Task2-Retail Strategy and Analytics

Xuan Fang

7/7/2021

Load required libraries and datasets

Note that you will need to install these libraries if you have never used these before.

Point the filePath to where you have downloaded the datasets to and

```
# Fill in the path to your working directory. If you are on a Windows machine, you will need to use for

filePath <- "C:/Users/user/Desktop/Quantium_Internship/Task02/"
data <- read.csv(paste0(filePath,"QVI_data.csv"))

#### Set themes for plots
theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = 0.5))
```

assign the data files to data.tables 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.

Calculate these measures over time for each store

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

```
data$YEARMONTH <- format(as.Date(data$DATE), "%Y%m") # %Y stands for 4-digit year
```

```
str(data)
```

```
## 'data.frame':    264834 obs. of  13 variables:
##  $ LYLTY_CARD_NBR  : int  1000 1002 1003 1003 1004 1005 1007 1007 1009 1010 ...
##  $ DATE            : chr  "2018-10-17" "2018-09-16" "2019-03-07" "2019-03-08" ...
##  $ STORE_NBR       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ TXN_ID          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ PROD_NBR        : int  5 58 52 106 96 86 49 10 20 51 ...
##  $ PROD_NAME       : chr  "Natural Chip        Compny SeaSalt175g" "Red Rock Deli Chikn&Garlic Aioli
##  $ PROD_QTY        : int  2 1 1 1 1 1 1 1 1 2 ...
##  $ TOT_SALES       : num  6 2.7 3.6 3 1.9 2.8 3.8 2.7 5.7 8.8 ...
##  $ PACK_SIZE       : int  175 150 210 175 160 165 110 150 330 170 ...
##  $ BRAND           : chr  "NATURAL" "RRD" "GRNWVES" "NATURAL" ...
##  $ LIFESTAGE       : chr  "YOUNG SINGLES/COUPLES" "YOUNG SINGLES/COUPLES" "YOUNG FAMILIES" "YOUNG FA
```

```
##  $ PREMIUM_CUSTOMER: chr  "Premium" "Mainstream" "Budget" "Budget" ...
##  $ YEARMONTH       : chr  "201810" "201809" "201903" "201903" ...
```

```
head(data)
```

```
##   LYLTY_CARD_NBR       DATE STORE_NBR TXN_ID PROD_NBR
## 1           1000 2018-10-17         1      1        5
## 2           1002 2018-09-16         1      2       58
## 3           1003 2019-03-07         1      3       52
## 4           1003 2019-03-08         1      4      106
## 5           1004 2018-11-02         1      5       96
## 6           1005 2018-12-28         1      6       86
##                                PROD_NAME PROD_QTY TOT_SALES PACK_SIZE
## 1 Natural Chip        Compny SeaSalt175g        2       6.0       175
## 2  Red Rock Deli Chikn&Garlic Aioli 150g        1       2.7       150
## 3  Grain Waves Sour    Cream&Chives 210G        1       3.6       210
## 4 Natural ChipCo      Hony Soy Chckn175g        1       3.0       175
## 5         WW Original Stacked Chips 160g        1       1.9       160
## 6                     Cheetos Puffs 165g        1       2.8       165
##        BRAND              LIFESTAGE PREMIUM_CUSTOMER YEARMONTH
## 1    NATURAL  YOUNG SINGLES/COUPLES          Premium    201810
## 2        RRD  YOUNG SINGLES/COUPLES       Mainstream    201809
## 3    GRNWVES          YOUNG FAMILIES           Budget    201903
## 4    NATURAL          YOUNG FAMILIES           Budget    201903
## 5 WOOLWORTHS  OLDER SINGLES/COUPLES       Mainstream    201811
## 6    CHEETOS MIDAGE SINGLES/COUPLES       Mainstream    201812
```

2) We define the measure calculations to use during the analysis. For each store and month calculate 1)total sales, 2)number of customers, 3)transactions per customer, 4)chips per customer and 5)the average price per unit.

```
## Hint: you can use uniqueN() to count distinct values in a column
measureOverTime <- data %>% group_by(STORE_NBR, YEARMONTH) %>%
  summarise(totSales=sum(TOT_SALES),
            nCustomers=uniqueN(LYLTY_CARD_NBR),
            nTxnPerCust=uniqueN(TXN_ID)/uniqueN(LYLTY_CARD_NBR),
            nChipsPerTxn=sum(PROD_QTY)/uniqueN(LYLTY_CARD_NBR),
            avgPricePerUnit=(sum(TOT_SALES)/sum(PROD_QTY)))
```

```
## `summarise()` has grouped output by 'STORE_NBR'. You can override using the `.groups` argument.
```

```
head(measureOverTime)
```

```
## # A tibble: 6 x 7
## # Groups:   STORE_NBR [1]
##   STORE_NBR YEARMONTH totSales nCustomers nTxnPerCust nChipsPerTxn
##       <int> <chr>        <dbl>      <int>       <dbl>        <dbl>
## 1         1 201807        207.         49        1.06         1.27
## 2         1 201808        176.         42        1.02         1.29
## 3         1 201809        279.         59        1.05         1.27
## 4         1 201810        188.         44        1.02         1.32
## 5         1 201811        193.         46        1.02         1.24
## 6         1 201812        190.         42        1.12         1.36
## # ... with 1 more variable: avgPricePerUnit <dbl>
```

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

Filter stores with full observation periods:

```
storesWithFullObs <- table(measureOverTime$STORE_NBR) %>% as.data.table() %>%
  filter(N==12) %>% setNames(c("STORE_NBR", "N"))
```

Filter to pre-trial period:

```
preTrialMeasures <- subset(measureOverTime, YEARMONTH<201902 & STORE_NBR %in% storesWithFullObs$STORE_NB
preTrialMeasures
```

```
## # A tibble: 1,820 x 7
## # Groups:   STORE_NBR [260]
##    STORE_NBR YEARMONTH totSales nCustomers nTxnPerCust nChipsPerTxn
##        <int> <chr>        <dbl>      <int>       <dbl>        <dbl>
##  1         1 201807        207.         49        1.06         1.27
##  2         1 201808        176.         42        1.02         1.29
##  3         1 201809        279.         59        1.05         1.27
##  4         1 201810        188.         44        1.02         1.32
##  5         1 201811        193.         46        1.02         1.24
##  6         1 201812        190.         42        1.12         1.36
##  7         1 201901        155.         35        1.03         1.2
##  8         2 201807        151.         39        1.05         1.18
##  9         2 201808        194.         39        1.10         1.41
## 10         2 201809        154.         36        1.03         1.14
## # ... with 1,810 more rows, and 1 more variable: avgPricePerUnit <dbl>
```

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. Let's write a function for this so that we don't have to calculate this for each trial store and control store pair.

