Quantium Task 2

Talib Izhar

```
#install.packages("tidyr")
#install.packages("data.table")
#install.packages("ggplot2")
```

Libraries

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.3.2
```

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.3.2
```

Data Import

```
#Importing the file from task 1

data <- fread("C:/Users/dexter/Documents/tlb docs/Data_Analytics/Forage/Merged_Data.csv")

#removing unnecessary columns we are going to work in this task.
data[,c('V1','PROD_NAME'):=NULL]
```

```
## Warning in `[.data.table`(data, , `:=`(c("V1", "PROD_NAME"), NULL)): Column
## 'V1' does not exist to remove
```

```
data[,.N,DATE][order(DATE)]
```

```
##            DATE   N
##   1: 2018-07-01 724
##   2: 2018-07-02 711
##   3: 2018-07-03 722
##   4: 2018-07-04 714
```

```
##   5: 2018-07-05 712
##  ---
## 360: 2019-06-26 723
## 361: 2019-06-27 709
## 362: 2019-06-28 730
## 363: 2019-06-29 745
## 364: 2019-06-30 744
```

Selecting control stores

–Trial period was from start of February 2019 to end of April 2019, store numbers 77, 86 and 88 are trial stores and client want control stores to be established stores that are operational for the entire observation period. Considering the monthly sales experience of each store. This can be broken down by: 1.total sales revenue 2.total number of customers 3.average number of transactions per customer First creating metrics and filter stores that are present during the trial period, however for this analysis, I'll be choosing the control store based on 2 metrics only i.e., Total Sales and Number of Customers.

```
###Creating a new column YEAR_MONTH in the data
data[,YEAR_MONTH:= year(DATE)*100 + month(DATE)]
#### Next,  define the measure calculations to use during the analysis for each month and store.
MeasurebyMonth <- data[, .(totSales=sum(TOT_SALES),
                           CustNum= uniqueN(CARD_NBR),
                          TxnperCust= uniqueN(TXN_ID)/uniqueN(CARD_NBR),
                          ChipsperTrnsc= sum(PROD_QTY)/uniqueN(TXN_ID),
                          avgPriceperUnit= sum(TOT_SALES)/sum(PROD_QTY)),
                       .(STORE_NBR,YEAR_MONTH)][order(YEAR_MONTH)]
```

Filter to the pre-trial period and stores with full observation periods We have data from 2018-07-01 to 2019-06-30 and we have divided for each month in MeasurebyMonth variable. So now we will filter stores that appear in all the 12 months and stores that appears before trial period.

```
StoreWithFullobs <- unique(MeasurebyMonth[, .N, STORE_NBR][N==12,STORE_NBR])

preTrialMeasures <- MeasurebyMonth[YEAR_MONTH < '2019-02' & STORE_NBR %in% StoreWithFullobs]
```

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. Writing a function for this.

```
calCorr <- function(inputTable, metricCol,storeComparison) {
  calcorrTable= data.table(store1= numeric(),store2=numeric(),corr_measure=numeric())
   storeNum <- unique(inputTable[,STORE_NBR])
    for (i in storeNum) {
      CalctdMeasure = data.table("store1" =storeComparison,
                                 "store2" = i,
                                 "corr_measure" =cor(inputTable[STORE_NBR==storeComparison,
                                                                eval(metricCol)],
                                                     inputTable[STORE_NBR == i,
                                                      eval(metricCol)]))

  calcorrTable <- rbind(calcorrTable, CalctdMeasure)
    }
   return(calcorrTable)
```

```
}
```

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. Writing function for this.

```
calculateMagnitudeDistance <- function(inputTable, metricCol, storeComparison) {
calcDistTable = data.table(store1= numeric(),store2=numeric(),YEAR_MONTH=numeric(), measure = numeric())

    StoreNum <- unique(inputTable[,STORE_NBR])

    for(i in StoreNum) {
    CalcMeasure = data.table("store1" = storeComparison
                             ,"store2" = i
                             , YEAR_MONTH = inputTable[STORE_NBR == storeComparison, YEAR_MONTH]
                             , measure = abs(inputTable[STORE_NBR == storeComparison, eval(metricCol)]
                             - inputTable[STORE_NBR == i,eval(metricCol)])
                            )

    calcDistTable <- rbind(calcDistTable,CalcMeasure)

}
###Standardize magnitude
 minMaxDist <- calcDistTable[, .(minDist = min(measure), maxDist = max(measure)), by =
              distTable <- merge(calcDistTable, minMaxDist, by = c("store1", "YEAR_MONTH"))
              distTable[, magnitudeMeasure := 1 - (measure - minDist)/(maxDist - minDist)]

             finalDistTable <- distTable[, .(mag_measure = mean(magnitudeMeasure)), by =
                               .(store1, store2)]
return(finalDistTable)

}
```

Using functions

```
trial_store <- 77
corr_nSales <- calCorr(preTrialMeasures, quote(totSales),trial_store )
corr_nSales[order(-corr_measure)]
```

```
##      store1 store2 corr_measure
##   1:     77     77    1.0000000
##   2:     77     71    0.9443031
##   3:     77     63    0.9322880
##   4:     77    119    0.8859164
##   5:     77    233    0.8699296
##  ---
## 256:     77    136   -0.7518980
## 257:     77    186   -0.7525036
## 258:     77     97   -0.7714027
## 259:     77     75   -0.8278233
## 260:     77    134   -0.8638656
```

for customers

```
corr_nCustomers <- calCorr(preTrialMeasures, quote(CustNum),trial_store )
corr_nCustomers[order(-corr_measure)]
```

```
##      store1 store2 corr_measure
##   1:     77     77    1.0000000
##   2:     77    233    0.9915849
##   3:     77    119    0.9803236
##   4:     77    254    0.9618198
##   5:     77     71    0.8966439
##  ---
## 256:     77    124   -0.7054507
## 257:     77     75   -0.7493631
## 258:     77    138   -0.7549207
## 259:     77     86   -0.7733576
## 260:     77    114   -0.8299531
```

Functions for calculating magnitude

```
magnitude_Sales <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales), trial_store)
magnitude_Customers <- calculateMagnitudeDistance(preTrialMeasures, quote(CustNum), trial_store)
```

Combining all scores

```
Corr_Weight <- 0.5
Score_Sales <- merge(corr_nSales,magnitude_Sales, by = c("store1", "store2"))[ , scoreNsales:= (corr_mea
Score_Customers <- merge(corr_nCustomers,magnitude_Customers, by = c("store1", "store2"))[ , scoreNcust
Score_Sales[order(-scoreNsales)]
```

```
##      store1 store2 corr_measure mag_measure scoreNsales
##   1:     77     77    1.0000000   1.0000000   1.0000000
##   2:     77    233    0.8699296   0.9862040   0.9280668
##   3:     77     17    0.8492859   0.8810210   0.8651535
##   4:     77    115    0.7432815   0.9362268   0.8397542
##   5:     77     41    0.7101164   0.9637978   0.8369571
##  ---
## 256:     77     95   -0.5226280   0.2932133  -0.1147073
## 257:     77     97   -0.7714027   0.4892063  -0.1410982
## 258:     77    201   -0.6644896   0.2783684  -0.1930606
## 259:     77     88   -0.6475019   0.1395720  -0.2539649
## 260:     77     75   -0.8278233   0.3122398  -0.2577918
```

```
Score_Customers[order(-scoreNcustomer)]
```

```
##      store1 store2 corr_measure mag_measure scoreNcustomer
##   1:     77     77    1.0000000   1.0000000      1.0000000
##   2:     77    233    0.9915849   0.9915689      0.9915769
##   3:     77    254    0.9618198   0.9317036      0.9467617
##   4:     77     41    0.7777138   0.9720959      0.8749049
##   5:     77    121    0.8043107   0.9388427      0.8715767
##  ---
## 256:     77    125   -0.5647432   0.2766016     -0.1440708
```

```
## 257:     77     86   -0.7733576   0.4181416     -0.1776080
## 258:     77    138   -0.7549207   0.3948138     -0.1800534
## 259:     77     75   -0.7493631   0.3466035     -0.2013798
## 260:     77    114   -0.8299531   0.3515302     -0.2392114
```

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 drivers by merging sales scores and customer scores
score_Control <- merge(Score_Sales,Score_Customers, by = c("store1", "store2"))
#adding a new column finalcontrolstore by finding the average
score_Control[, finalControlStore := scoreNsales * 0.5 + scoreNcustomer *0.5]
score_Control[order(-finalControlStore)]
```

