Retail Strategy and Analytics - Task 2

Gavan Corke

```
library(tidyr)
library(readxl)
library(data.table)
library(ggplot2)
library(ggmosaic)
library(readr)
library(stringr)
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(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(lubridate)
```

```
## Loading required package: timechange

##
## 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
```

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.

Task 2

```r
#-------------------------#
# Data Upload and Cleaning
#-------------------------#


filePath <- ""
qvi_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
# Create new column month ID.
# Want to find control stores simlar to the three chosen (77, 86, 88)
# In terms of monthly overall sales, customers and transactions per customer.
# Use floor_date function on current date columns to get it into yyyymm format

# convert DATE column into 'date' data type first

qvi_data[, YEARMONTH := format(DATE, "%Y%m")]

#For each store and month calculate
#       total sales,
#       number of customers,
#       transactions per customer,
#       chips per customer and the average price per unit.
```

```r
# Hint: you can use uniqueN() to count distinct values in a column.

measureOverTime <- qvi_data[, .(totSales = sum(TOT_SALES),
                             nCustomers = uniqueN(LYLTY_CARD_NBR),
                            nTxnPerCust = length(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)]



#In the data.table package in R, the special symbol .N is used to refer to the
#number of rows in the current group of data.

#pre-trial period is before Feb 2019.

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

# Gives us the stores which have full observations.

storesWithFullObs <- unique(measureOverTime[, .N, STORE_NBR][N == 12, STORE_NBR])

# the data of the stores before the trial period, so before feburary 2019.

preTrialMeasures <- measureOverTime[YEARMONTH < 201902 & STORE_NBR %in%
                                      storesWithFullObs, ]

#Create a function as 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.

# Use pearson correlation.

#Let's define:
#inputTable - as a metric table with potential comparison stores,
#metricCol - as the store metric (e.g total sales) used to calculate correlation on, and
#storeComparison - as the store number of the trial store.

# The function below takes a data.table a column of interest and trial store and runs
# a pearson correlation between the trial store and every other store chosen. e.g.(those)
# stores where there is full observations over 12 months and the preTrial data.
```

```r
calculateCorrelation <- function(inputTable, metricCol , storeComparison){
  # USE QUOTE() Function for metricCol
  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(x =inputTable[STORE_NBR == storeComparison, eval(metricCol)]
                                          y = inputTable[STORE_NBR==i,
```

```
                                                   eval(metricCol)])
                                    )

      calcCorrTable <- rbind(calcCorrTable, calculatedMeasure)
      # 'fills in the empty calcCorrTable I believe?
    }
  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.

# Create a function to calculate a standardised magnitude distance for a measure,
# looping through each control store.



calculateMagnitudeDistance <- function(inputTable, metricCol, storeComparison){
  calcDistTable = data.table(Store1 = numeric(), Store2 = numeric(), YEARMONTH =
                               numeric(), measure = numeric())
  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(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("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)
}


#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.

# Calculate correlations against store 77 (one of the trial stores)
```

```r
# correlation between store 77 and potential control stores w.r.t total sales

trial_sales_77 <- calculateCorrelation(preTrialMeasures, quote(totSales), 77)

# correlation between store 77 and potential control stores w.r.t ncustomers
# per month

trial_customers_77 <- calculateCorrelation(preTrialMeasures, quote(nCustomers), 77)

#magnitude difference sales

dist_sales_77 <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales),
                                            77)

#magnitude difference no customers per month

dist_month_77 <- calculateMagnitudeDistance(preTrialMeasures, quote(nCustomers),
                                            77)
```

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. Note that if we consider it more important for the trend of the drivers to be similar, we can increase the weight of the correlation score (a simple average gives a weight of 0.5 to the corr_weight) or if we consider the absolute size of the drivers to be more important, we can lower the weight of the correlation score.

```r
corr_weight <- 0.5

score_nSales <- merge(trial_sales_77, dist_sales_77,
                      by = c("Store1","Store2"))[, scoreNSales := 0.5*(corr_measure+mag_measure)]
score_nCustomers <- merge(trial_customers_77,dist_month_77,
                      by = c("Store1","Store2"))[, scoreNCust := (0.5*corr_measure+0.5*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.

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

#Select store to be the control if it has the highest finalControlScore (and it is
# not store 77)
```

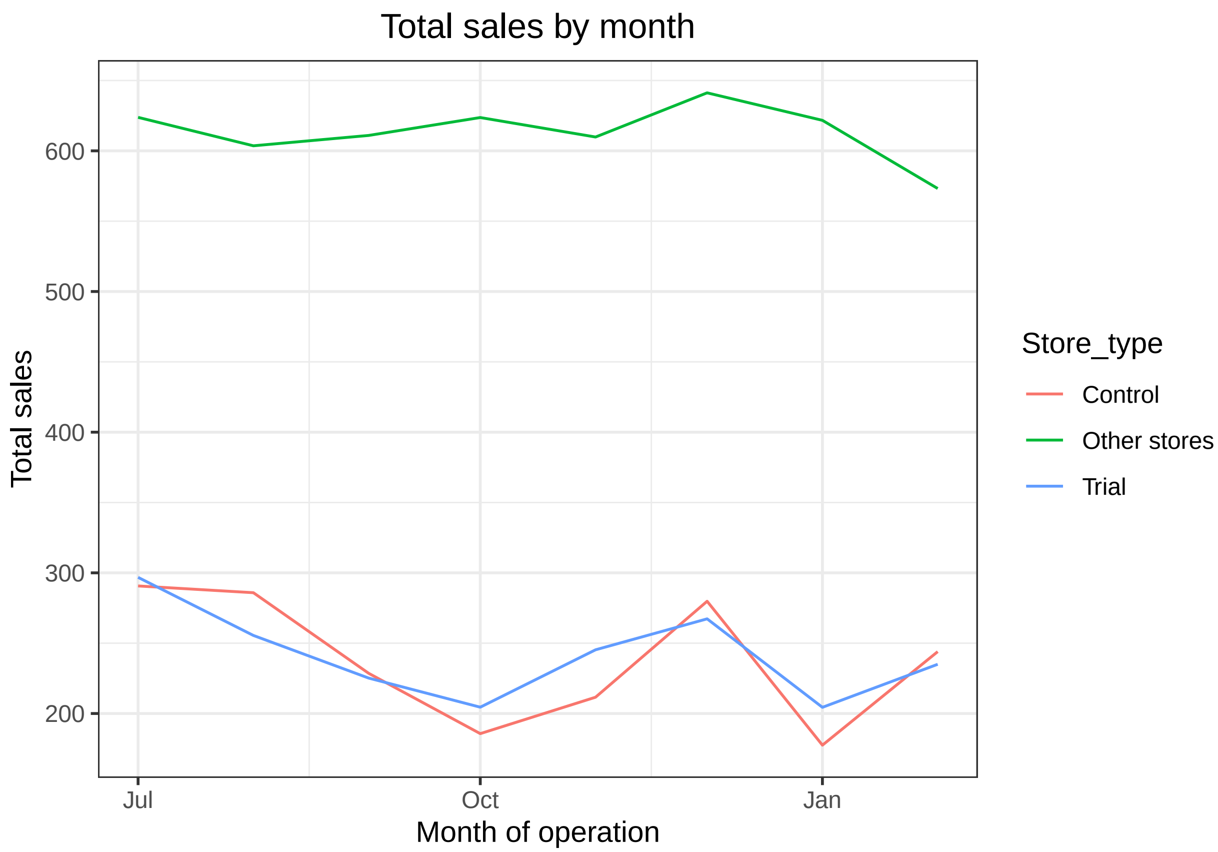
```r
score_Control[finalControlScore == sort(score_Control[ , finalControlScore], decreasing = TRUE)[2], ]
```

```
##    Store1 Store2 corr_measure.x mag_measure.x scoreNSales corr_measure.y
## 1:     77    233      0.9037742     0.9852649   0.9445195      0.9903578
##    mag_measure.y scoreNCust finalControlScore
## 1:     0.9927733  0.9915655         0.9680425
```