Correaltion Create a function to calculate correlation for a measure, looping through each control store.

```
#### Let's define `inputTable` as a metric table with potential comparison stores,
####                `metricCol` as the store metric used to calculate correlation on, and
####                `storeComparison` as the store number of the trial store.

calculateCorrelation <- function(inputTable, metricCol, storeComparison){
          calcCorrTable = data.table(Store1 = numeric(), Store2 = numeric(), corr_measure = numeric())
          trialStore <-  inputTable%>%filter(STORE_NBR==storeComparison)%>%pull(metricCol)
          storeNumbers = unique(inputTable$STORE_NBR)

            for (i in storeNumbers){
             control_st <- inputTable%>%filter(STORE_NBR==i)%>%pull(metricCol)
             calculatedMeasure = data.table("Store1" = storeComparison,
                                            "Store2" = i,
                                            "corr_measure" = cor(trialStore, control_st))

             calcCorrTable <- rbind(calcCorrTable, calculatedMeasure) }

           return(calcCorrTable)
}
```

Standardised Metric 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.

Let's write a function for this.

Solution 1

```
calculateMagnitudeDistance <- function(inputTable, metricCol, storeComparison){
                          calcDistTable = data.table(Store1 = numeric(), Store2 = numeric(),
                                                     YEARMONTH = numeric(), measure = numeric())
                          storeNumbers <- unique(inputTable$STORE_NBR)
                          trialStore <- inputTable%>%filter(STORE_NBR==storeComparison)%>%pull(metricCol

                            for (i in storeNumbers){
                             control_st = inputTable%>%filter(STORE_NBR==i)%>%pull(metricCol)
                             calculatedMeasure = data.table("Store1" = storeComparison,
                                                            "Store2" = i,
                                                            "YEARMONTH" = inputTable$YEARMONTH,
                                                            "measure" = abs(trialStore- control_st))
                            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)
}
```

Find the control stores 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.

Part 0ne: Trial Store 77 Use the function you created to calculate correlations against store 77 using total sales and number of customers.

```
trial_store <- 77

#### Monthly Overall Sales Correlation Table:
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales), trial_store)

#### Monthly Number of Customers Correlation Table:
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers), trial_store)
```

Then, use the functions for calculating magnitude.

```
#### Monthly Overall Sales Magnitude Table:
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales), trial_store)

#### Monthly Number of Customers Magnitude Table:
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures, quote(nCustomers), trial_store)
```

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.

Create a combined score composed of correlation and magnitude, by first merging the correlations table with the magnitude table.

```
#### A simple average on the scores would be 0.5 * corr_measure + 0.5 * mag_measure
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales,
                      by = c("Store1", "Store2"))[, scoreNSales := corr_weight*corr_measure+(1-corr_weig

score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = c("Store1", "Store2"))[,scoreNCust
```

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 by first merging our sales scores and customer scores into a sing
score_Control <- merge(score_nSales, score_nCustomers , by = c("Store1", "Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]
```

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