```
##      store1 store2 corr_measure.x mag_measure.x scoreNsales corr_measure.y
##   1:     77     77      1.0000000     1.0000000  1.00000000      1.0000000
##   2:     77    233      0.8699296     0.9862040  0.92806678      0.9915849
##   3:     77     41      0.7101164     0.9637978  0.83695710      0.7777138
##   4:     77     17      0.8492859     0.8810210  0.86515347      0.7239437
##   5:     77    115      0.7432815     0.9362268  0.83975415      0.7525964
##  ---
## 256:     77    125     -0.4103731     0.2986862 -0.05584343     -0.5647432
## 257:     77    201     -0.6644896     0.2783684 -0.19306064     -0.2595379
## 258:     77    138     -0.5653179     0.5012631 -0.03202742     -0.7549207
## 259:     77    114     -0.5637157     0.4587724 -0.05247165     -0.8299531
## 260:     77     75     -0.8278233     0.3122398 -0.25779175     -0.7493631
##      mag_measure.y scoreNcustomer finalControlStore
##   1:     1.0000000    1.000000000        1.00000000
##   2:     0.9915689    0.991576866        0.95982182
##   3:     0.9720959    0.874904859        0.85593098
##   4:     0.9629785    0.843461109        0.85430729
##   5:     0.9737034    0.863149897        0.85145202
##  ---
## 256:     0.2766016   -0.144070798       -0.09995711
## 257:     0.2451040   -0.007216978       -0.10013881
## 258:     0.3948138   -0.180053448       -0.10604044
## 259:     0.3515302   -0.239211426       -0.14584154
## 260:     0.3466035   -0.201379773       -0.22958576
```

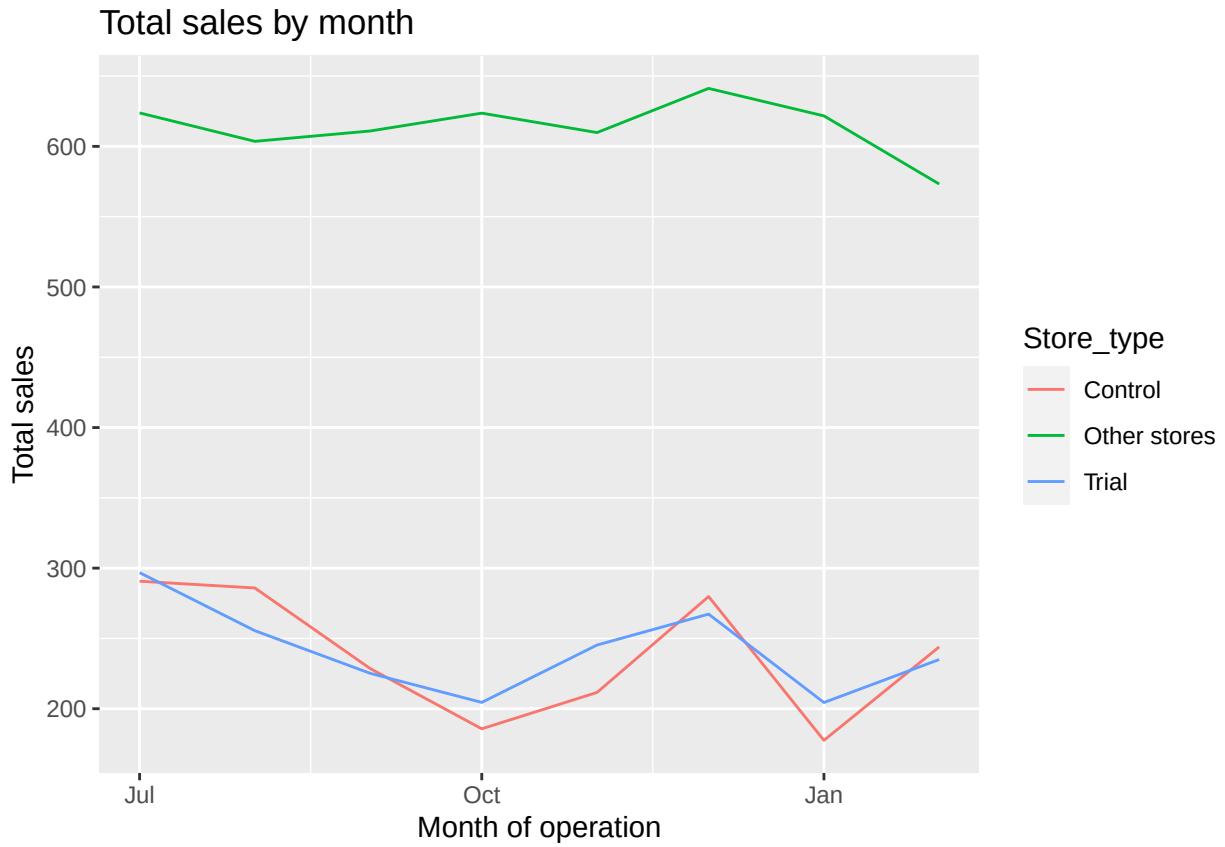
```
#Finding the highest scored sore
control_store <- score_Control[store1 == trial_store,][order(-finalControlStore)][2,store2]
control_store
```

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

Store 233 is the most related store to trial store 77. Now checking if drivers are similar visually in the period before trial. First sales.

```
MeasurebyMonthSales <-MeasurebyMonth

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

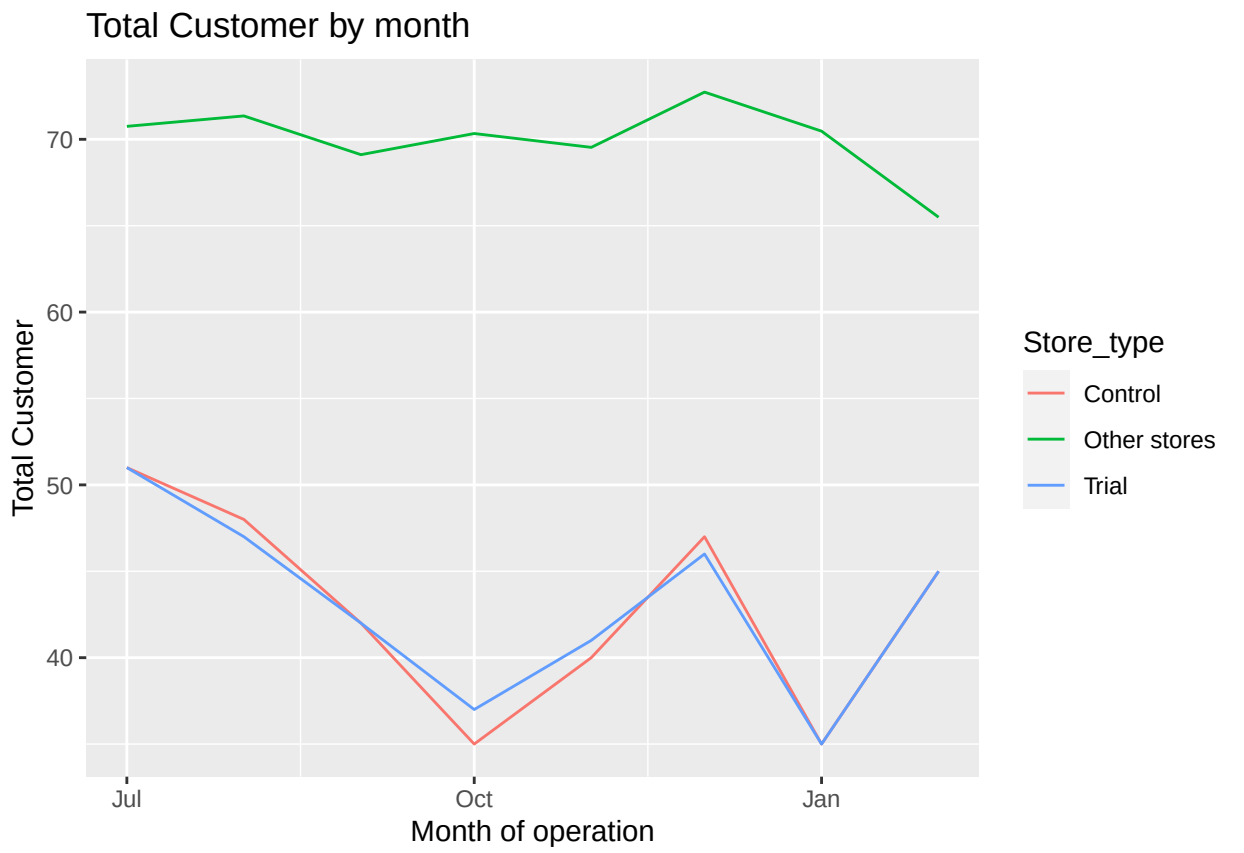
```
                                        ifelse(STORE_NBR == control_store,
"Control", "Other stores"))
                              ][, totSales := mean(totSales), by = c("YEAR_MONTH",
"Store_type")
                              ][, TransactionMonth := as.Date(paste(YEAR_MONTH %/%
100, YEAR_MONTH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



Now, checking for customers.

```
MeasurebyMonthCustomer <-MeasurebyMonth

pastCustomer <- MeasurebyMonthCustomer[, Store_type := ifelse(STORE_NBR == trial_store,
"Trial",
                                        ifelse(STORE_NBR == control_store,
"Control", "Other stores"))
                              ][, totCustomer := mean(CustNum), by = c("YEAR_MONTH",
"Store_type")
                              ][, TransactionMonth := as.Date(paste(YEAR_MONTH %/%
100, YEAR_MONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
                              ][YEAR_MONTH < 201903 , ]
ggplot(pastCustomer, aes(TransactionMonth, totCustomer, color = Store_type)) +
  geom_line() +
```

```
  labs(x = "Month of operation", y = "Total Customer", title = "Total Customer by month")
```



Assesment of Trial Now we'll 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 psre-trial trial store sales
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
YEAR_MONTH < 201902, sum(totSales)]/preTrialMeasures[STORE_NBR == control_store &
YEAR_MONTH < 201902, sum(totSales)]
#### Apply the scaling factor
MeasurebyMonthSales <- MeasurebyMonth
scaledControlSales <- MeasurebyMonthSales[STORE_NBR == control_store, ][ ,
controlSales := totSales * scalingFactorForControlSales]
```

Calculating percentage difference between the scaled control sales and the trial store's sales during the trial period.

```
percentageDiff <- merge(scaledControlSales[, c("YEAR_MONTH","controlSales")],
                        MeasurebyMonth[STORE_NBR == trial_store, c("totSales","YEAR_MONTH")],
                    by = "YEAR_MONTH")[, percentageDiff := abs(controlSales-totSales)/controlSales]
```