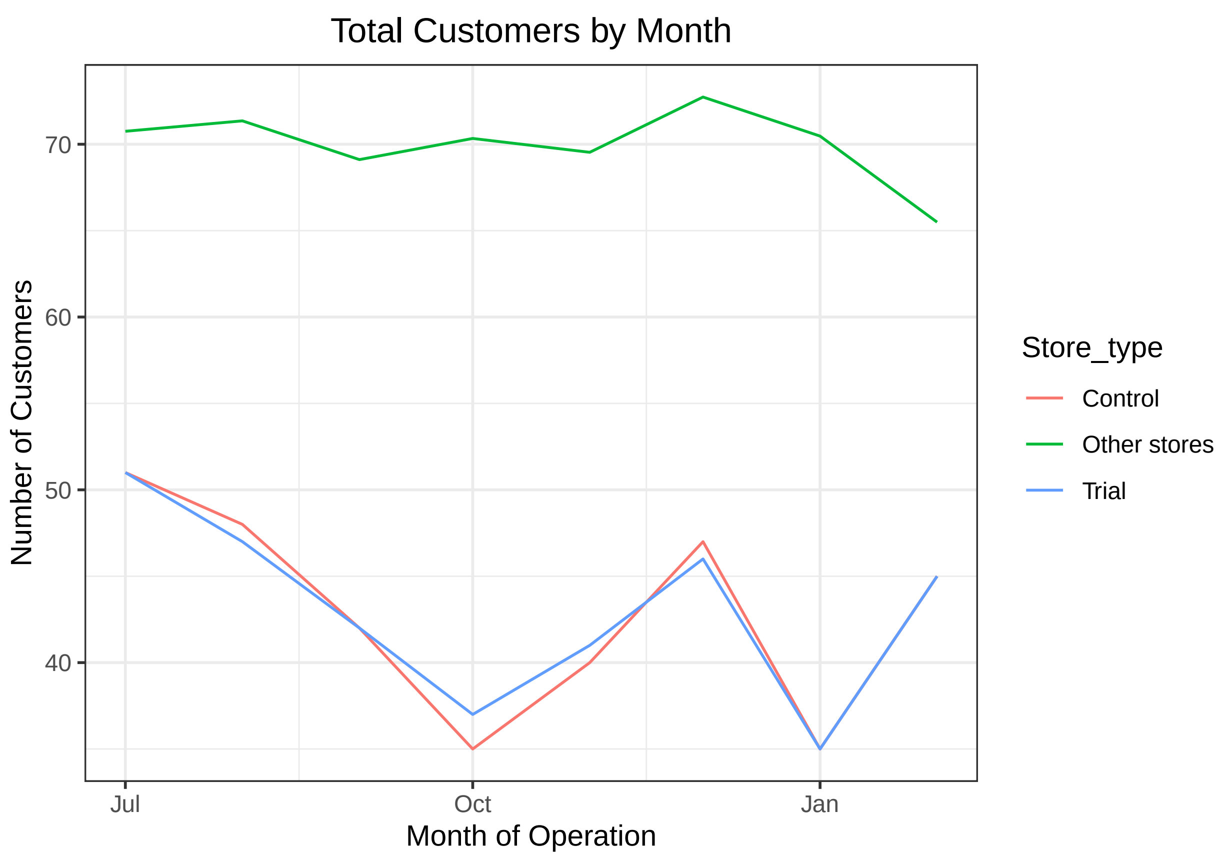
```r
#store 233 as control for store 77

control_store <- 233
trial_store <- 77
```

```r
#convert YEARMONTH so binary operator can be used in the function



measureOverTimeSales <- measureOverTime

measureOverTimeSales[ , YEARMONTH := as.numeric(YEARMONTH) ]
```

```r
# Look at total past sales.

pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store,
                                                         "Trial",
                                                         ifelse(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")
```



```r
# Look at past number of customers

measureOverTimeCusts <- measureOverTime

measureOverTimeCusts[ , YEARMONTH := as.numeric(YEARMONTH)]

pastCustomers <- measureOverTimeCusts[, Store_type:= ifelse(STORE_NBR== trial_store,
                                                            "Trial",
                                                            ifelse(STORE_NBR == control_store,
                                                                   "Control", "Other stores"))
][, noCustomers := mean(nCustomers), by = c("YEARMONTH",
                                            "Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/%
                                        100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
][YEARMONTH < 201903, ]
ggplot(pastCustomers, aes(TransactionMonth, noCustomers, color = Store_type)) +
  geom_line() +
  labs(x = "Month of Operation", y = "Number of Customers", title = "Total Customers by Month")
```



#Trial Period Assessment

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

```r
# 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)]

# total sales for trial store over the months / total sales for control store over the months

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

#What we are trying to do here is make the pre-trial sales similar so that
# we can see if the difference between the two during the trial is noticeable
# and not just due to the possible large difference during the pre-trial. So they
# start from a similar baseline.

#Calculate the percentage difference between scaled control sales
#and trial sales.

percentageDiff <- merge(measureOverTimeSales[STORE_NBR == trial_store,
                                             c("totSales", "YEARMONTH")],
                        scaledControlSales[ , c("controlSales", "YEARMONTH")],
                        by = "YEARMONTH"
                          )[, percentageDiff := abs(totSales-controlSales)/
                              (0.5*(totSales+controlSales))]



# to see it in percentage

pdiff_100 <- percentageDiff[ , percentageDiff_100:= percentageDiff*100]
```

#T-test

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.