```
#### Select control stores based on the highest matching store
#### (closest to 1 but not the store itself, i.e. the second ranked highest store)
#### Select the most appropriate control store for trial store 77
#### by finding the store with the highest final score.
control_store <- score_Control[order(-finalControlScore), ][2,Store2]
control_store
```

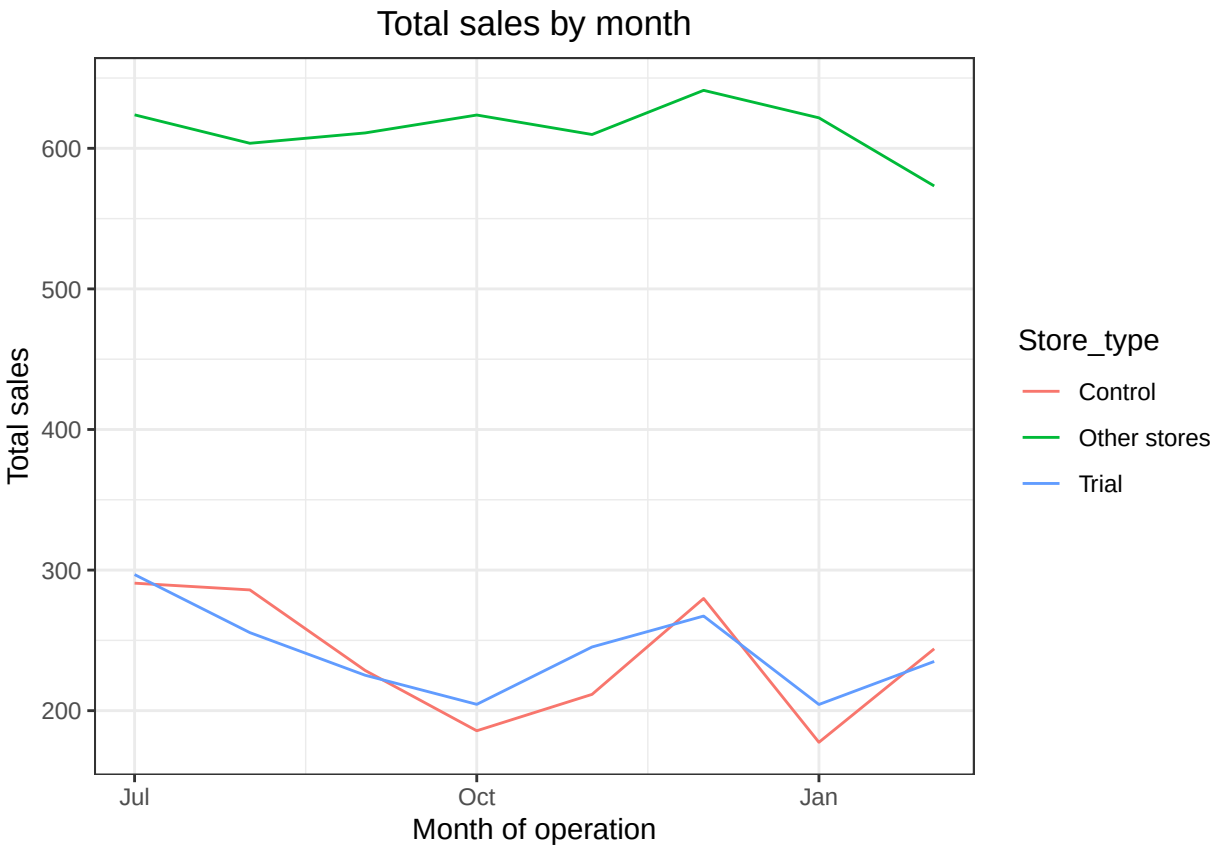
```
## [1] 233
```

Looks like store 233 will be a control store for trial store 77.

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

We'll look at total sales first.

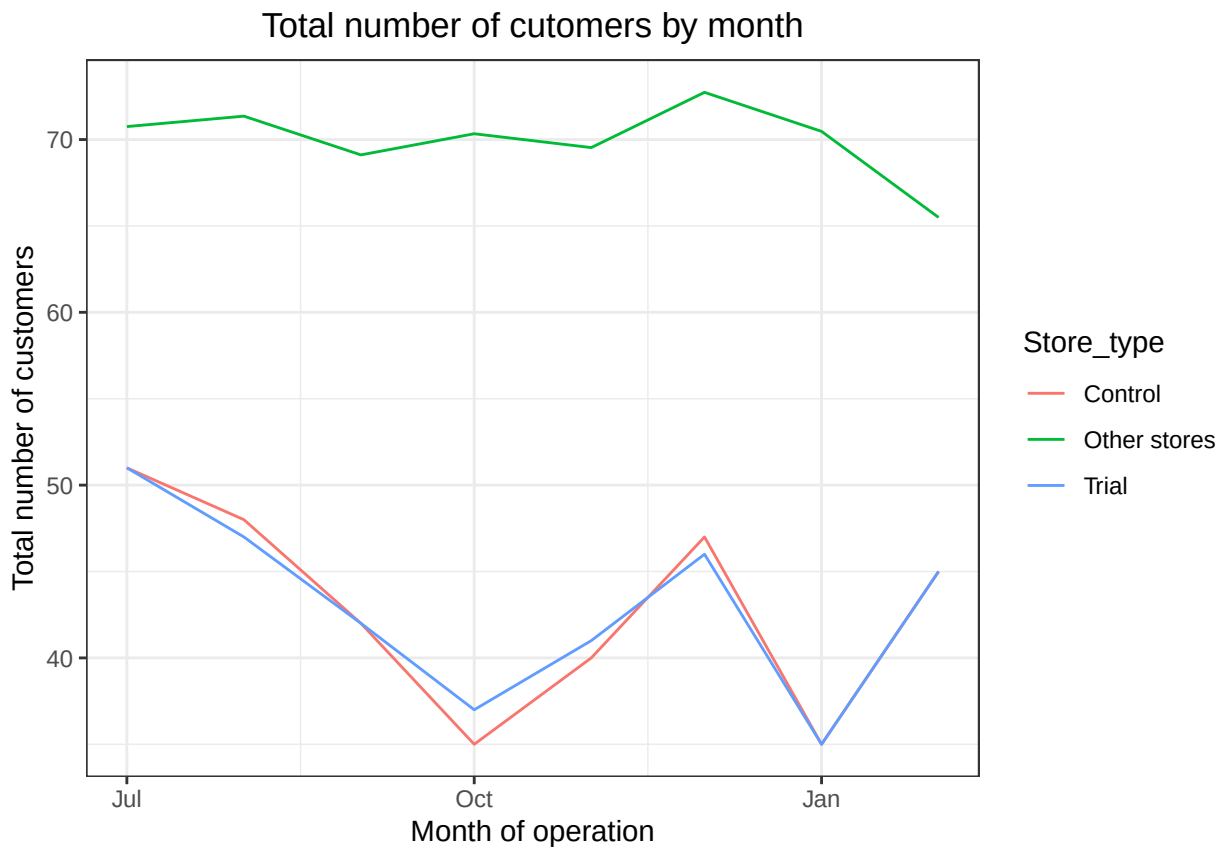
```
#### Visual checks on trends based on the drivers
measureOverTimeSales <- as.data.table(measureOverTime)

#### Note: %/% indicates integer division, %% indicates x mod y
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                         ifelse(STORE_NBR == control_store,"Control", "

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



Sales are trending in a similar way.

Next, we look at the number of customers.

```
#### Conduct visual checks on customer count trends by comparing the trial store to the control store a
measureOverTimeCusts <- as.data.table(measureOverTime)
pastCustomers <- measureOverTimeCusts[, Store_type:=ifelse(STORE_NBR == trial_store, "Trial", ifelse(ST

#### Plot
ggplot(pastCustomers, aes(TransactionMonth, numberCustomers, color =Store_type )) +
 geom_line() +
 labs(x = "Month of operation", y = "Total number of customers", title = "Total number of cutomers by m
```



The trend in number of customers is also similar.

Assessment of trial (Store 77) 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.

```
#### Scale pre-trial control sales to match pre-trial trial store sales
preTrialMeasures <- as.data.table(preTrialMeasures)

scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum(tot$

#### Apply the scaling factor
measureOverTimeSales <- as.data.table(measureOverTime)
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][ ,controlSales := totSales*scal
scaledControlSales
```

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.

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

Let's see if the difference is significant!