Checking if the difference is significant.

```
#### Here null hypothesis is that the trial period is the same as the pre-trial
#period, taking standard deviation based on the scaled percentage difference
#in the pre-trial period
stdDev <- sd(percentageDiff[YEAR_MONTH < 201902 , percentageDiff])
#### Since, 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.
### Calculating the t-values for the trial months. After that, find the
#95th percentile of the t distribution with the appropriate degrees of freedom
#to check whether the hypothesis is statistically significant
percentageDiff[, tvalue := (percentageDiff-0)/stdDev][, TransactionMonth := as.Date(paste(YEAR_MONTH %/%
100, YEAR_MONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
                              ][YEAR_MONTH < 201905 & YEAR_MONTH > 201901, .(TransactionMonth, tvalue) ]
```

```
##    TransactionMonth     tvalue
## 1:       2019-02-01  0.8080315
## 2:       2019-03-01  6.9834334
## 3:       2019-04-01 11.6799494
```

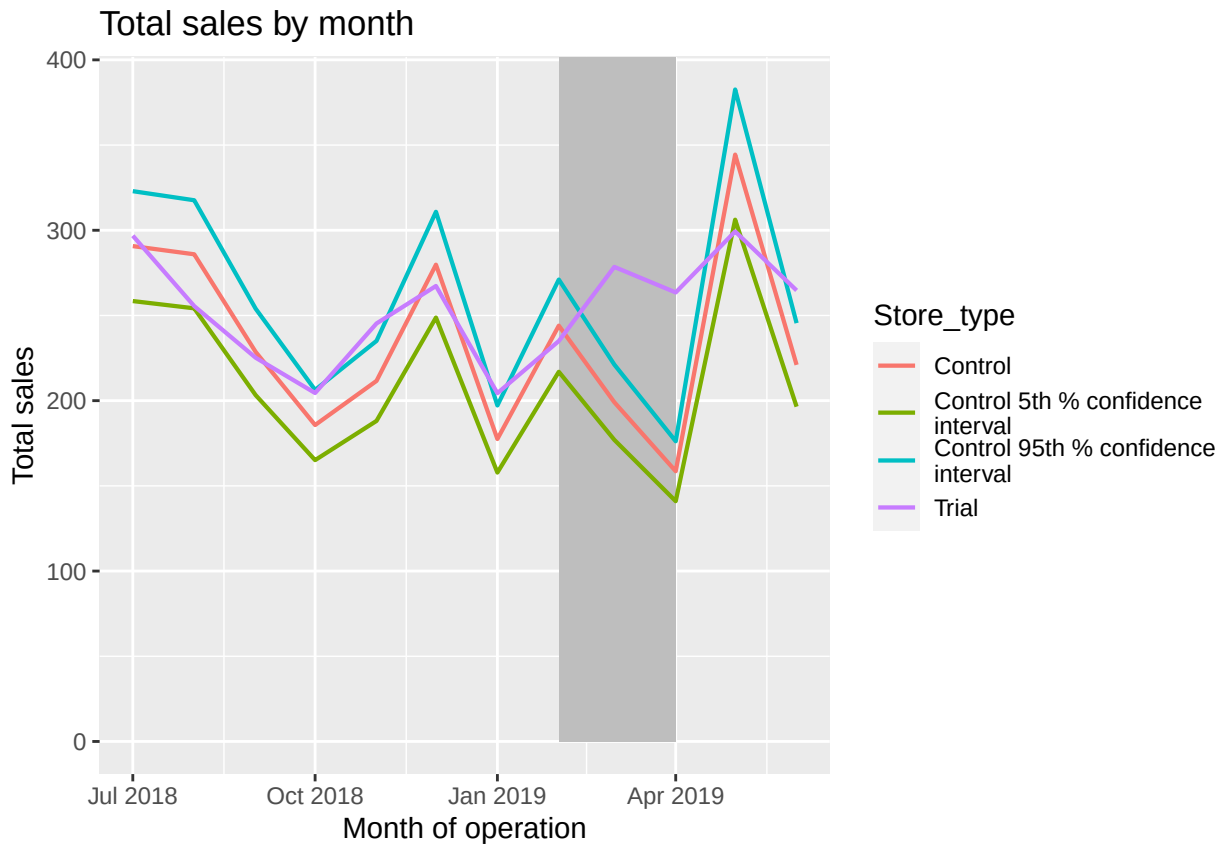
```
#### Find the 95th percentile of the t distribution with the appropriate
#### degrees of freedom to compare against
qt(0.95, df = degreesOfFreedom)
```

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

t-value is larger than 95th percentile of t-distribution for month March & April. This means increase in sales in trial stores in March & April is greater than in control stores. 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.

```
MeasurebyMonthSales <- MeasurebyMonth
#### Trial and control store total sales
pastSales <- MeasurebyMonthSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
ifelse(STORE_NBR == control_store, "Control", "Other stores"))
][, totSales := mean(totSales), by = c("YEAR_MONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEAR_MONTH %/% 100, YEAR_MONTH %% 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 graph
```

```
ggplot(trialAssessment, aes(TransactionMonth, totSales, color = Store_type)) +
geom_rect(data = trialAssessment[ YEAR_MONTH < 201905 & YEAR_MONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
ymin = 0 , ymax =Inf, color = NULL), show.legend = FALSE, fill = "GREY") +
geom_line( linewidth=0.75) +
labs(x = "Month of operation", y = "Total sales", title = "Total sales by month")
```



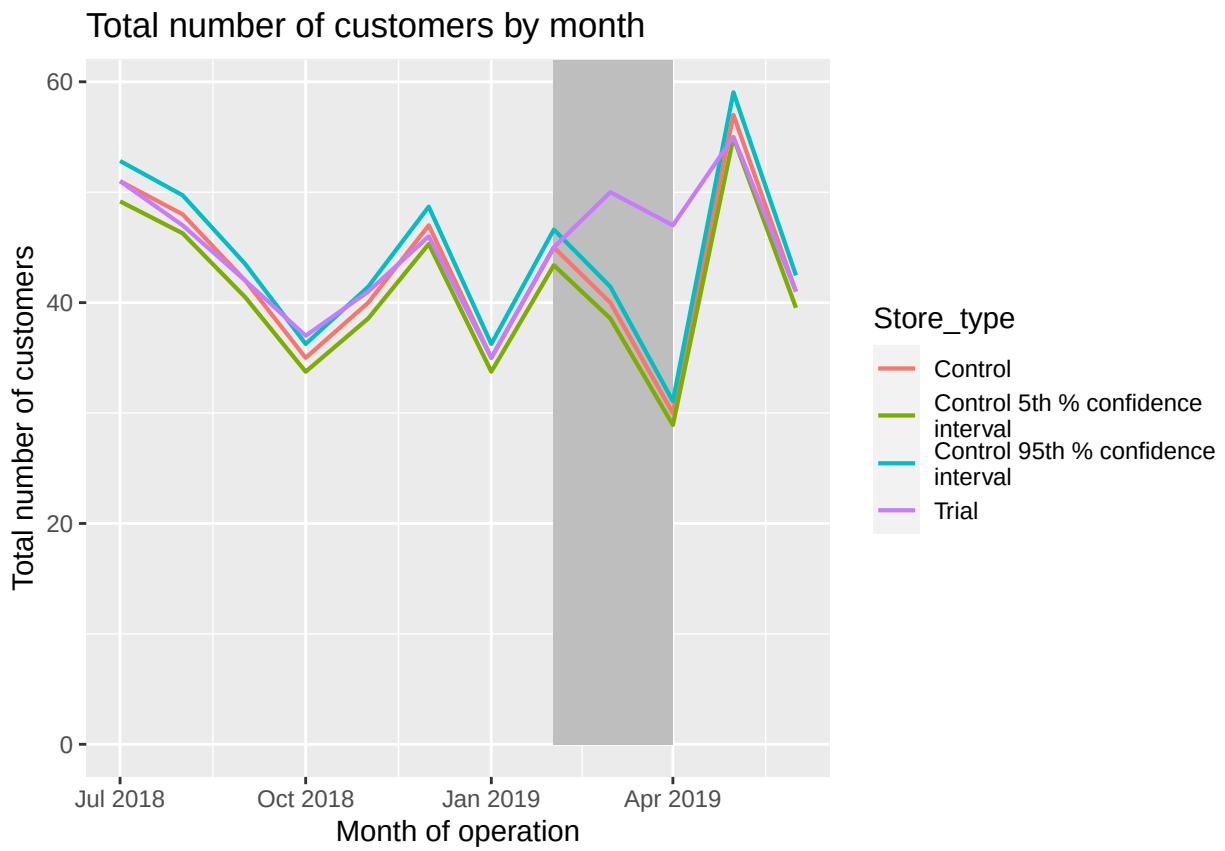
Trial in store 77 is significantly different to its control store in the trial period as the trial store performance lies outside the 5% to 95% confidence interval of the control store in two of the three trial months. Now assessing this for number of customers.

```
#### Scale pre-trial control customers to match pre-trial trial store customers
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store &
YEAR_MONTH < 201902, sum(CustNum)]/preTrialMeasures[STORE_NBR == control_store &
YEAR_MONTH < 201902, sum(CustNum)]
#### Apply the scaling factor
MeasurebyMonthCustomer <- MeasurebyMonth
scaledControlCustomer <- MeasurebyMonthCustomer[STORE_NBR == control_store, ][ ,
controlCustomer := CustNum * scalingFactorForControlCust]

percentageDiff <- merge(scaledControlCustomer[, c("YEAR_MONTH","controlCustomer")],
                        MeasurebyMonth[STORE_NBR == trial_store, c("CustNum","YEAR_MONTH")],
                    by = "YEAR_MONTH")[, percentageDiff := abs(controlCustomer-CustNum)/controlCustome
```