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

#Note that there are 8 months in the pre-trial period

length(percentageDiff[YEARMONTH < 201902, percentageDiff])
```

```
## [1] 7
```

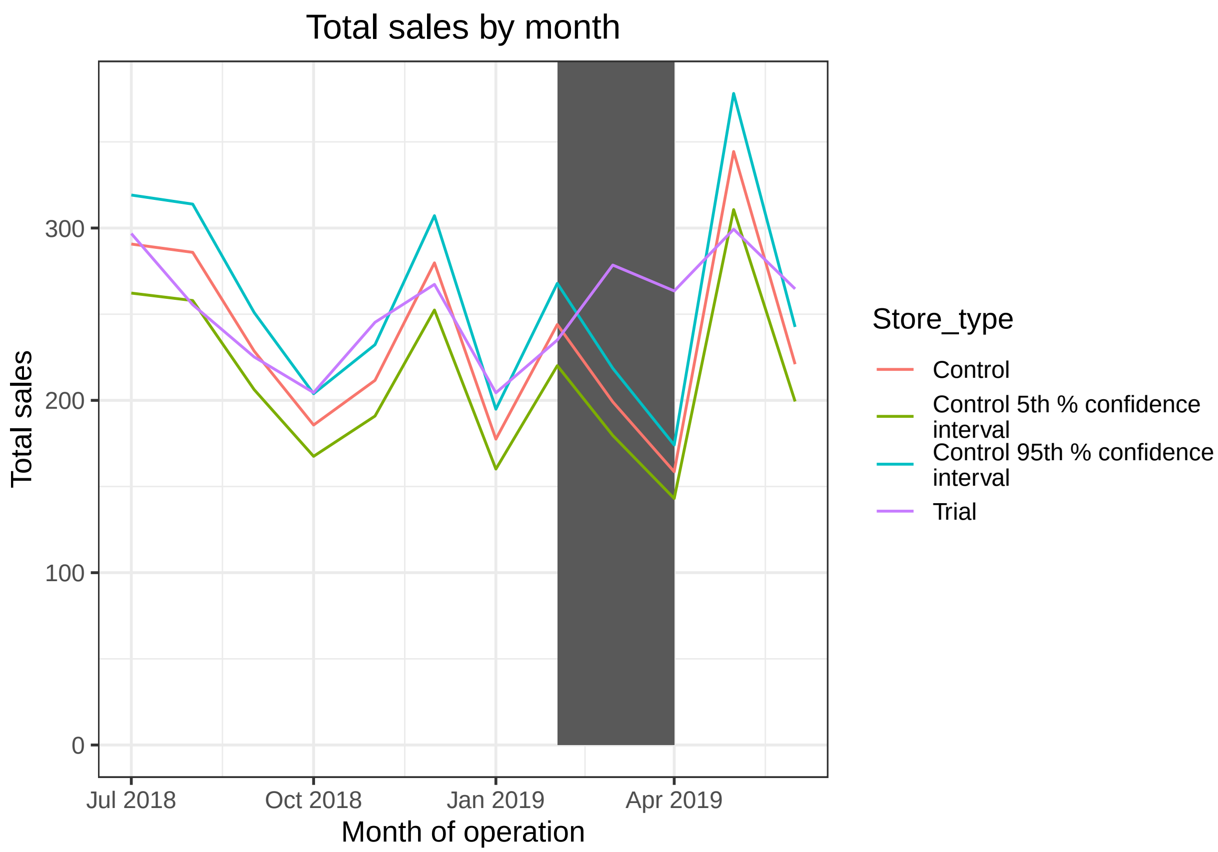
```r
# hence 8 - 1 = 7 degrees of freedom


degreesOfFreedom <- 7

# Calculate t-value for each month. (Compare trial vs control store during the
# trial months)
```

```r
percentageDiff[, tValue := (percentageDiff-0)/stdDev
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1,
                                      sep = "-"), "%Y-%m-%d")
][YEARMONTH < 201905 & YEARMONTH > 201901, .(TransactionMonth,tValue)]
```

```
##    TransactionMonth   tValue
## 1:       2019-02-01 1.244840
## 2:       2019-03-01 6.330932
## 3:       2019-04-01 9.709819
```

```r
#TransactionMonth   tValue
#1:       2019-02-01 1.244840
#2:       2019-03-01 6.330932
#3:       2019-04-01 9.709819

#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
# Over to you! Create new variables Store_type, totSales and TransactionMonth in
#the data table.

pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store,
                        "Trial", ifelse(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")
][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.
```

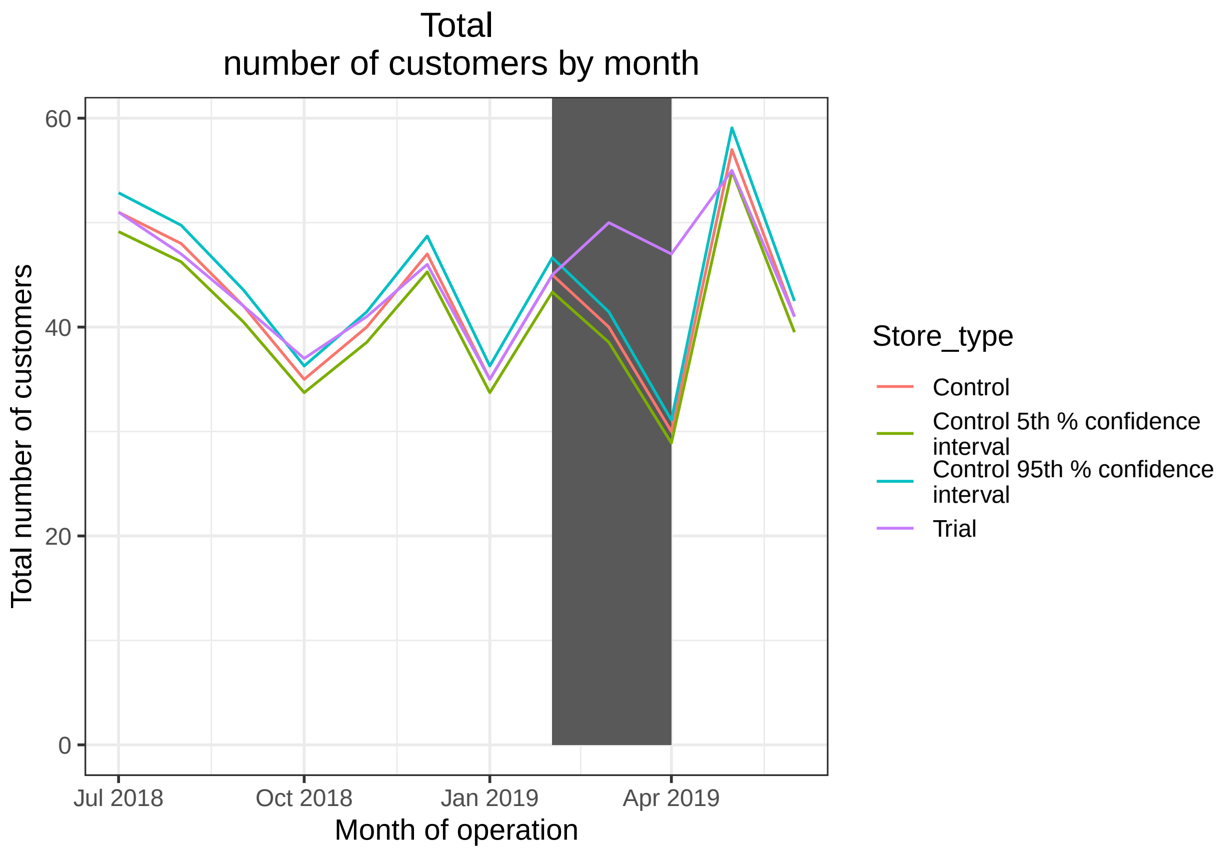
```
#### This would be a repeat of the steps before for total sales
#### Scale pre-trial control customers to match pre-trial trial store customers

scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store &
                                                YEARMONTH < 201902, sum(nCustomers)]/preTrialMeasures


