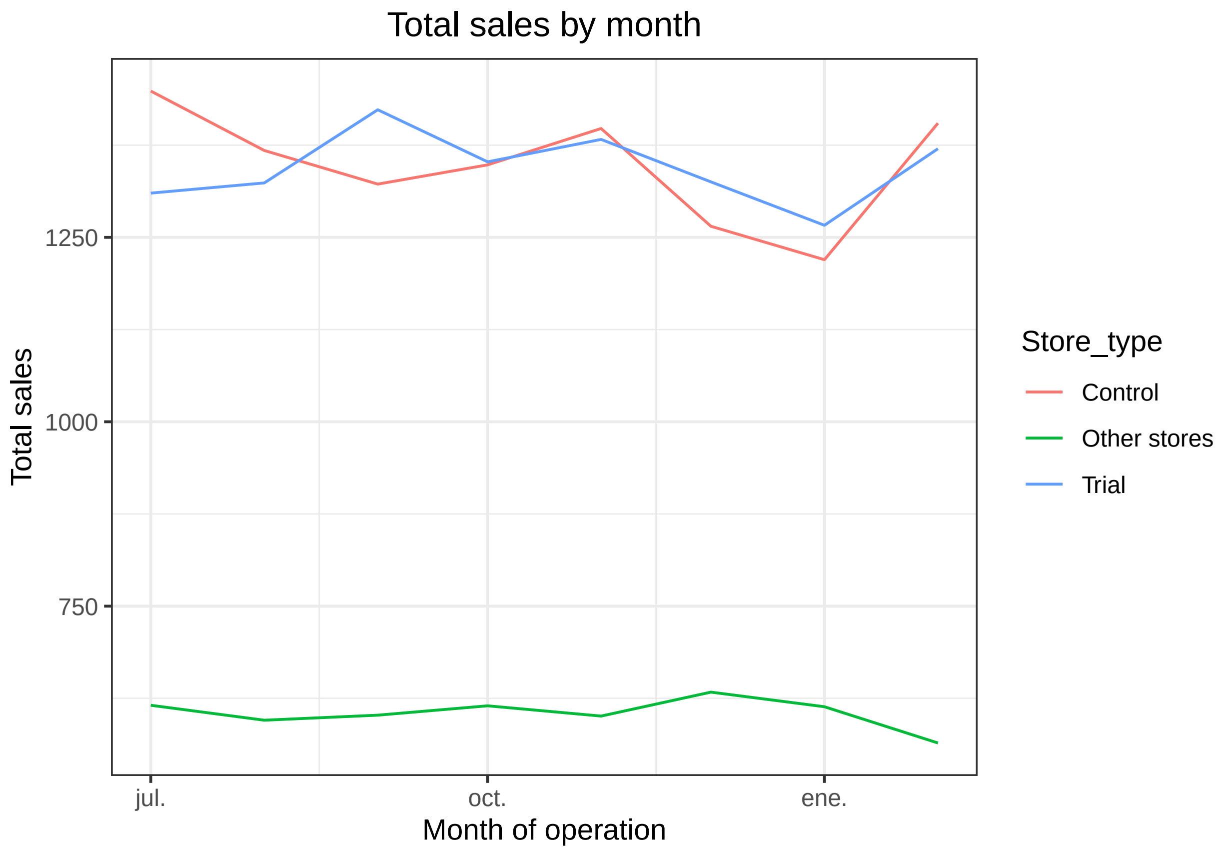
```
##    Store2 finalControlScore
## 1:     88         1.0000000
## 2:    178         0.7970827
## 3:    237         0.7998667
```

```
control_store <- 237
```