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

```
#### pre-trial period (201807-201901)
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
#### Note that there are 8 months in the pre-trial period
#### hence 7 - 1 = 6 degrees of freedom
degreesOfFreedom <- 6


#### 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. After that, find the 95th percentile of the t distrib

#### to check whether the hypothesis is statistically significant.
#### Hint: The test statistic here is (x - u)/standard deviation

percentageDiff[ , tvalue := (percentageDiff - 0)/stdDev][ , TransactionMonth := as.Date(paste(as.numeri
```

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

```
qt(0.95, df = degreesOfFreedom)
```

```
## [1] 1.94318
```

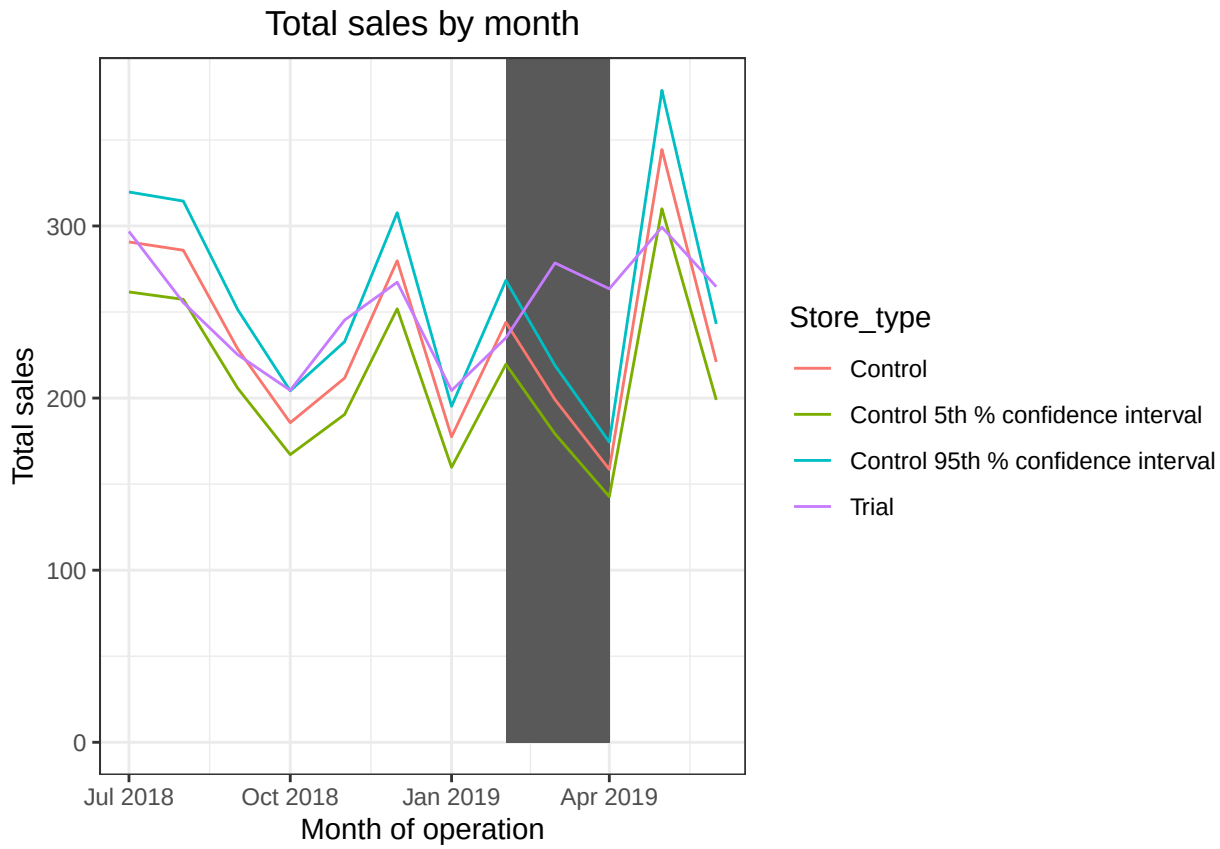
We can observe that the t-value is much larger than the 95th percentile value of the t-distribution for March and April - i.e. the increase in sales in the trial store in March and April is statistically greater than in the control store.

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 <- as.data.table(measureOverTime)
#### Trial and control store total sales
#### Create new variables Store_type, totSales and TransactionMonth in the data table.
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR ==trial_store, "Trial", ifelse(STORE_

#### Control store 95th percentile
#### Note: 95% percent within two standard deviations
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)
```

Plot

```
#### 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 th 5% to 95% confidence interval of the control store in two of the three trial months.