#### 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"))
```

```r
]
#Calculate the percentage difference between scaled control sales and trial
#sales

percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH",
                                                    "controlCustomers")],

                        measureOverTimeCusts[STORE_NBR == trial_store,
                                             c("nCustomers", "YEARMONTH")],
                        by = "YEARMONTH"
)[, percentageDiff :=
    abs(controlCustomers-nCustomers)/controlCustomers]


#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])

percentageDiff[, tValue := (percentageDiff-0)/stdDev
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1,
                                      sep = "-"), "%Y-%m-%d")
][YEARMONTH < 201905 & YEARMONTH > 201901, .(TransactionMonth,tValue)]
```

```
##    TransactionMonth     tValue
## 1:       2019-02-01  0.1833522
## 2:       2019-03-01 13.4763876
## 3:       2019-04-01 30.7787247
```

```r
t_vals2 <- percentageDiff[, tValue := (percentageDiff-0)/stdDev
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1,
                                      sep = "-"), "%Y-%m-%d")

][YEARMONTH < 201905 & YEARMONTH > 201901, .(TransactionMonth,tValue)]

# Trial and control store number of customers

pastCustomers <- measureOverTimeCusts[, nCusts := mean(nCustomers), by =
                                        c("YEARMONTH", "Store_type")
][Store_type %in% c("Trial", "Control"), ]

# Control store 95th percentile

pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",
][, nCusts := nCusts * (1 + stdDev * 2)
][, Store_type := "Control 95th % confidence
interval"]

# Control store 5th percentile
pastCustomers_Controls5 <- pastCustomers[Store_type == "Control",
][, nCusts := nCusts * (1 - stdDev * 2)
][, Store_type := "Control 5th % confidence
```

```
interval"]
trialAssessment <- rbind(pastCustomers, pastCustomers_Controls95,
                         pastCustomers_Controls5)

#Plot these visually too.

#Hint: geom_rect creates a rectangle in the plot. Use this to highlight the
#trial period in our graph.

ggplot(trialAssessment, aes(TransactionMonth, nCusts, 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 number of customers", title = "Total
 number of customers by month")
```




```
#Now we can repeat the same methodology and outline for the other two stores 86 and 88.


measureOverTime <- qvi_data[, .(totSales = sum(TOT_SALES),
                                nCustomers = uniqueN(LYLTY_CARD_NBR),
                                nTxnPerCust = length(TXN_ID)/uniqueN(LYLTY_CARD_NBR),
                                nChipsPerTxn = sum(PROD_QTY)/uniqueN(TXN_ID),
                                avgPricePerUnit = sum(TOT_SALES)/sum(PROD_QTY)),
```

```r
                            by = c("STORE_NBR", "YEARMONTH")][order(STORE_NBR, YEARMONTH)]

#Use the functions we created earlier to calculate correlations
#and magnitude for each potential control store.

trial_store <- 86
# correlation between store 86 and potential control stores w.r.t total sales

trial_sales_86 <- calculateCorrelation(preTrialMeasures, quote(totSales), 86)

trial_customers_86 <- calculateCorrelation(preTrialMeasures, quote(nCustomers),86)

#magnitude difference no customers per month

dist_sales_86 <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales),
                                           86)

dist_month_86 <- calculateMagnitudeDistance(preTrialMeasures, quote(nCustomers),
                                           86)
#combined score composed of correlation and magnitude

corr_weight <- 0.5

score_nSales2 <- merge(trial_sales_86, dist_sales_86,
                      by = c("Store1","Store2"))[, scoreNSales2 := 0.5*(corr_measure+mag_measure)]

score_nCustomers2 <- merge(trial_customers_86,dist_month_86,
                          by = c("Store1","Store2"))[, scoreNCust2 := (0.5*corr_measure+0.5*mag_measure)


score_Control2 <- merge(score_nSales2, score_nCustomers2, by = c("Store1", "Store2"))

score_Control2[, finalControlScore := scoreNSales2 * 0.5 + scoreNCust2 * 0.5]

score_Control2[finalControlScore == sort(score_Control2[ , finalControlScore], decreasing = TRUE)[2], ]
```

```
##    Store1 Store2 corr_measure.x mag_measure.x scoreNSales2 corr_measure.y
## 1:     86    155      0.8778817     0.9629637    0.9204227      0.9428756
##    mag_measure.y scoreNCust2 finalControlScore
## 1:     0.9850373   0.9639565         0.9421896
```

```r
#store 155 as control for store 77

control_store <- 155
trial_store <- 86

measureOverTimeSales <- measureOverTime

measureOverTimeSales[ , YEARMONTH := as.numeric(YEARMONTH) ]

# Trial and control store total sales
# Over to you! Create new variables Store_type, totSales and TransactionMonth in
#the data table.
```

```r
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store,
                                                         "Trial", ifelse(STORE_NBR == control_store, "Co
][, totSales := mean(totSales), by = c("YEARMONTH","Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
][Store_type %in% c("Trial", "Control"), ][YEARMONTH < 201903]

ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line(aes(linetype = Store_type)) +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by
 month - Trial store 86 test")
```


```r
#sales are trending in a similar way.Next, number of customers.
#Scaling factor again


measureOverTimeCusts <- measureOverTime

pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR ==
                                                                trial_store, "Trial",
                                                             ifelse(STORE_NBR == control_store,
                                                                    "Control", "Other stores"))
][, noCustomers := mean(nCustomers), by =
     c("YEARMONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/%
                                        100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
```

```
][YEARMONTH < 201903 , ]