Visual checks on trends based on the drivers

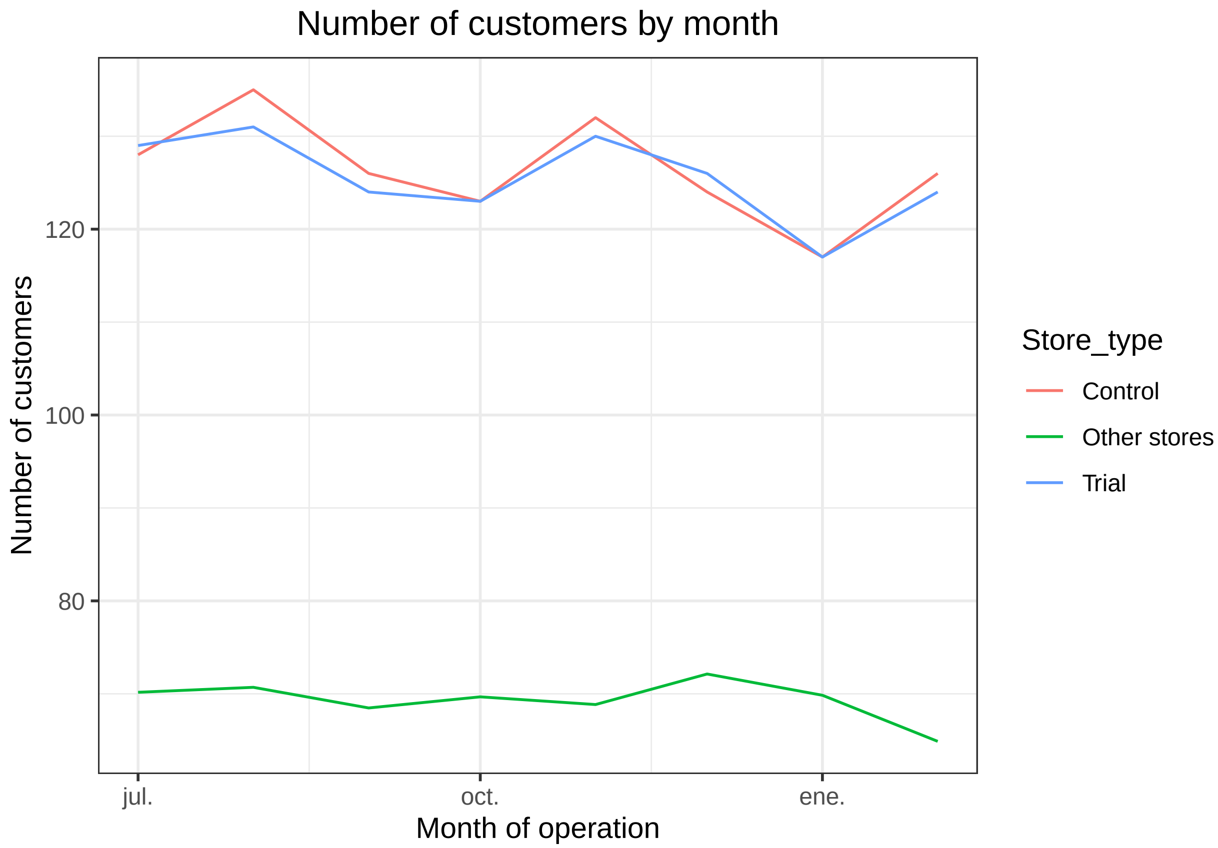
Sales

```r
measureOverTimeSales <- copy(measureOverTime)
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
            ifelse(STORE_NBR == control_store, "Control", "Other stores"))
            ][, totalSales := mean(totSales), by = c("YEARMONTH","Store_type")
            ][, TransactionMonth := as.Date(paste(as.numeric(YEARMONTH)%/%100, as.numeric(YEARMONTH)%%1
            ][as.numeric(YEARMONTH) < 201903 , ]

ggplot(pastSales, aes(TransactionMonth, totalSales, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by month")
```



Number of customers.

```r
measureOverTimeCusts <- copy(measureOverTime)
pastCustomers <- measureOverTimeCusts[,Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                  ifelse(STORE_NBR == control_store, "Control
                                                    , totalCustomers := mean(nCustomers), by
                                                      , TransactionMonth := as.Date(paste(as.
                                                        as.numeric(YEARMONTH) < 201903 , ]

ggplot(pastCustomers, aes(TransactionMonth, totalCustomers, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Number of customers", title = "Number of customers by month")
```