Check if difference is significant visually.

```
stdDev <- sd(percentageDiff[YEAR_MONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7
#### Trial and control store number of customers
pastCustomers <- MeasurebyMonthCustomer[, nCusts := mean(CustNum), by =
c("YEAR_MONTH", "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 everything into one nice graph.
#### 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[ YEAR_MONTH < 201905 & YEAR_MONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0 ,
ymax = Inf, color = NULL), show.legend = FALSE, fill = "GREY") +
  geom_line(size=0.75) + labs(x = "Month of operation", y = "Total number of customers", title = "Total
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



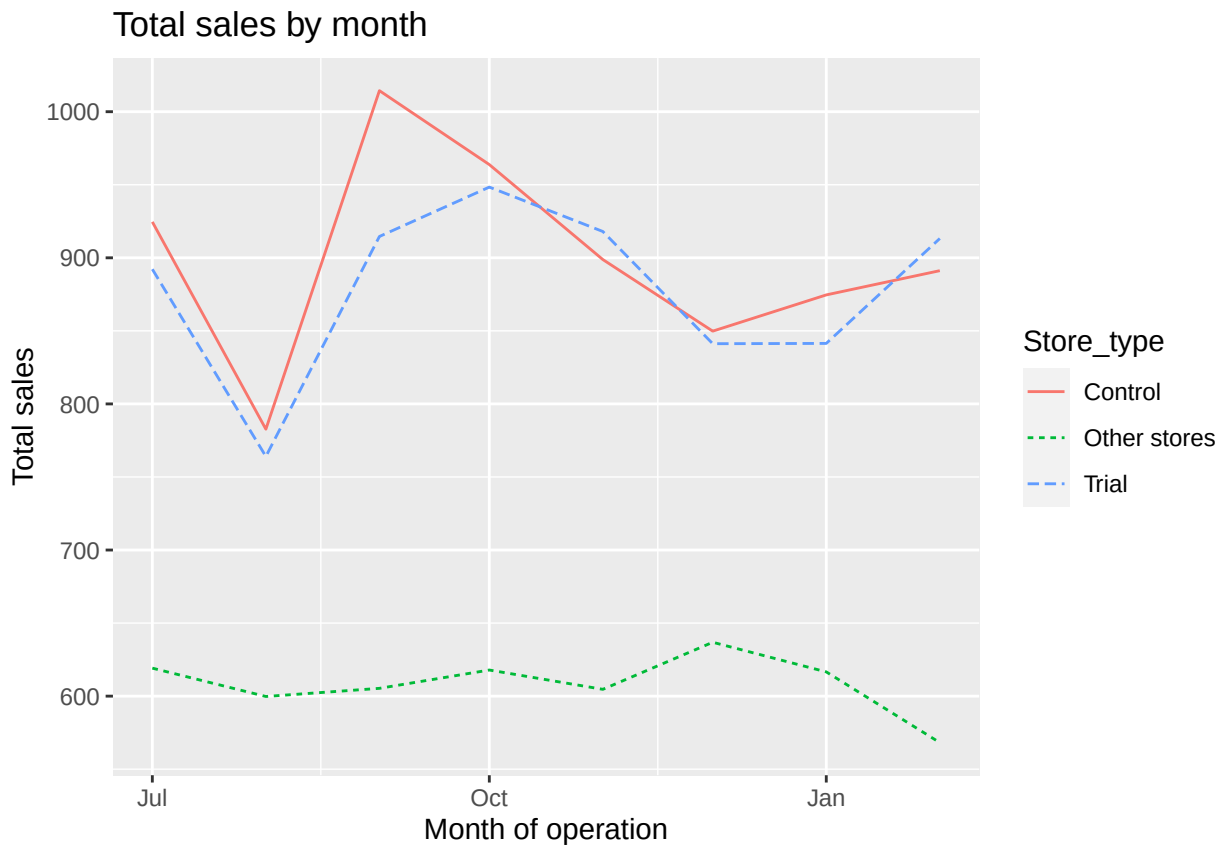
The trial is outside the 5th and 95th % confidence interval for customers as well. Finding control stores & assessing impact of trial store for each of other two trial store. ## Trial Store 86

```
### Reassigning the metrics to the variable.
MeasurebyMonth <- data[, .(totSales=sum(TOT_SALES),
                           CustNum= uniqueN(CARD_NBR),
                          TxnperCust= uniqueN(TXN_ID)/uniqueN(CARD_NBR),
                          ChipsperTrnsc= sum(PROD_QTY)/uniqueN(TXN_ID),
                          avgPriceperUnit= sum(TOT_SALES)/sum(PROD_QTY)),
                      .(STORE_NBR,YEAR_MONTH)][order(YEAR_MONTH)]
trial_store <- 86
corr_nSales <- calCorr(preTrialMeasures, quote(totSales),trial_store )
corr_nCustomers <- calCorr(preTrialMeasures, quote(CustNum),trial_store )
magnitude_Sales <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales), trial_store)
magnitude_Customers <- calculateMagnitudeDistance(preTrialMeasures, quote(CustNum), trial_store)

Corr_Weight <- 0.5
Score_Sales <- merge(corr_nSales,magnitude_Sales, by = c("store1", "store2"))[ , scoreNsales:= (corr_mea
Score_Customers <- merge(corr_nCustomers,magnitude_Customers, by = c("store1", "store2"))[ , scoreNcusto

score_Control <- merge(Score_Sales, Score_Customers, by = c("store1","store2"))
score_Control[, finalControlScore := scoreNsales * 0.5 + scoreNcustomer * 0.5]

control_store <- score_Control[store1 == trial_store,
][order(-finalControlScore)][2, store2]
control_store
```

```
## [1] 155
```

Let's check visually if the drivers are indeed similar in the period before the trial. We'll look at total sales first.

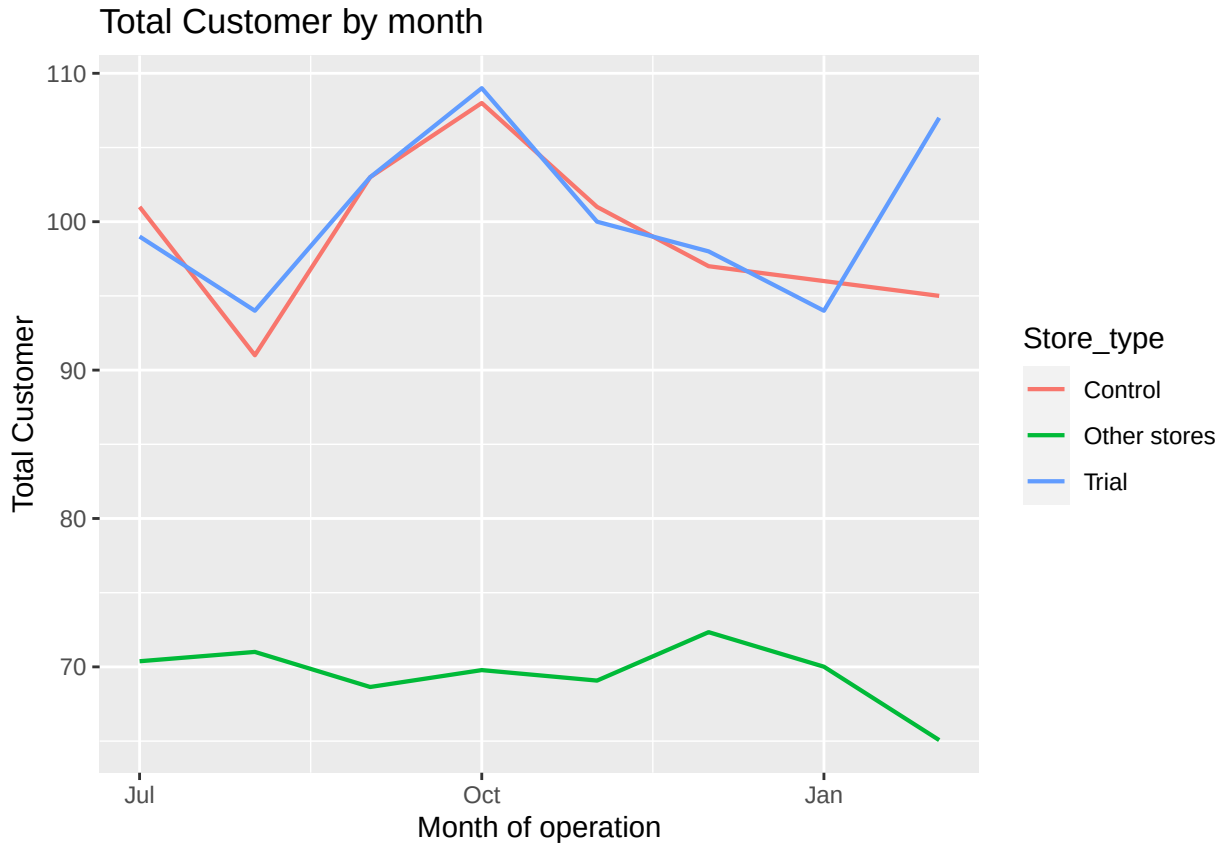
```
MeasurebyMonthSales <-MeasurebyMonth

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



Sales are trending in similar way. Now checking number of customers.

```
MeasurebyMonthCustomer <-MeasurebyMonth

pastCustomer <- MeasurebyMonthCustomer[, Store_type := ifelse(STORE_NBR == trial_store,
"Trial",
                                       ifelse(STORE_NBR == control_store,
"Control", "Other stores"))
```