ggplot(pastCustomers, aes(TransactionMonth, noCustomers, color =
                             Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Total number of customers", title = "Total
 number of customers by month - Trial store 86 test")
```

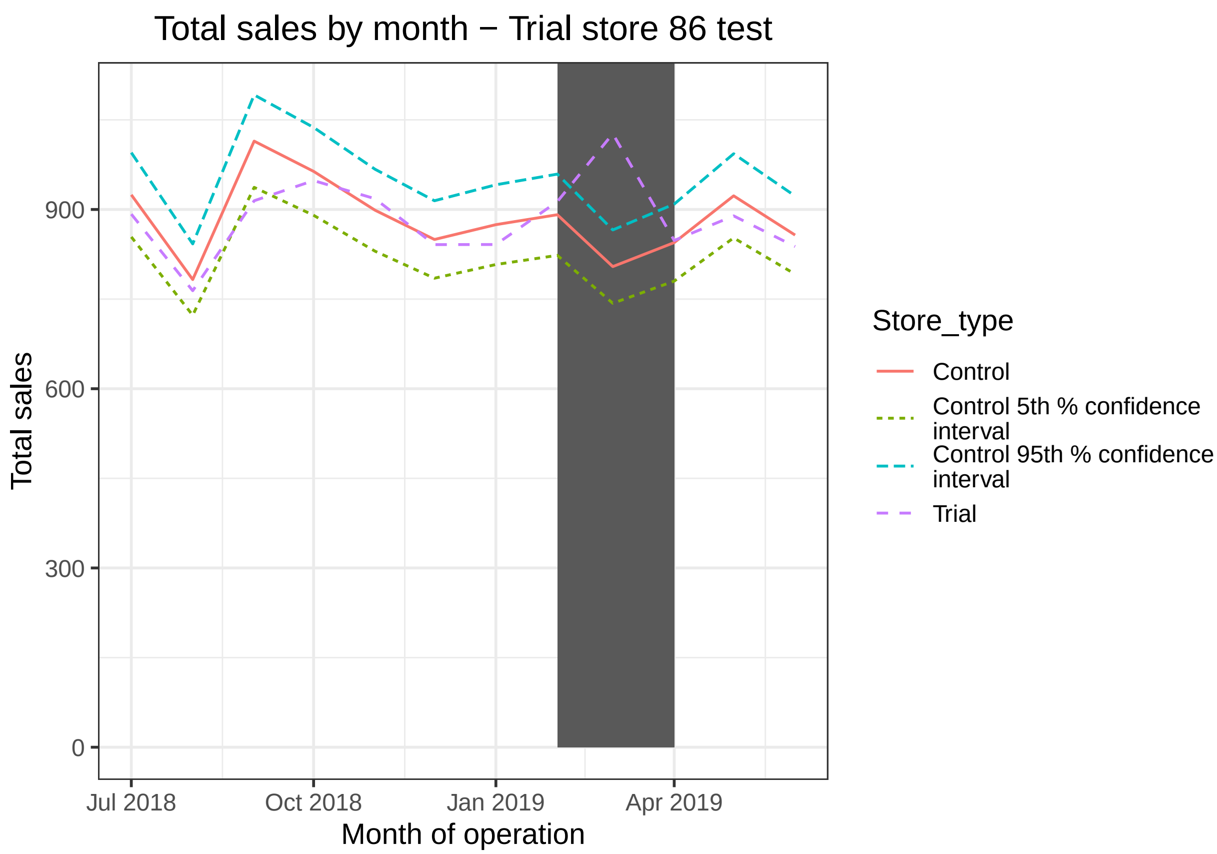

```
#


calingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
                              YEARMONTH < 201902, sum(totSales)]/
  preTrialMeasures[STORE_NBR == control_store & YEARMONTH < 201902, sum(totSales)]


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"))
]
```

```r
# Apply the scaling factor

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


#Calculate the percentage difference between scaled control sales
#and trial sales

percentageDiff <- merge(measureOverTimeSales[STORE_NBR == trial_store,
                                             c("totSales", "YEARMONTH")],
                        scaledControlSales[ , c("controlSales", "YEARMONTH")],
                        by = "YEARMONTH"
)[, percentageDiff := abs(totSales-controlSales)/
    (0.5*(totSales+controlSales))]

#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

#Calculate the standard deviation of percentage differences during
#the pre-trial period.

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

degreesOfFreedom <- 7


percentageDiff[, tValue := (percentageDiff-0)/stdDev
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1,
                                      sep = "-"), "%Y-%m-%d")
][YEARMONTH < 201905 & YEARMONTH > 201901, .(TransactionMonth,tValue)]
```

```
##    TransactionMonth     tValue
## 1:       2019-02-01 0.02735318
## 2:       2019-03-01 5.76447221
## 3:       2019-04-01 0.50046234
```

```r
#Trial and control store total sales

#Create a table with sales by store type and month.
#### Hint: We only need data for the trial and control store.


measureOverTimeSales <- measureOverTime

pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store,
                                                         "Trial", ifelse(STORE_NBR == control_store, "C
][, totSales := mean(totSales), by = c("YEARMONTH","Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/%
                                        100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
][Store_type %in% c("Trial", "Control"), ]
```

```r
#Calculate the 5th and 95th percentile for control store sales.
#### Hint: The 5th and 95th percentiles can be approximated by using two standard
#deviations away from the mean.
#### Hint2: Recall that the variable stdDev earlier calculates standard deviation
#in percentages, and not dollar sales.

# 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"]

#Then, create a combined table with columns from pastSales,
#pastSales_Controls95 and pastSales_Controls5

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(aes(linetype = Store_type)) +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by month - Trial store 86 test
```



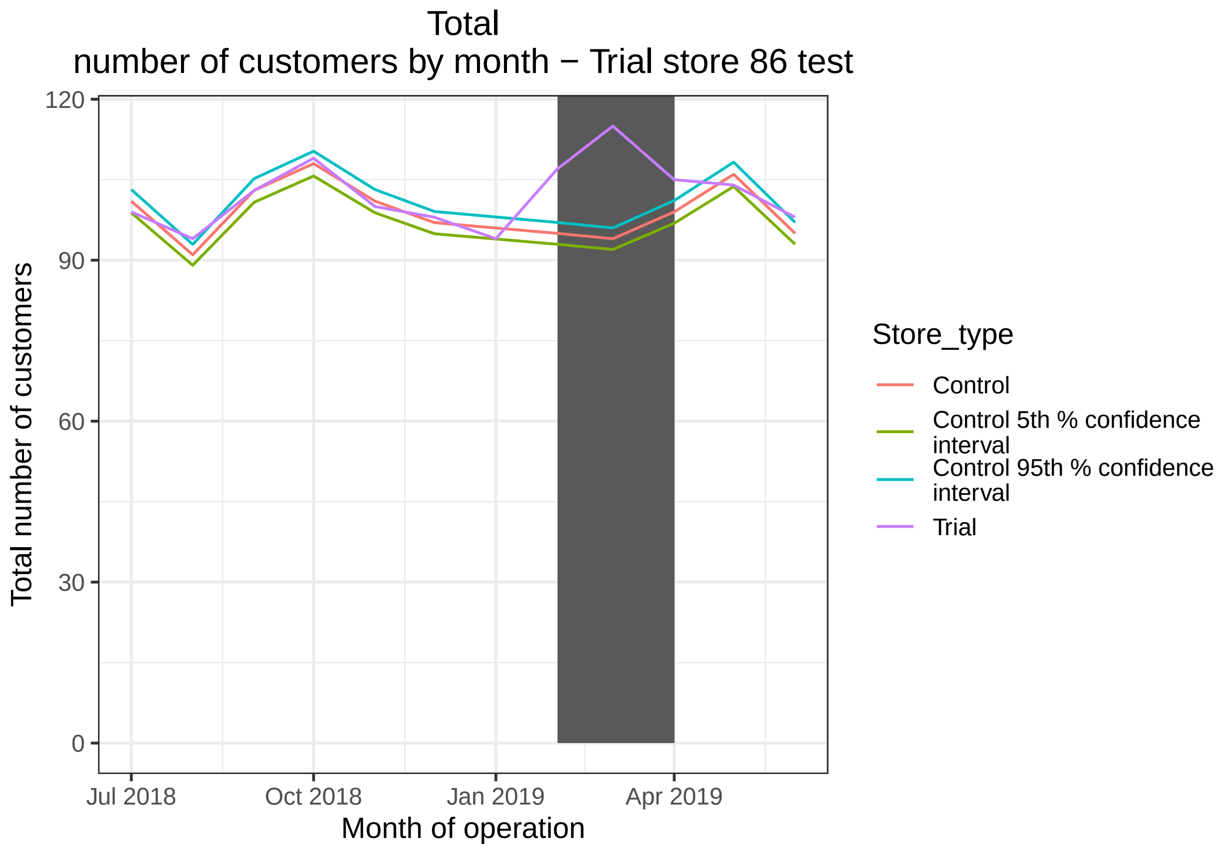
The results show that the trial in store 86 is not significantly different to its control store in the trial period as the trial store performance lies inside the 5% to 95% confidence interval of the control store in two of the three trial months. Let's have a look at assessing this for the number of 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, sum(nCustomers)]/preTrialMeasures[STORE_NBR == control_store &
YEARMONTH < 201902, sum(nCustomers)]

#### 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")],
```

```r
                        measureOverTimeCusts[STORE_NBR == trial_store,
                                             c("nCustomers", "YEARMONTH")],
                        by = "YEARMONTH"
)[, percentageDiff :=
    abs(controlCustomers-nCustomers)/controlCustomers]

#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

percentageDiff[, tValue := (percentageDiff-0)/stdDev
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1,
sep = "-"), "%Y-%m-%d")
][YEARMONTH < 201905 & YEARMONTH > 201901, .(TransactionMonth,tValue)]
```

```
##    TransactionMonth    tValue
## 1:       2019-02-01 11.819082
## 2:       2019-03-01 20.903430
## 3:       2019-04-01  5.670772
```