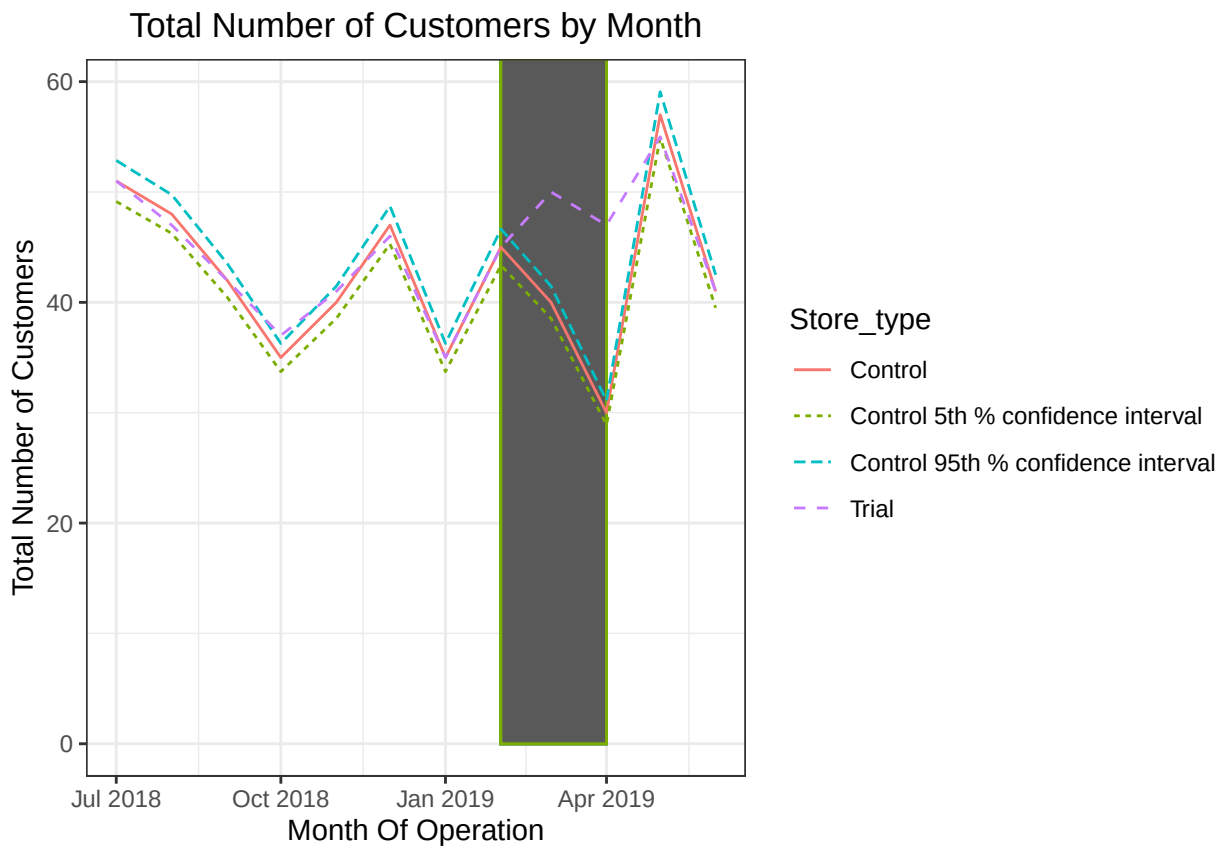
Let's have a look at assessing this for number of customers as well.

```
#### This would be a repeat of the steps before for total sales
#### Scale pre-trial control customers to match pre-trial trial store customers
#### Compute a scaling factor to align control store customer counts to our trial store.
#### Then, apply the scaling factor to control store customer counts.
#### Finally, calculate the percentage difference between scaled control store customers and trial cust
#### Scale pre-trial control customer counts to match pre-trial trial store customer counts
preTrialMeasures <- as.data.table(preTrialMeasures)

scalingFactorForControlCust  <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum(nCus

#### Apply the scaling factor
measureOverTimeCusts <- as.data.table(measureOverTime)
scaledControlCustomers <- measureOverTimeSales[STORE_NBR == control_store, ][ ,controlCustomers := nCust

#### Calculate the percentage difference between scaled control store customers and trial customers.
percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH", "controlCustomers")], measureOverTimeCus
```

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

```
#### As our null hypothesis is that the trial period is the same as the pre-trial period, let's take th
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
```

```
degreesOfFreedom <- 6

#### Trial and control store number of customers
measureOverTimeCusts <- as.data.table(measureOverTime)
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store, "Trial", ifelse(S
][Store_type %in% c("Trial", "Control"), ]

###Control 95th percentile
pastCustomers_Control95 <- pastCustomers[Store_type == "Control",][, nCusts := nCusts * (1 + stdDev * 2)

###Control 5th percentile
pastCustomers_Control5 <- pastCustomers[Store_type == "Control",][, nCusts := nCusts * (1 - stdDev * 2)]

trialAssessment <- rbind(pastCustomers,pastCustomers_Control95,pastCustomers_Control5)

###Visualize
ggplot(trialAssessment, aes(TransactionMonth, nCusts, color = Store_type)) +
  geom_rect(data = trialAssessment[YEARMONTH < 201905 & YEARMONTH > 201901 , ], aes(xmin = min(Transacti
  geom_line(aes(linetype = Store_type)) +
  labs(x = "Month Of Operation",
       y = "Total Number of Customers",
       title = "Total Number of Customers by Month")
```



Let's repeat finding the control store and assessing the impact of the trial for each of the other two trial stores.

Part Two: Trial Store 86 Use the function you created to calculate correlations against store 77 using total sales and number of customers.

```
trial_store <-  86

#### Monthly Overall Sales Correlation Table:
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales), trial_store)

#### Monthly Number of Customers Correlation Table:
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers), trial_store)
```

Then, use the functions for calculating magnitude.

```
#### Monthly Overall Sales Magnitude Table:
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales), trial_store)

#### Monthly Number of Customers Magnitude Table:
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures, quote(nCustomers), trial_store)
```

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.

Create a combined score composed of correlation and magnitude, by first merging the correlations table with the magnitude table.

```
#### A simple average on the scores would be 0.5 * corr_measure + 0.5 * mag_measure
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1", "Store2"))[, scoreNSales := corr_

score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = c("Store1", "Store2"))[,scoreNCus
```

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 by first merging our sales scores and customer scores into a sin
score_Control <- merge(score_nSales, score_nCustomers , by = c("Store1", "Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]
```

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