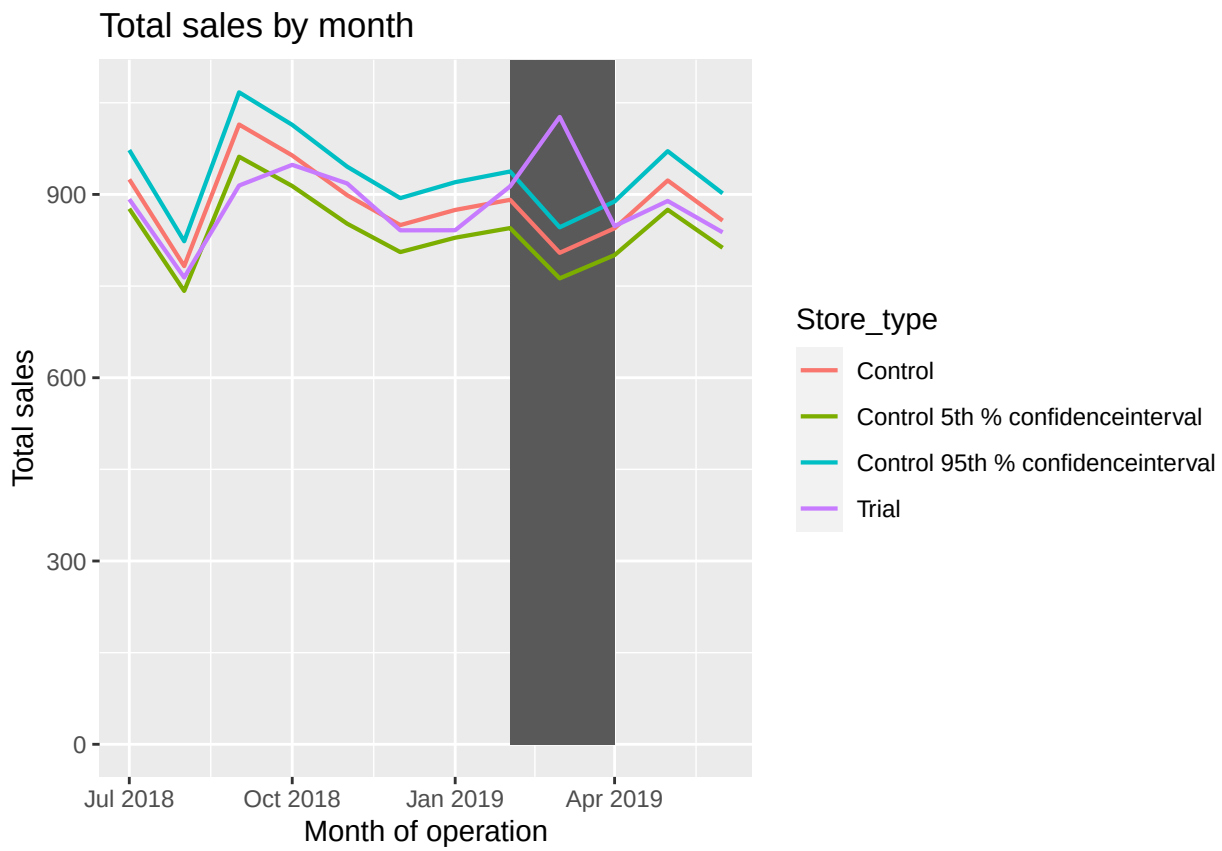
```
                                ][, totCustomer := mean(CustNum), by = c("YEAR_MONTH",
"Store_type")
                                ][, TransactionMonth := as.Date(paste(YEAR_MONTH %/%
100, YEAR_MONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
                                ][YEAR_MONTH < 201903 , ]

ggplot(pastCustomer, aes(TransactionMonth, totCustomer, color = Store_type)) +
  geom_line(size=0.75) +
  labs(x = "Month of operation", y = "Total Customer", title = "Total Customer by month")
```



The trend for number of customers is also similar, now assessing the impact of trial on sales.

```
#### Scale pre-trial control sales to match pre-trial trial store sales
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
YEAR_MONTH < 201902, sum(totSales)]/preTrialMeasures[STORE_NBR == control_store &
YEAR_MONTH < 201902, sum(totSales)]
#### Apply the scaling factor
MeasurebyMonthSales <- MeasurebyMonth
scaledControlSales <- MeasurebyMonthSales[STORE_NBR == control_store, ][,controlSales := totSales * sca

###Calculating percentage difference between the scaled control sales and the trial store's
### sales during the trial period.
percentageDiff <- merge(scaledControlSales[, c("YEAR_MONTH","controlSales")],
                        MeasurebyMonth[STORE_NBR == trial_store, c("totSales","YEAR_MONTH")],
                    by = "YEAR_MONTH")[, percentageDiff := abs(controlSales-totSales)/controlSales]
```

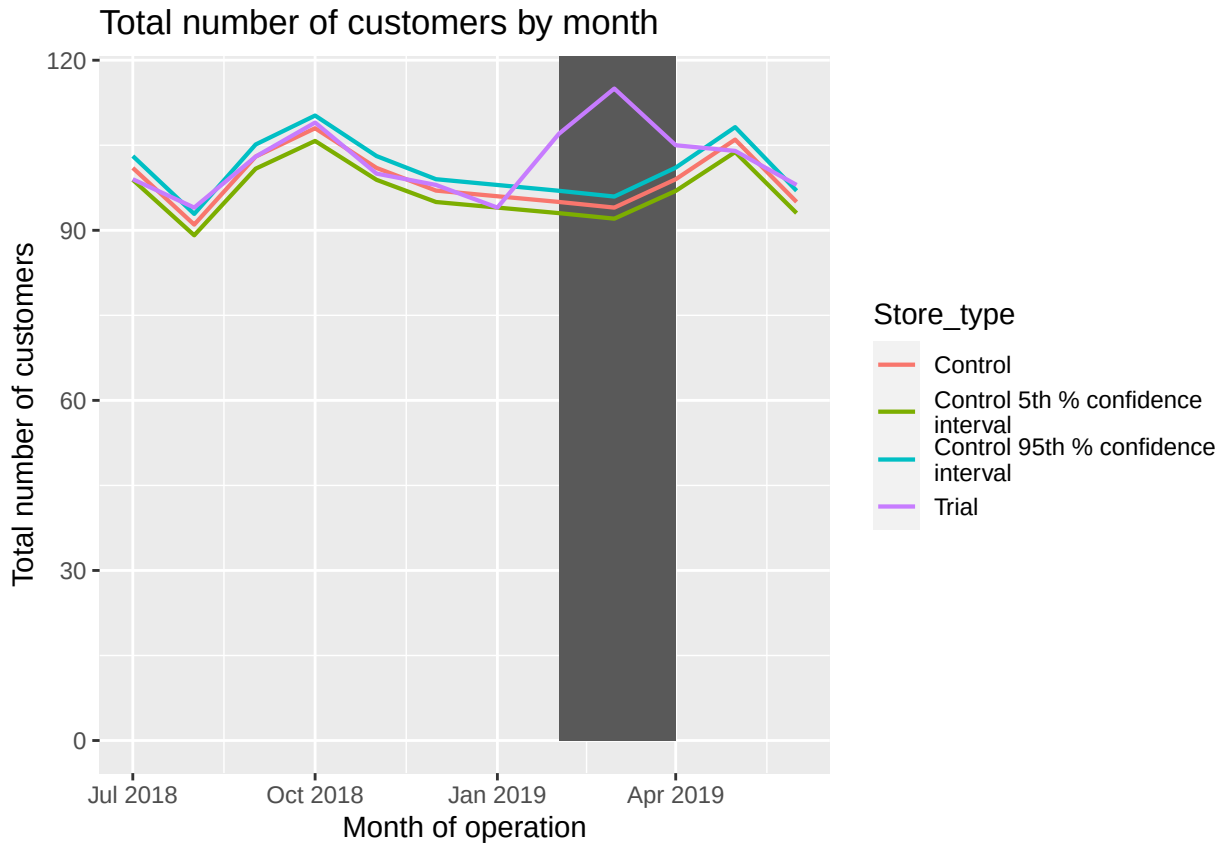
```r
### Standard deviation of percentage difference during pre-trial period
stdDev <- sd(percentageDiff[YEAR_MONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7

MeasurebyMonthSales <- MeasurebyMonth
#### Trial and control store total sales
pastSales <- MeasurebyMonthSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
ifelse(STORE_NBR == control_store, "Control", "Other stores"))
][, totSales := mean(totSales), by = c("YEAR_MONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEAR_MONTH %/% 100, YEAR_MONTH %% 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 % confidenceinterval"]
#### Control store 5th percentile
pastSales_Controls5 <- pastSales[Store_type == "Control",
][, totSales := totSales * (1 - stdDev * 2)
][, Store_type := "Control 5th % confidenceinterval"]
trialAssessment <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)
#### Plotting these in one graph
ggplot(trialAssessment, aes(TransactionMonth, totSales, color = Store_type)) +
geom_rect(data = trialAssessment[ YEAR_MONTH < 201905 & YEAR_MONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
ymin = 0 , ymax =Inf, color = NULL), show.legend = FALSE) +
geom_line(size=0.75) +
labs(x = "Month of operation", y = "Total sales", title = "Total sales by month")
```



Trial in store 86 is not significantly different to its control store as performance lies inside 5% to 95% confidence interval except for one month. Now analyzing impact of trial on number of customers.

```
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store &
YEAR_MONTH < 201902, sum(CustNum)]/preTrialMeasures[STORE_NBR == control_store &
YEAR_MONTH < 201902, sum(CustNum)]
#### Apply the scaling factor
MeasurebyMonthCustomer <- MeasurebyMonth
scaledControlCustomer <- MeasurebyMonthCustomer[STORE_NBR == control_store, ][ ,
controlCustomer := CustNum * scalingFactorForControlCust]

percentageDiff <- merge(scaledControlCustomer[, c("YEAR_MONTH","controlCustomer")],
                        MeasurebyMonth[STORE_NBR == trial_store, c("CustNum","YEAR_MONTH")],
                    by = "YEAR_MONTH")[, percentageDiff := abs(controlCustomer-CustNum)/controlCustome
####
stdDev <- sd(percentageDiff[YEAR_MONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7
#### Trial and control store number of customers
pastCustomers <- MeasurebyMonthCustomer[, nCusts := mean(CustNum), by =
c("YEAR_MONTH", "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 everything into one nice graph.
#### 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[ YEAR_MONTH < 201905 & YEAR_MONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0 ,
ymax = Inf, color = NULL), show.legend = FALSE) +
  geom_line(size=0.75) + labs(x = "Month of operation", y = "Total number of customers", title = "Total
```