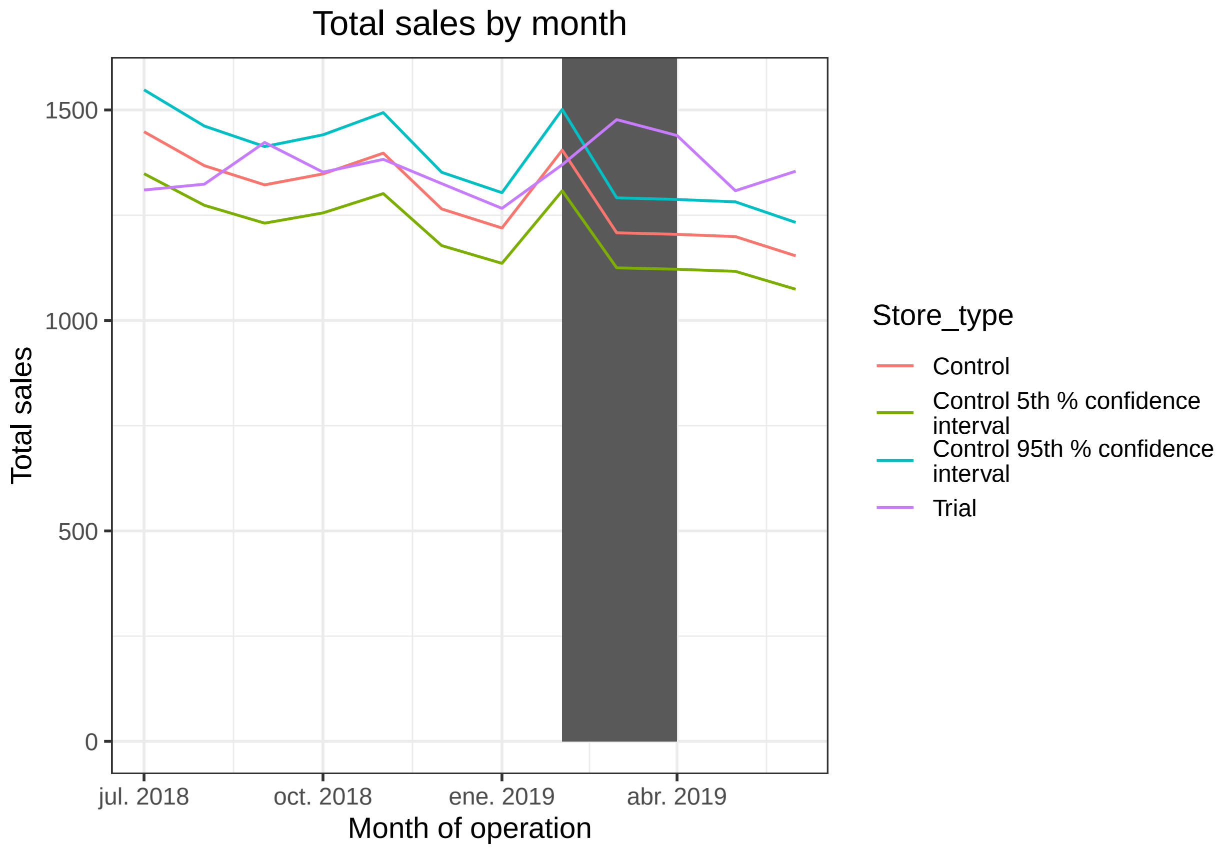
Assessment of trial Store 86

Assessing for sales

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


### Apply the scaling factor
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][ ,
                                                                          controlSales := totSales * sca


### Calculate the percentage difference between scaled control sales and trial sales
trialSales <- measureOverTimeSales[STORE_NBR == trial_store,]

percentageDiff <- merge(scaledControlSales[,c('YEARMONTH', 'controlSales')],
                        trialSales[,c('YEARMONTH','totSales')],
                        by = 'YEARMONTH')[, percentageDiff := (abs(controlSales-totSales)/
                                                                  ((controlSales+totSales)/2))]