```
#### Select control stores based on the highest matching store
#### (closest to 1 but not the store itself, i.e. the second ranked highest store)
#### Select the most appropriate control store for trial store 77 by finding the store with
#### the highest final score.
control_store <- score_Control[order(-finalControlScore), ][2,Store2]
control_store
```

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

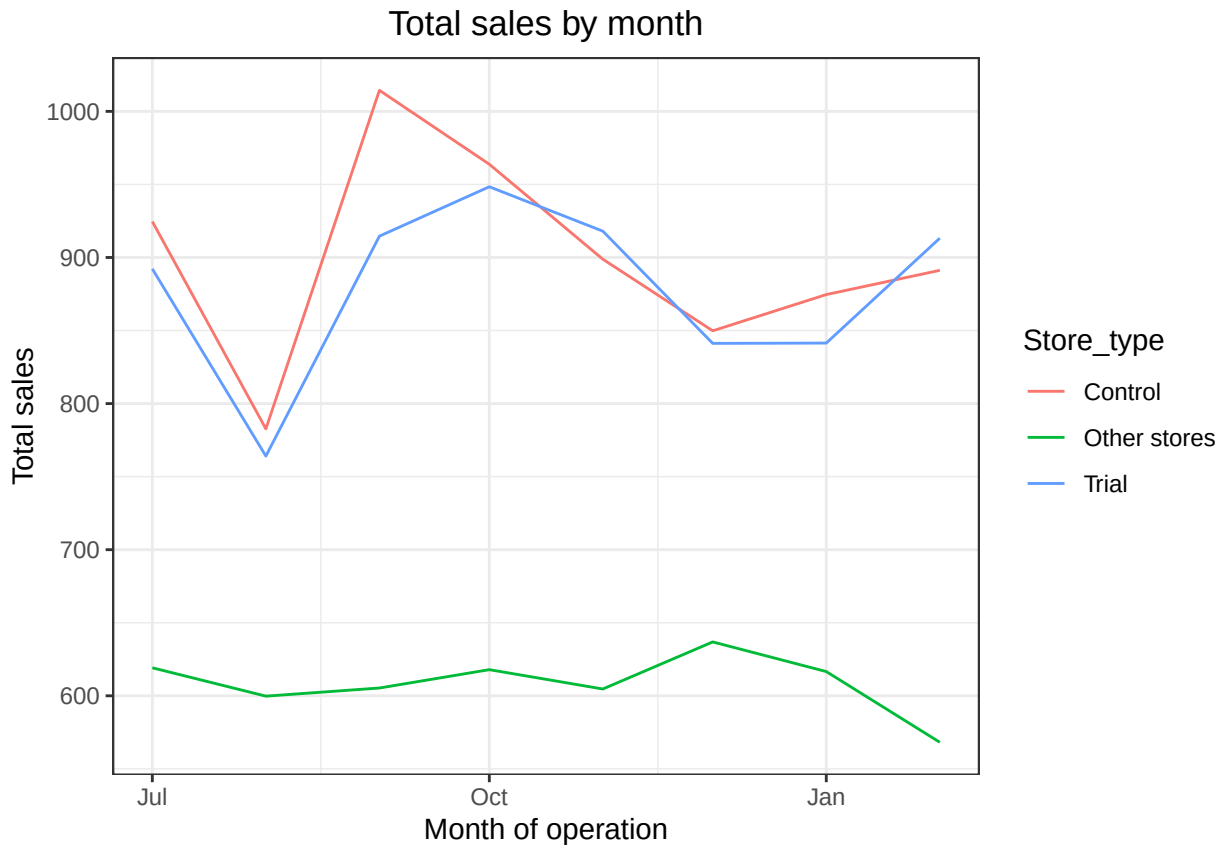
Looks like store 155 will be a control store for trial store 86. Again, let's check visually if the drivers are indeed similar in the period before the trial.

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

```
#### Visual checks on trends based on the drivers
measureOverTimeSales <- as.data.table(measureOverTime)
```

```
#### Note: %/% indicates integer division, %% indicates x mod y
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                         ifelse(STORE_NBR == control_store,"Control", "
```

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

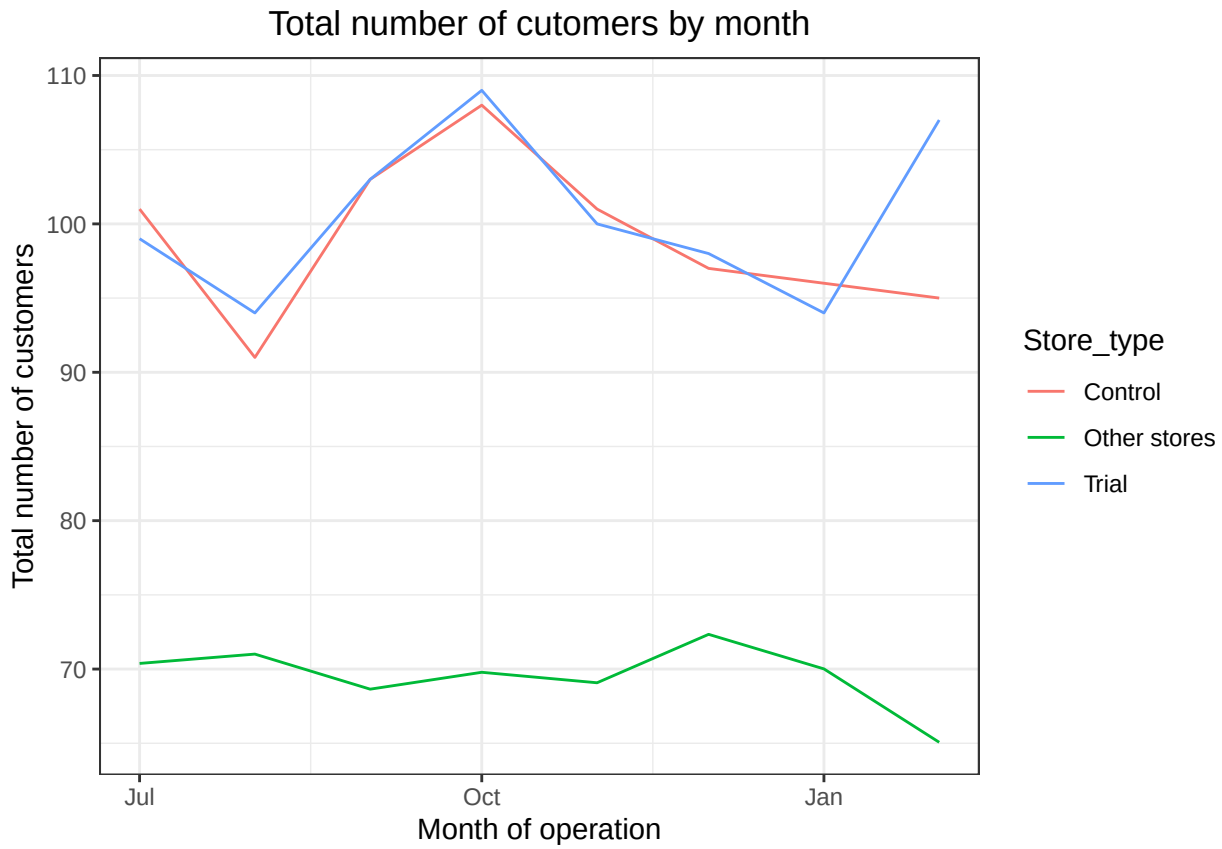


Great, sales are trending in a similar way

Next, number of customers.