The number of customers are higher in all three months, meaning there was an impact on number of customers during trial period but the sales were not relatively higher. ## Trial Store 88

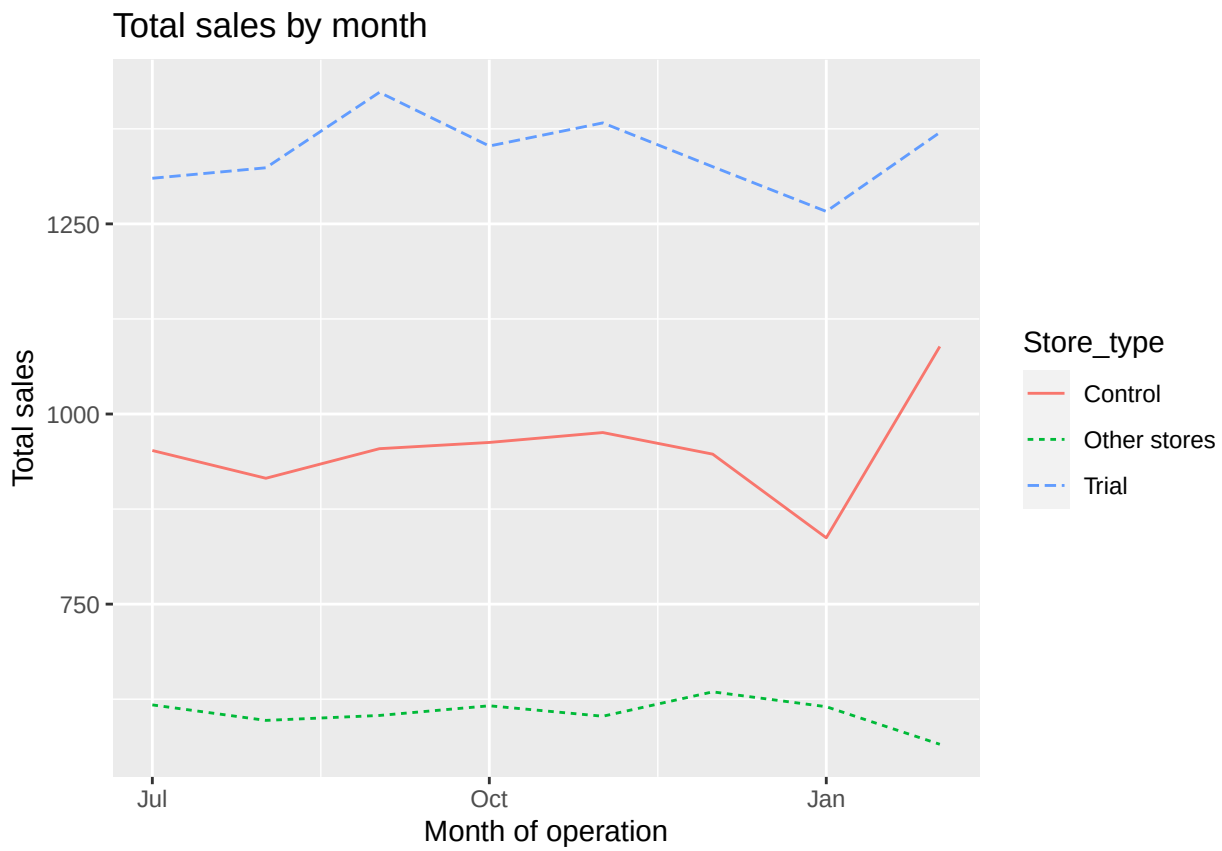
```
### Reassigning the metrics to the variable.
MeasurebyMonth <- data[, .(totSales=sum(TOT_SALES),
                           CustNum= uniqueN(CARD_NBR),
                          TxnperCust= uniqueN(TXN_ID)/uniqueN(CARD_NBR),
                          ChipsperTrnsc= sum(PROD_QTY)/uniqueN(TXN_ID),
                          avgPriceperUnit= sum(TOT_SALES)/sum(PROD_QTY)),
                      .(STORE_NBR,YEAR_MONTH)][order(YEAR_MONTH)]
```

```
trial_store <- 88
corr_nSales <- calCorr(preTrialMeasures, quote(totSales),trial_store )
corr_nCustomers <- calCorr(preTrialMeasures, quote(CustNum),trial_store )

magnitude_Sales <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales), trial_store)
magnitude_Customers <- calculateMagnitudeDistance(preTrialMeasures, quote(CustNum), trial_store)

Corr_Weight <- 0.5
Score_Sales <- merge(corr_nSales,magnitude_Sales, by = c("store1", "store2"))[ , scoreNsales:= (corr_mea
Score_Customers <- merge(corr_nCustomers,magnitude_Customers, by = c("store1", "store2"))[ , scoreNcusto

#combine drivers by merging sales scores and customer scores
score_Control <- merge(Score_Sales,Score_Customers, by = c("store1", "store2"))
#adding a new column
score_Control[, finalControlStore := scoreNsales * 0.5 + scoreNcustomer * 0.5]
#Finding the highest scored sore
control_store <- score_Control[store1 == trial_store,][order(-finalControlStore)][2,store2]
control_store
```

```
## [1] 178
```

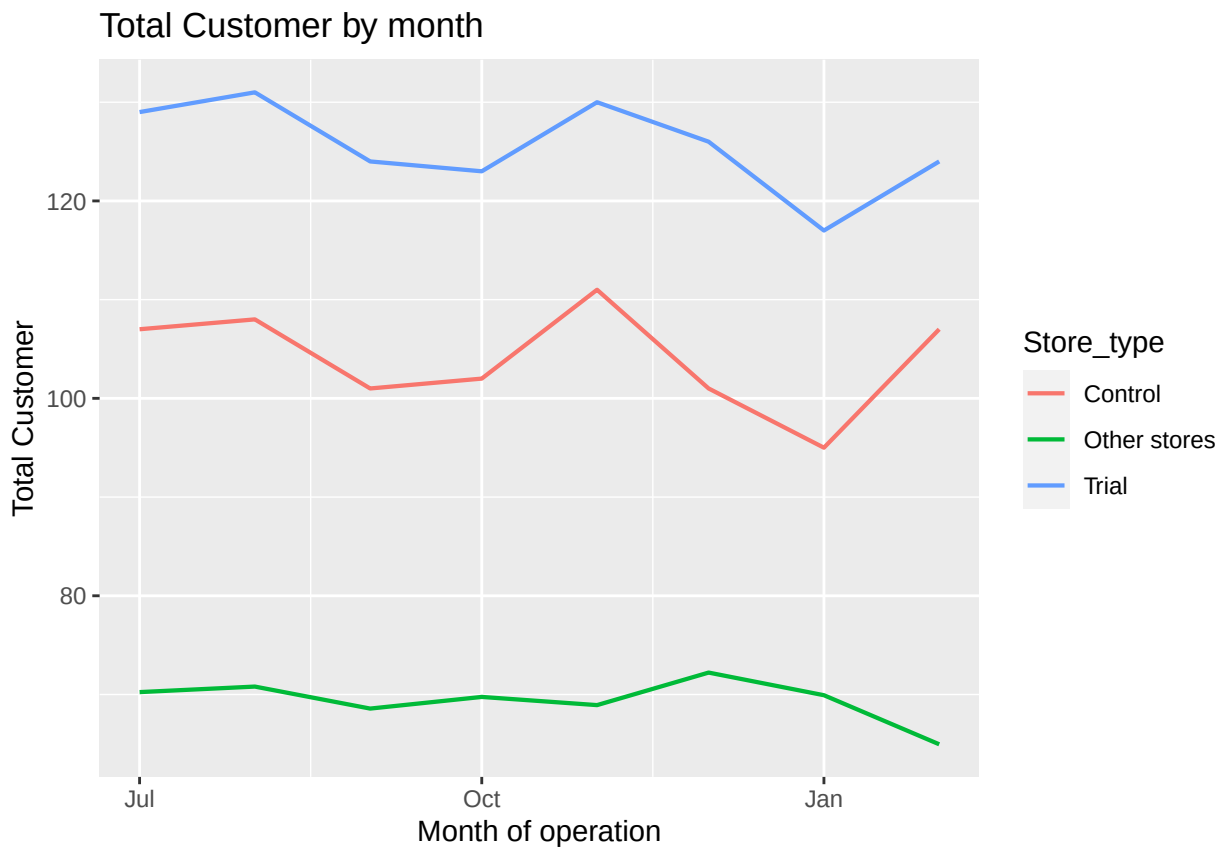
Store 178 is most related to trial store 88. Now, we'll see how much it is related visually.

```
###
MeasurebyMonthSales <-MeasurebyMonth

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



The trend is similar however the sales for trial period is significantly higher than store 178.