###Let's see if the difference is significant!
```

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

# Calculate the t-values for the trial months.
percentageDiff[, tValue := abs(controlSales-totSales)/stdDev][
  , TransactionMonth := as.Date(paste(as.numeric(YEARMONTH)%/%100, as.numeric(YEARMONTH)%%100, 1,
                                      sep = "-"),"%Y-%m-%d")]
```

```r
### Trial and control store total sales
pSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                       ifelse(STORE_NBR == control_store, "Ot
                                                         , totalSales := mean(totSales), by = c("YEARMON
                                                           , TransactionMonth := as.Date(paste(as.numeri
                                                             Store_type %in% c("Trial", "Control"), ]

#### Control store 95th percentile
pSales_Controls95 <- pSales[Store_type == "Control",
][, totalSales := totalSales * (1 + stdDev * 2)
][, Store_type := "Control 95th % confidence
interval"]

#### Control store 5th percentile
pSales_Controls5 <- pSales[Store_type == "Control",
][, totalSales := totalSales * (1 - stdDev * 2)
][, Store_type := "Control 5th % confidence
interval"]

trialAssessment <- rbind(pSales, pSales_Controls95, pSales_Controls5)

#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, totalSales, 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")
```



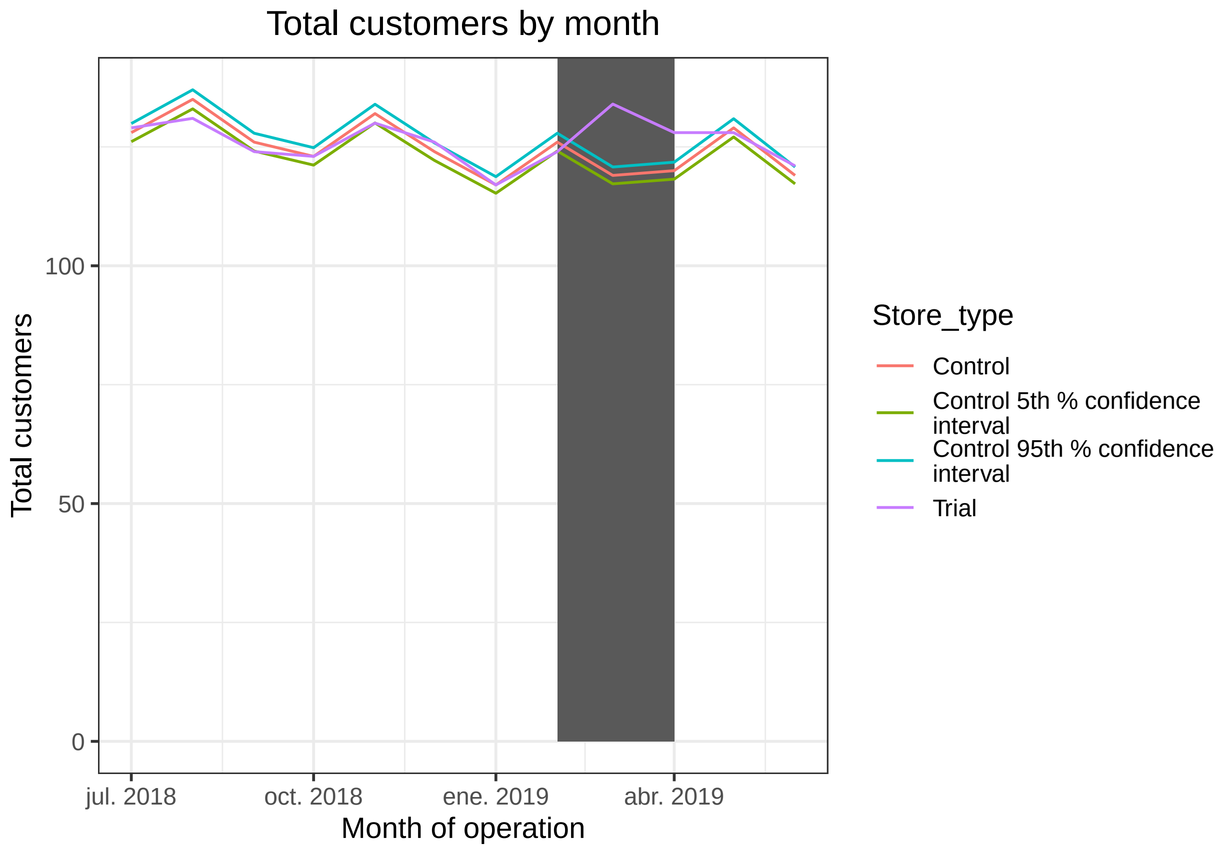
The results show that the trial in store 88 is significantly different to its control store in the trial period in two of the three trial months.