```
#### Conduct visual checks on customer count trends by comparing the trial store to the control store a
measureOverTimeCusts <- as.data.table(measureOverTime)
pastCustomers <- measureOverTimeCusts[, Store_type:=ifelse(STORE_NBR == trial_store, "Trial", ifelse(ST
```

```
#### Plot
ggplot(pastCustomers, aes(TransactionMonth, numberCustomers, color =Store_type )) +
 geom_line() +
 labs(x = "Month of operation", y = "Total number of customers", title = "Total number of cutomers by m
```



Good, the trend in number of customers is also similar.

Let's now assess the impact of the trial on sales.

```
#### Scale pre-trial control sales to match pre-trial trial store sales
preTrialMeasures <- as.data.table(preTrialMeasures)

scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum(tot

#### Apply the scaling factor
measureOverTimeSales <- as.data.table(measureOverTime)
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][ ,controlSales := totSales*scal
```

Assessment of trial (Store 86) 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.

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

Let's see if the difference is significant!

```
#### As our null hypothesis is that the trial period is the same as the pre-trial period, let's take th
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
```

```
#### Note that there are 8 months in the pre-trial period
#### hence 7 - 1 = 6 degrees of freedom
degreesOfFreedom <- 6


#### 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. After that, find the 95th percentile of the t distrib

#### to check whether the hypothesis is statistically significant.
#### Hint: The test statistic here is (x - u)/standard deviation

percentageDiff[ , tvalue := (percentageDiff - 0)/stdDev][ , TransactionMonth := as.Date(paste(as.numeri
```

```
##    TransactionMonth    tvalue
## 1:       2019-02-01  2.179542
## 2:       2019-03-01 12.226922
## 3:       2019-04-01  1.364580
```

```
qt(0.95, df = degreesOfFreedom)
```

```
## [1] 1.94318
```

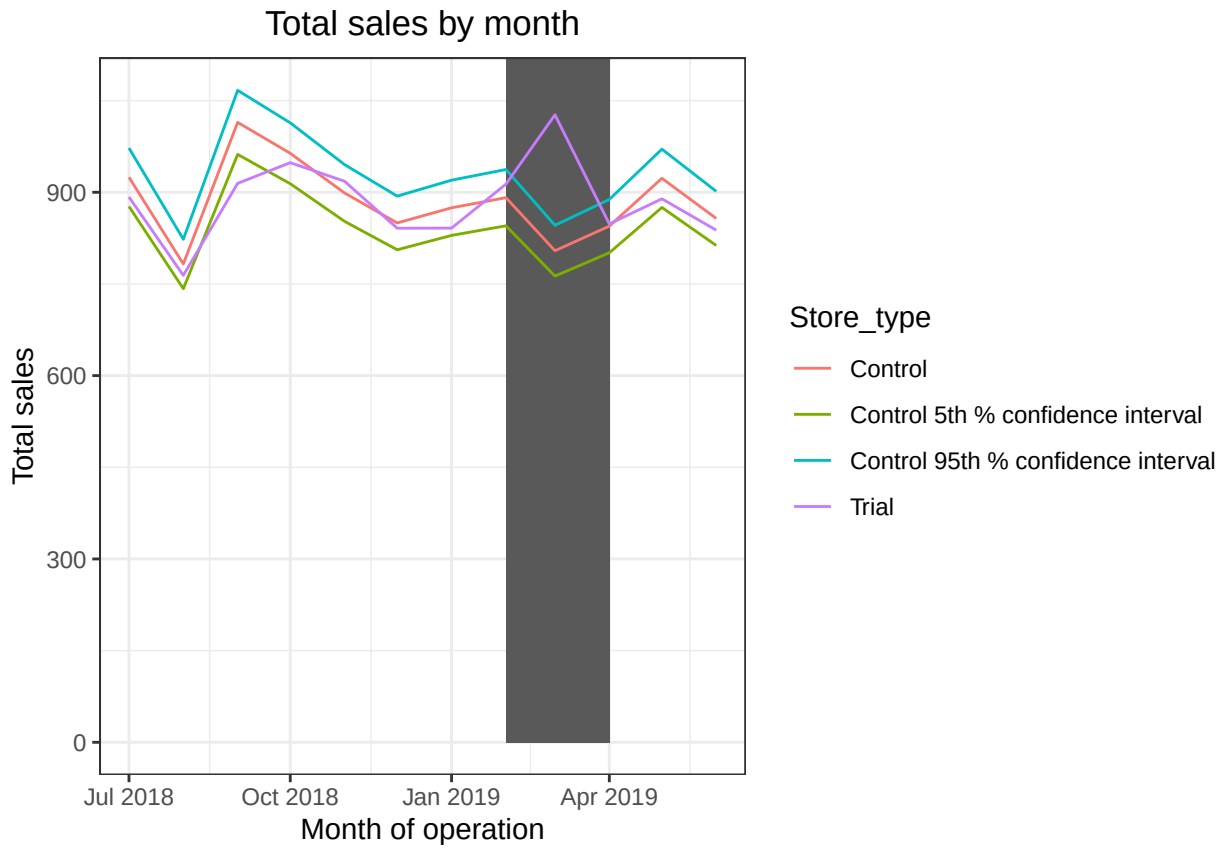
We can observe that the t-value is much larger than the 95th percentile value of the t-distribution for March - i.e. the increase in sales in the trial store in March is statistically greater than in the control store.

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 <- as.data.table(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_

#### Control store 95th percentile
#### Note: 95% percent within two standard deviations
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)
```