```
#Analyzing trend for Total Customers by month
MeasurebyMonthCustomer <-MeasurebyMonth

pastCustomer <- MeasurebyMonthCustomer[, Store_type := ifelse(STORE_NBR == trial_store,
"Trial",
                                       ifelse(STORE_NBR == control_store,
"Control", "Other stores"))
                                ][, totCustomer := mean(CustNum), by = c("YEAR_MONTH",
"Store_type")
                                ][, TransactionMonth := as.Date(paste(YEAR_MONTH %/%
100, YEAR_MONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
                                ][YEAR_MONTH < 201903 , ]
ggplot(pastCustomer, aes(TransactionMonth, totCustomer, color = Store_type)) +
  geom_line(size=0.75) +
  labs(x = "Month of operation", y = "Total Customer", title = "Total Customer by month")
```



Again the trend for number of customers is similar but the difference is significant.Let's look at the first 5 stores that are most related to store 88 to check manually if any other store is most related instead of 178.

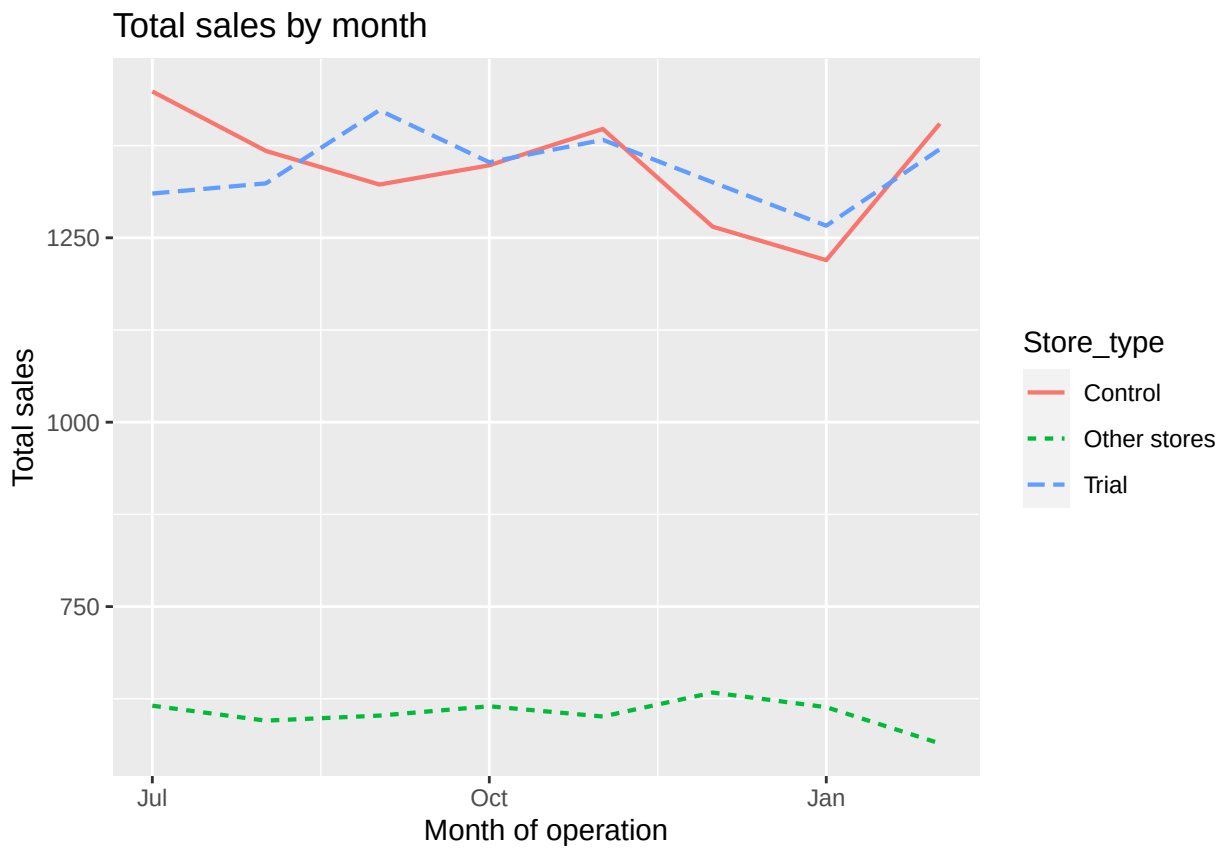
```
head(score_Control[order(-finalControlStore)])
```

```
##    store1 store2 corr_measure.x mag_measure.x scoreNsales corr_measure.y
## 1:     88     88      1.0000000     1.0000000   1.0000000      1.0000000
## 2:     88    178      0.4762637     0.7019083   0.5890860      0.8825679
## 3:     88    238      0.5412904     0.8561314   0.6987109      0.5884505
## 4:     88     69      0.2211324     0.7142712   0.4677018      0.9378170
## 5:     88    259      0.4676555     0.7112675   0.5894615      0.6330597
## 6:     88    237     -0.2143920     0.9549292   0.3702686      0.9013208
##    mag_measure.y scoreNcustomer finalControlStore
## 1:     1.0000000      1.0000000         1.0000000
## 2:     0.8232306      0.8528993         0.7209926
## 3:     0.8921469      0.7402987         0.7195048
## 4:     0.8688871      0.9033521         0.6855269
## 5:     0.8244049      0.7287323         0.6590969
## 6:     0.9855166      0.9434187         0.6568437
```

I visually checked from store 238 to 237 and find out store 237 is visually most related to the trial store. Here's the visual.

```
MeasurebyMonth <- data[, .(totSales=sum(TOT_SALES),
                           CustNum= uniqueN(CARD_NBR),
```

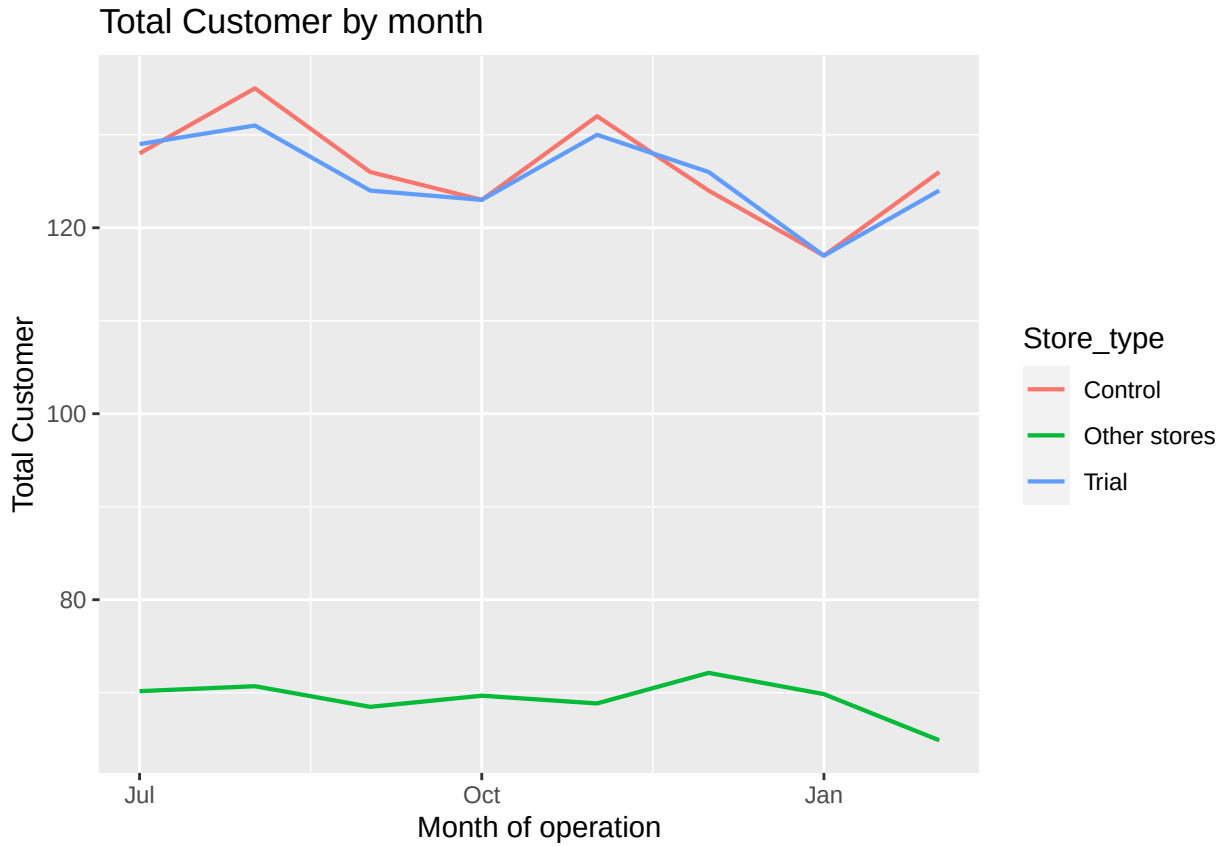
```
                            TxnperCust= uniqueN(TXN_ID)/uniqueN(CARD_NBR),
                            ChipsperTrnsc= sum(PROD_QTY)/uniqueN(TXN_ID),
                            avgPriceperUnit= sum(TOT_SALES)/sum(PROD_QTY)),
                          .(STORE_NBR,YEAR_MONTH)][order(YEAR_MONTH)]

control_store <- 237

MeasurebyMonthSales <-MeasurebyMonth

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

ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line(aes(linetype = Store_type),size=0.75) +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by month")
```



Now, checking for the number of customers.