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

#### Control store 95th percentile
pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",
][, nCusts := nCusts * (1 + stdDev * 2)
][, Store_type := "Control 95th % confidence
interval"]
#### Control store 5th percentile
pastCustomers_Controls5 <- pastCustomers[Store_type == "Control",
][, nCusts := nCusts * (1 - stdDev * 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 = 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 number of customers", title = "Total
number of customers by month - Trial store 86 test")
```



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 88 but as we saw, sales were not significantly higher. We should check with the Category Manager if there were special deals in the trial store that were may have resulted in lower prices, impacting the results.

Trial Store 88

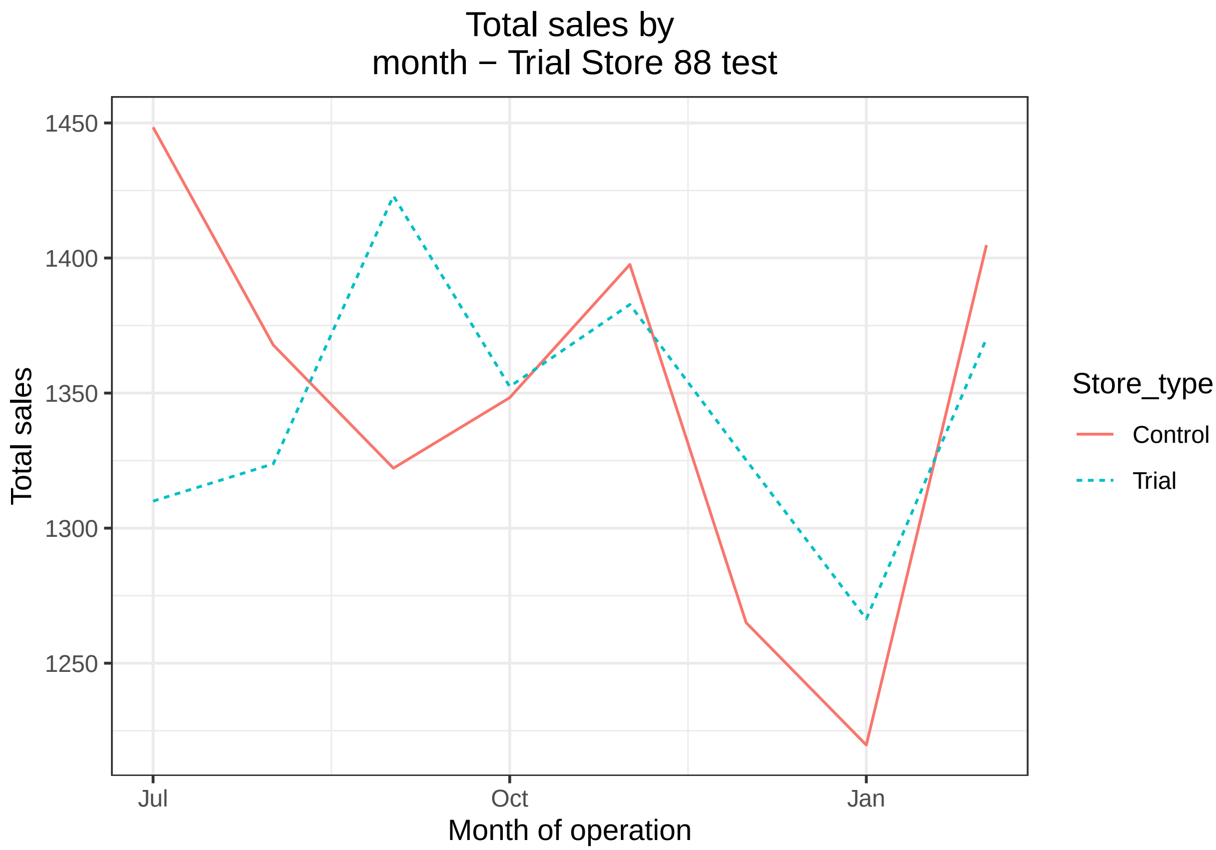
```
# Now repeat the same process for trial store 88

measureOverTime <- qvi_data[, .(totSales = sum(TOT_SALES),
                                nCustomers = uniqueN(LYLTY_CARD_NBR),
                                nTxnPerCust = length(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)]

#Use the functions we created earlier to calculate correlations
#and magnitude for each potential control store.

trial_store <- 88
# correlation between store 88 and potential control stores w.r.t total sales

trial_sales_88 <- calculateCorrelation(preTrialMeasures, quote(totSales), 88)
```

```r
trial_customers_88 <- calculateCorrelation(preTrialMeasures, quote(nCustomers),88)

#magnitude difference no customers per month

dist_sales_88 <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales),
                                            88)

dist_month_88 <- calculateMagnitudeDistance(preTrialMeasures, quote(nCustomers),
                                            88)
#combined score composed of correlation and magnitude

corr_weight <- 0.5

score_nSales2 <- merge(trial_sales_88, dist_sales_88,
                       by = c("Store1","Store2"))[, scoreNSales2 := 0.5*(corr_measure+mag_measure)]

score_nCustomers2 <- merge(trial_customers_88,dist_month_88,
                           by = c("Store1","Store2"))[, scoreNCust2 := (0.5*corr_measure+0.5*mag_measur


score_Control2 <- merge(score_nSales2, score_nCustomers2, by = c("Store1", "Store2"))

score_Control2[, finalControlScore := scoreNSales2 * 0.5 + scoreNCust2 * 0.5]