Assessing for number of customers

```
scalingFactorForControlCustomers <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum
  preTrialMeasures[STORE_NBR == control_store & YEARMONTH < 201902, sum(nCustomers)]

### Apply the scaling factor
scaledControlCusts <- measureOverTimeCusts[STORE_NBR == control_store,
][ ,controlCustomers := nCustomers * scalingFactorForControlCustomers]


### Calculate the percentage difference between scaled control customers and trial customers
trialCusts <- measureOverTimeCusts[STORE_NBR == trial_store,]

percentageDiffCusts <- merge(scaledControlCusts[,c('YEARMONTH', 'controlCustomers')],
                             trialCusts[,c('YEARMONTH','totalCustomers')],
                             by = 'YEARMONTH')[, percentageDiff := (abs(controlCustomers-totalCustomers
                                                                  ((controlCustomers+totalCustomers


###Let's see if the difference is significant
```

```r
stdDevCust <- sd(percentageDiffCusts[YEARMONTH < 201902 , percentageDiff])

### Calculate the t-values for the trial months.

percentageDiffCusts[, tValue := abs(controlCustomers-totalCustomers)/stdDevCust
][, TransactionMonth := as.Date(paste(as.numeric(YEARMONTH)%/%100, as.numeric(YEARMONTH)%%100, 1,
                                      sep = "-"),"%Y-%m-%d")]
```

```r
### Trial and control store number of customers
pCusts <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                      ifelse(STORE_NBR == control_store, "Control", "Ot
][, totalCustomers2 := mean(totalCustomers), by = c("YEARMONTH","Store_type")
][, TransactionMonth := as.Date(paste(as.numeric(YEARMONTH)%/%100, as.numeric(YEARMONTH)%%100, 1, sep =
][Store_type %in% c("Trial", "Control"), ]

#### Control store 95th percentile
pCusts_Controls95 <- pCusts[Store_type == "Control",
][, totalCustomers2 := totalCustomers2 * (1 + stdDevCust * 2)
][, Store_type := "Control 95th % confidence
interval"]

#### Control store 5th percentile
pCusts_Controls5 <- pCusts[Store_type == "Control",
][, totalCustomers2 := totalCustomers2 * (1 - stdDevCust * 2)
][, Store_type := "Control 5th % confidence
interval"]

trialAssessmentCusts <- rbind(pCusts, pCusts_Controls95, pCusts_Controls5)

#### Plotting these in one nice graph
ggplot(trialAssessmentCusts, aes(TransactionMonth, totalCustomers2, color = Store_type)) +
  geom_rect(data = trialAssessmentCusts[ 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 customers", title = "Total customers by month")
```



Total number of customers in the trial period for the trial store is significantly higher than the control store for two out of three months, which indicates a positive trial effect.

We've found control stores 233, 155, 237 for trial stores 77, 86 and 88 respectively.

The results for trial stores 77 and 88 during the trial period show a significant difference in at least two of the three trial months but this is not the case for trial store 86. We can check with the client if the implementation of the trial was different in trial store 86 but overall, the trial shows a significant increase in sales. Now that we have finished our analysis, we can prepare our presentation to the Category Manager.