```
MeasurebyMonthCustomer <-MeasurebyMonth

pastCustomer <- MeasurebyMonthCustomer[, Store_type := ifelse(STORE_NBR == trial_store,
"Trial",
```

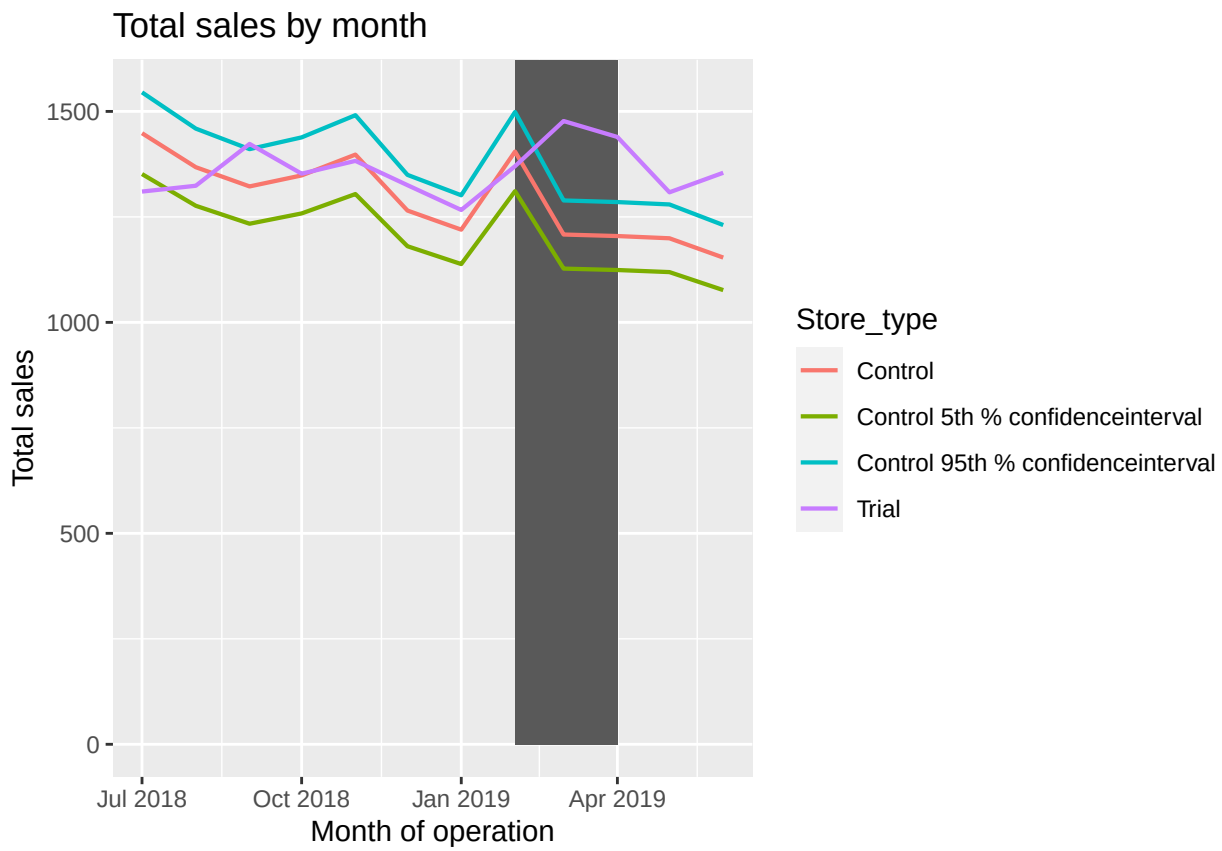
```
                                        ifelse(STORE_NBR == control_store,
"Control", "Other stores"))
                                ][, totCustomer := mean(CustNum), by = c("YEAR_MONTH",
"Store_type")
                                ][, TransactionMonth := as.Date(paste(YEAR_MONTH %/%
100, YEAR_MONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
                                ][YEAR_MONTH < 201903 , ]
ggplot(pastCustomer, aes(TransactionMonth, totCustomer, color = Store_type)) +
  geom_line(size=0.75) +
  labs(x = "Month of operation", y = "Total Customer", title = "Total Customer by month")
```



Now analyzing impact of trial on sales.

```
#### Scale pre-trial control sales to match pre-trial trial store sales
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
YEAR_MONTH < 201902, sum(totSales)]/preTrialMeasures[STORE_NBR == control_store &
YEAR_MONTH < 201902, sum(totSales)]
#### Apply the scaling factor
MeasurebyMonthSales <- MeasurebyMonth
scaledControlSales <- MeasurebyMonthSales[STORE_NBR == control_store, ][,controlSales := totSales * scal

###Calculating percentage difference between the scaled control sales and the trial store's
### sales during the trial period.
percentageDiff <- merge(scaledControlSales[, c("YEAR_MONTH","controlSales")],
                        MeasurebyMonth[STORE_NBR == trial_store, c("totSales","YEAR_MONTH")],
                    by = "YEAR_MONTH")[, percentageDiff := abs(controlSales-totSales)/controlSales]
```

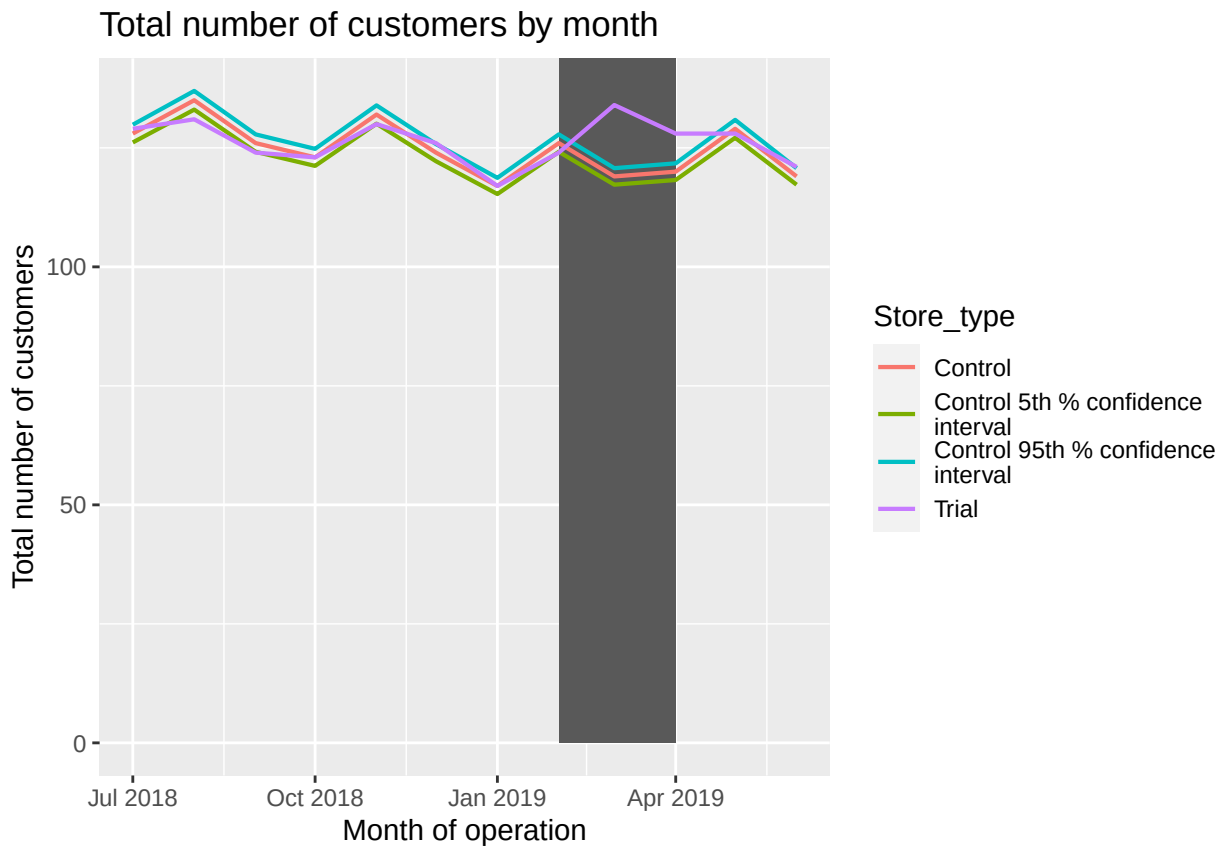
```r
### Standard deviation of percentage difference during pre-trial period
stdDev <- sd(percentageDiff[YEAR_MONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7

MeasurebyMonthSales <- MeasurebyMonth
#### Trial and control store total sales
pastSales <- MeasurebyMonthSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
ifelse(STORE_NBR == control_store, "Control", "Other stores"))
][, totSales := mean(totSales), by = c("YEAR_MONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEAR_MONTH %/% 100, YEAR_MONTH %% 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 % confidenceinterval"]
#### Control store 5th percentile
pastSales_Controls5 <- pastSales[Store_type == "Control",
][, totSales := totSales * (1 - stdDev * 2)
][, Store_type := "Control 5th % confidenceinterval"]
trialAssessment <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)
#### Plotting these in one graph
ggplot(trialAssessment, aes(TransactionMonth, totSales, color = Store_type)) +
geom_rect(data = trialAssessment[ YEAR_MONTH < 201905 & YEAR_MONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
ymin = 0 , ymax =Inf, color = NULL), show.legend = FALSE) +
geom_line(size=0.75) +
labs(x = "Month of operation", y = "Total sales", title = "Total sales by month")
```



This graph shows the trial in store 88 is different to its control store as the performance lies outside 5% and 95% confidence interval for two months i.e., March and April. Now checking impact of trial on number of customers.

```
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store &
YEAR_MONTH < 201902, sum(CustNum)]/preTrialMeasures[STORE_NBR == control_store &
YEAR_MONTH < 201902, sum(CustNum)]
#### Apply the scaling factor
MeasurebyMonthCustomer <- MeasurebyMonth
scaledControlCustomer <- MeasurebyMonthCustomer[STORE_NBR == control_store, ][ ,
controlCustomer := CustNum * scalingFactorForControlCust]

percentageDiff <- merge(scaledControlCustomer[, c("YEAR_MONTH","controlCustomer")],
                        MeasurebyMonth[STORE_NBR == trial_store, c("CustNum","YEAR_MONTH")],
                    by = "YEAR_MONTH")[, percentageDiff := abs(controlCustomer-CustNum)/controlCustome
####
stdDev <- sd(percentageDiff[YEAR_MONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7
#### Trial and control store number of customers
pastCustomers <- MeasurebyMonthCustomer[, nCusts := mean(CustNum), by =
c("YEAR_MONTH", "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 everything into one nice graph.
#### 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[ YEAR_MONTH < 201905 & YEAR_MONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0 ,
ymax = Inf, color = NULL), show.legend = FALSE) +
  geom_line(size=0.75) + labs(x = "Month of operation", y = "Total number of customers", title = "Total
```



The total number of customers in trial period is higher than control store.

Conclusion

We can conclude that for trial stores 77, 86, 88 the control stores are 233, 155 and 237 respectively. Now since we have finished the analysis it's time for presentation.