score_Control2[finalControlScore == sort(score_Control2[ , finalControlScore], decreasing = TRUE)[2], ]
```

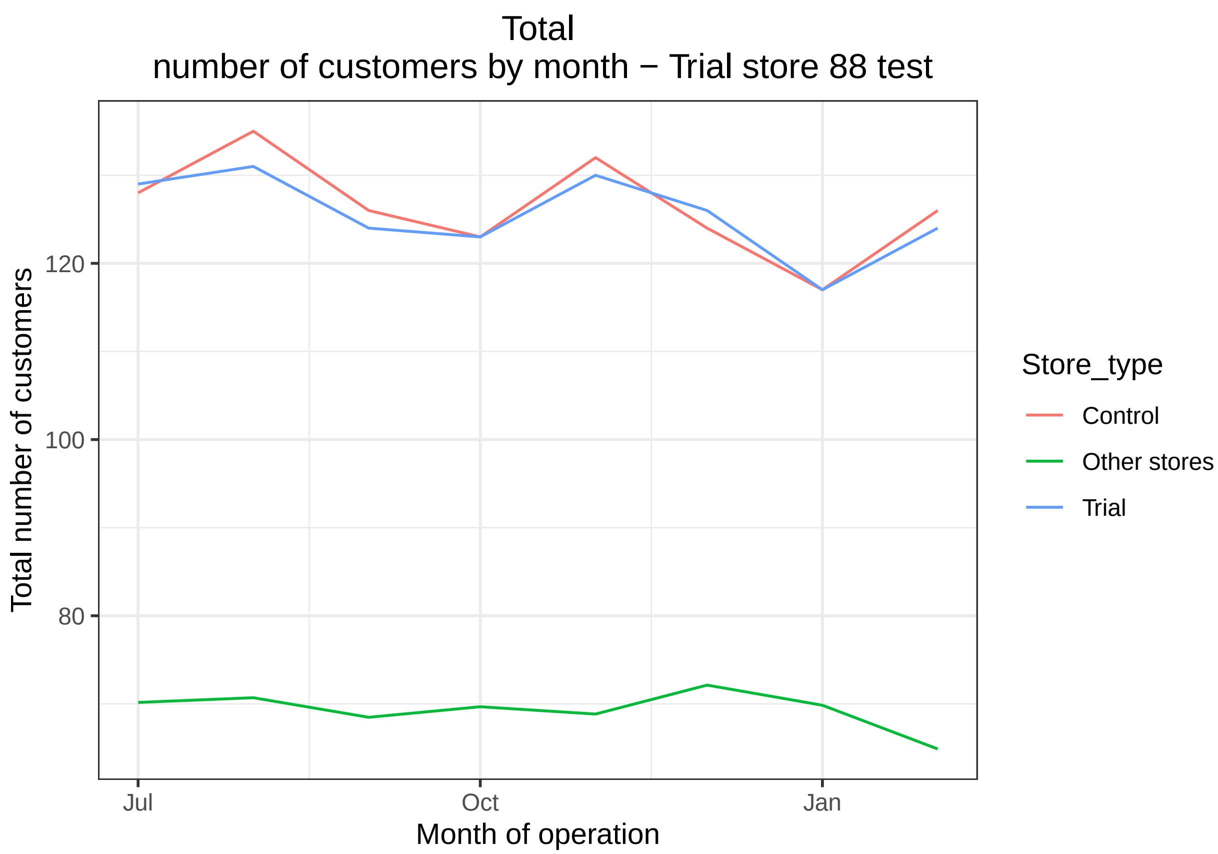
```
##    Store1 Store2 corr_measure.x mag_measure.x scoreNSales2 corr_measure.y
## 1:     88    237      0.3084792     0.9560757    0.6322774      0.9473262
##    mag_measure.y scoreNCust2 finalControlScore
## 1:     0.9875857    0.967456         0.7998667
```

```r
#store 237 as control for store 88

control_store <- 237
trial_store <- 88

measureOverTimeSales <- measureOverTime

measureOverTimeSales[ , YEARMONTH := as.numeric(YEARMONTH) ]

# Trial and control store total sales
# Over to you! Create new variables Store_type, totSales and TransactionMonth in
#the data table.

pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store,
                                                         "Trial", ifelse(STORE_NBR == control_store, "C
][, totSales := mean(totSales), by = c("YEARMONTH","Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
][Store_type %in% c("Trial", "Control"), ][YEARMONTH < 201903]

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

```
month - Trial Store 88 test")
```




```
#sales are trending in a similar way.Next, number of customers.
#Scaling factor again


measureOverTimeCusts <- measureOverTime

pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR ==
                                                          trial_store, "Trial",
                                                        ifelse(STORE_NBR == control_store,
                                                               "Control", "Other stores"))
][, noCustomers := mean(nCustomers), by =
    c("YEARMONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/%
                                            100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
][YEARMONTH < 201903 , ]

ggplot(pastCustomers, aes(TransactionMonth, noCustomers, color =
                            Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Total number of customers", title = "Total
 number of customers by month - Trial store 88 test")
```



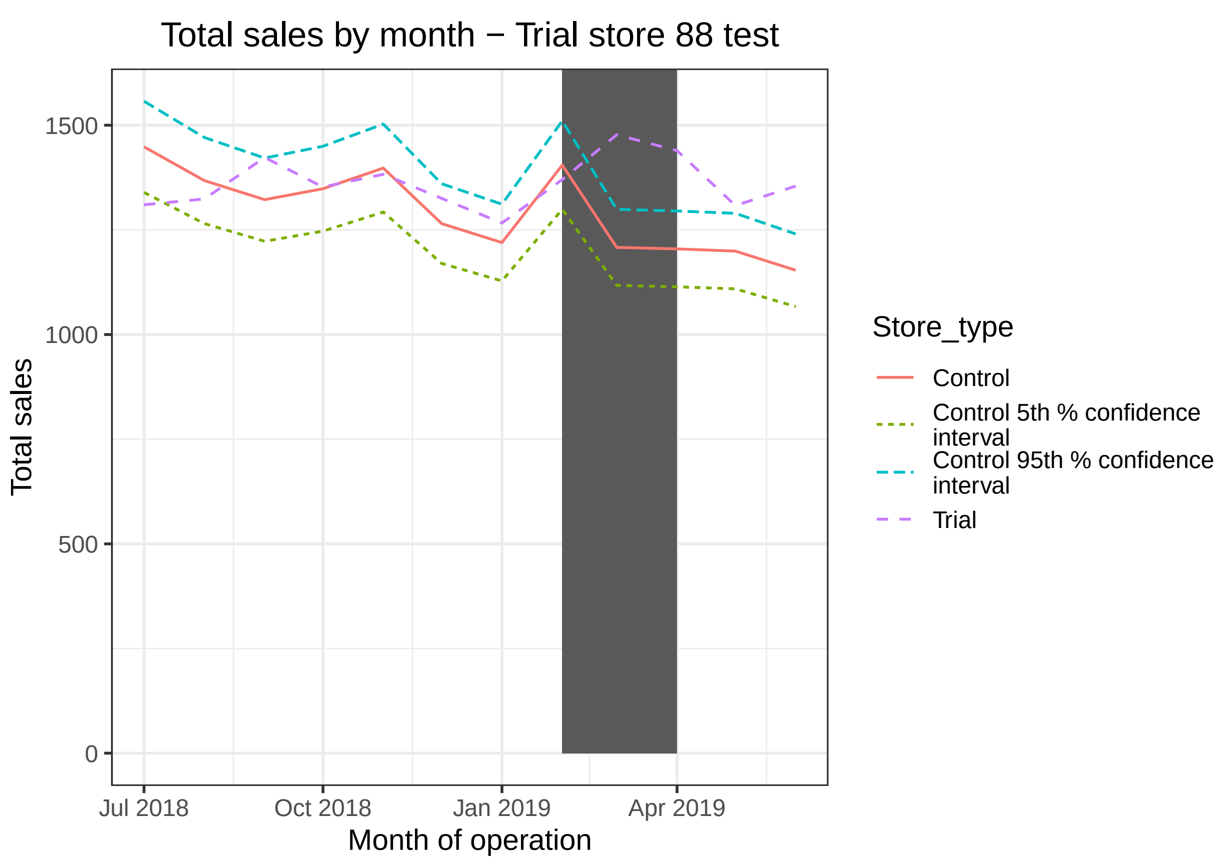

```
#


calingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
                                                  YEARMONTH < 201902, sum(totSales)]/
  preTrialMeasures[STORE_NBR == control_store & YEARMONTH < 201902, sum(totSales)]


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"))
]

# Apply the scaling factor

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


#Calculate the percentage difference between scaled control sales
#and trial sales
```

```
percentageDiff <- merge(measureOverTimeSales[STORE_NBR == trial_store,
                                             c("totSales", "YEARMONTH")],
                        scaledControlSales[ , c("controlSales", "YEARMONTH")],
                        by = "YEARMONTH"
)[, percentageDiff := abs(totSales-controlSales)/
    (0.5*(totSales+controlSales))]

#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

#Calculate the standard deviation of percentage differences during
#the pre-trial period.

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

degreesOfFreedom <- 7

percentageDiff[, tValue := (percentageDiff-0)/stdDev
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1,
sep = "-"), "%Y-%m-%d")
][YEARMONTH < 201905 & YEARMONTH > 201901, .(TransactionMonth,tValue)]
```

```
##    TransactionMonth   tValue
## 1:       2019-02-01 1.284168
## 2:       2019-03-01 4.714263
## 3:       2019-04-01 4.108247
```

```
#Trial and control store total sales

#Create a table with sales by store type and month.
#### Hint: We only need data for the trial and control store.


measureOverTimeSales <- measureOverTime

pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store,
                                                        "Trial", ifelse(STORE_NBR == control_store, "C
][, totSales := mean(totSales), by = c("YEARMONTH","Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/%
                                     100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
][Store_type %in% c("Trial", "Control"), ]

#Calculate the 5th and 95th percentile for control store sales.
#### Hint: The 5th and 95th percentiles can be approximated by using two standard
#deviations away from the mean.
#### Hint2: Recall that the variable stdDev earlier calculates standard deviation
#in percentages, and not dollar sales.

# 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"]

#Then, create a combined table with columns from pastSales,
#pastSales_Controls95 and pastSales_Controls5

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(aes(linetype = Store_type)) +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by month - Trial store 88 test
```



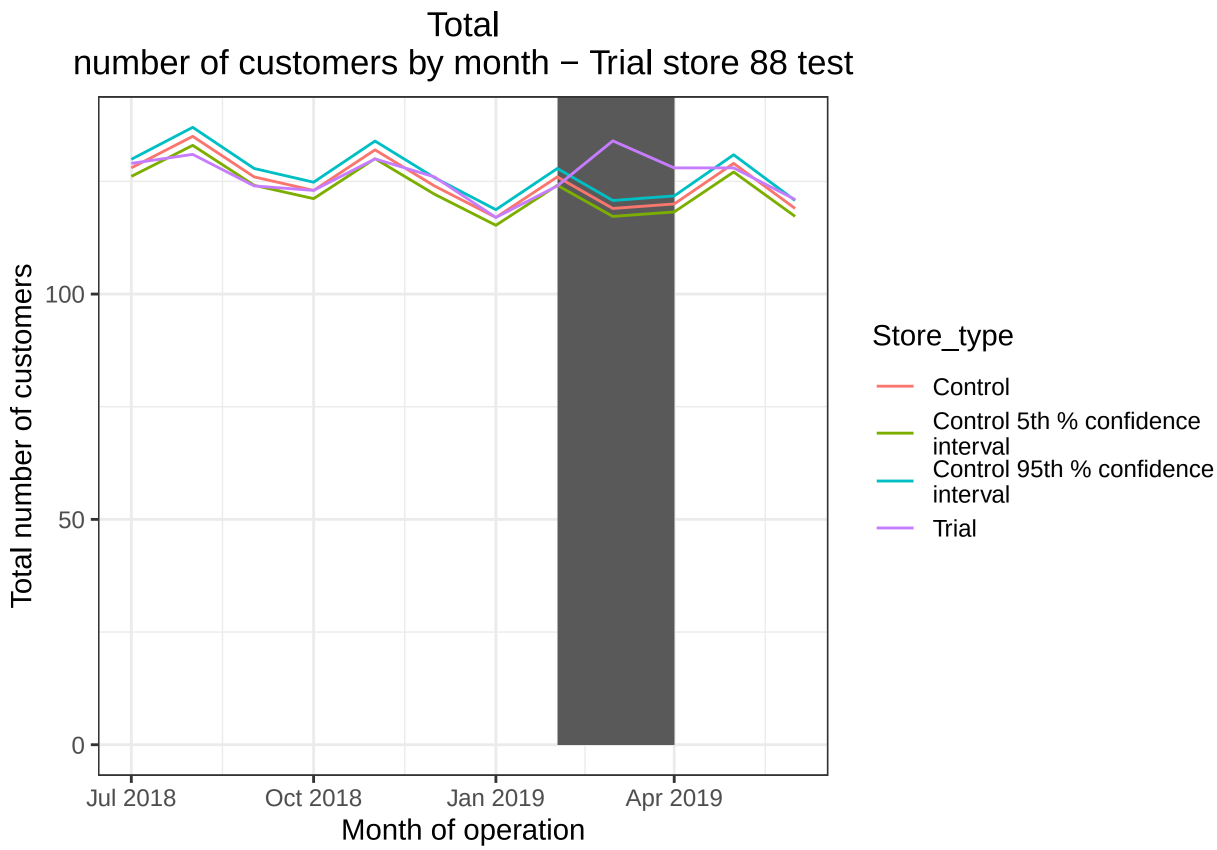

```
#### This would be a repeat of the steps before for total sales
#### Scale pre-trial control customers to match pre-trial trial store customers

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

```r
#### 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")],

                        measureOverTimeCusts[STORE_NBR == trial_store,
                                             c("nCustomers", "YEARMONTH")],
                        by = "YEARMONTH"
)[, percentageDiff :=
    abs(controlCustomers-nCustomers)/controlCustomers]

#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

percentageDiff[, tValue := (percentageDiff-0)/stdDev
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1,
sep = "-"), "%Y-%m-%d")
][YEARMONTH < 201905 & YEARMONTH > 201901, .(TransactionMonth,tValue)]
```

```
##    TransactionMonth    tValue
## 1:       2019-02-01  1.387456
## 2:       2019-03-01 17.873693
## 3:       2019-04-01  9.814423
```

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

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

```r
trialAssessment <- rbind(pastCustomers, pastCustomers_Controls95,
                         pastCustomers_Controls5)
#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, nCusts, 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 number of customers", title = "Total
number of customers by month - Trial store 88 test")
```



Conclusion

Good work! 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.