Plot

```
#### 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 86 is significantly different to its control store in March as the trial store performance lies outside of the 5% to 95% confidence interval of the control store in two of the three trial months.

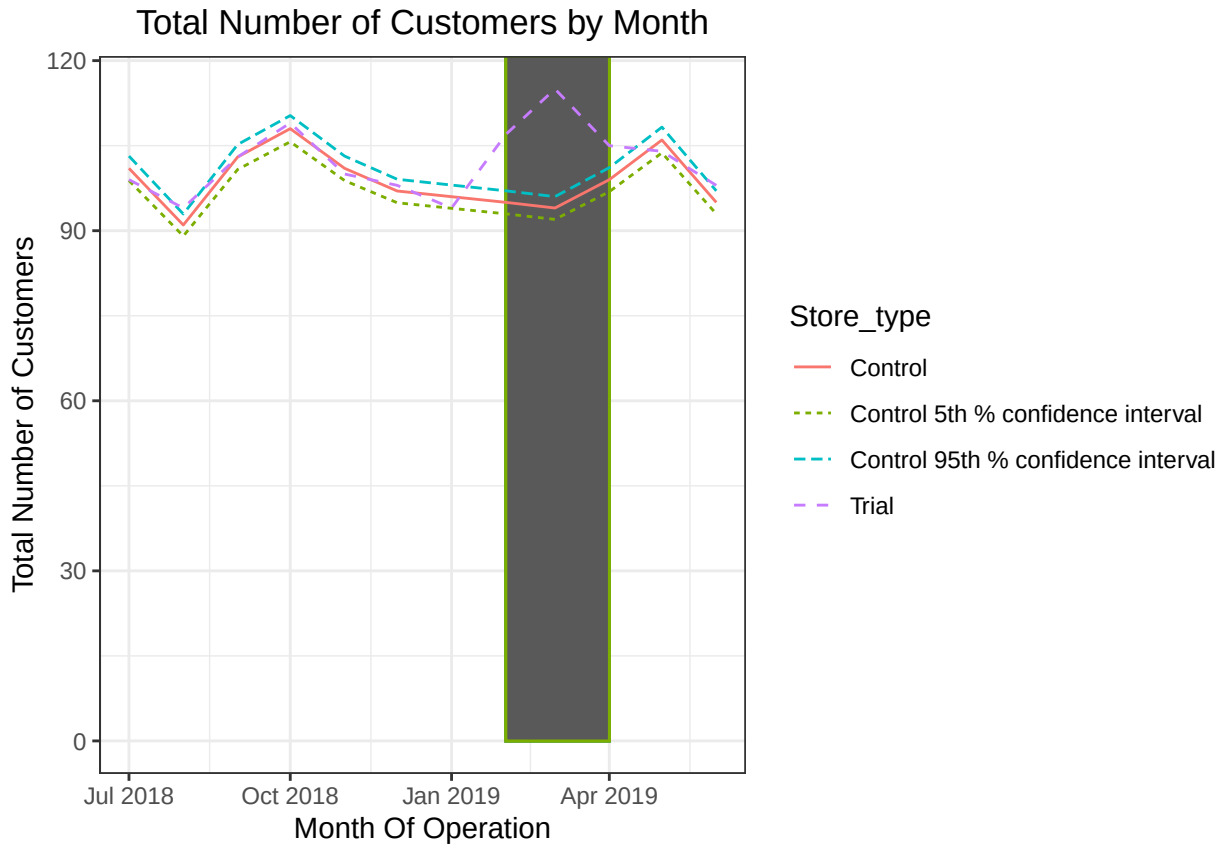
Let's have a look at assessing this for number of customers as well.

```
#### This would be a repeat of the steps before for total sales
#### Scale pre-trial control customers to match pre-trial trial store customers
#### Compute a scaling factor to align control store customer counts to our trial store.
#### Then, apply the scaling factor to control store customer counts.
#### Finally, calculate the percentage difference between scaled control store customers and trial cust
#### Scale pre-trial control customer counts to match pre-trial trial store customer counts
preTrialMeasures <- as.data.table(preTrialMeasures)

scalingFactorForControlCust  <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum(nCus

#### Apply the scaling factor
measureOverTimeCusts <- as.data.table(measureOverTime)
scaledControlCustomers <- measureOverTimeSales[STORE_NBR == control_store, ][ ,controlCustomers := nCust

scaledControlCustomers
```

```
#### Calculate the percentage difference between scaled control store customers and trial customers.
percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH", "controlCustomers")], measureOverTimeCu
```

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

```
#### As our null hypothesis is that the trial period is the same as the pre-trial period, let's take th
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])

degreesOfFreedom <- 6

#### Trial and control store number of customers
measureOverTimeCusts <- as.data.table(measureOverTime)
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store, "Trial", ifelse(S
][Store_type %in% c("Trial", "Control"), ]

###Control 95th percentile
pastCustomers_Control95 <- pastCustomers[Store_type == "Control",][, nCusts := nCusts * (1 + stdDev * 2)

###Control 5th percentile
pastCustomers_Control5 <- pastCustomers[Store_type == "Control",][, nCusts := nCusts * (1 - stdDev * 2)]

trialAssessment <- rbind(pastCustomers,pastCustomers_Control95,pastCustomers_Control5)

###Visualize
ggplot(trialAssessment, aes(TransactionMonth, nCusts, color = Store_type)) +
  geom_rect(data = trialAssessment[YEARMONTH < 201905 & YEARMONTH > 201901 , ], aes(xmin = min(Transacti
  geom_line(aes(linetype = Store_type)) +
  labs(x = "Month Of Operation",
       y = "Total Number of Customers",
       title = "Total Number of Customers by Month")
```



Same results as above.

Part Three: Trial Store 88 Use the function you created to calculate correlations against store 88 using total sales and number of customers.

```
trial_store <-  88

#### Monthly Overall Sales Correlation Table:
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales), trial_store)

#### Monthly Number of Customers Correlation Table:
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers), trial_store)
```

Use the functions for calculating magnitude.

```
#### Monthly Overall Sales Magnitude Table:
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales), trial_store)

#### Monthly Number of Customers Magnitude Table:
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures, quote(nCustomers), trial_store)
```

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.

Create a combined score composed of correlation and magnitude, by first merging the correlations table with the magnitude table.

```
#### A simple average on the scores would be 0.5 * corr_measure + 0.5 * mag_measure
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1", "Store2"))[, scoreNSales := corr_

score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = c("Store1", "Store2"))[,scoreNCus
```

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 by first merging our sales scores and customer scores into a sin
score_Control <- merge(score_nSales, score_nCustomers , by = c("Store1", "Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]
```

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

```
#### Select control stores based on the highest matching store (closest to 1 but not the store itself,
#### Select the most appropriate control store for trial store 77 by finding the store with the highest
control_store <- score_Control[order(-finalControlScore), ][2,Store2]
control_store
```

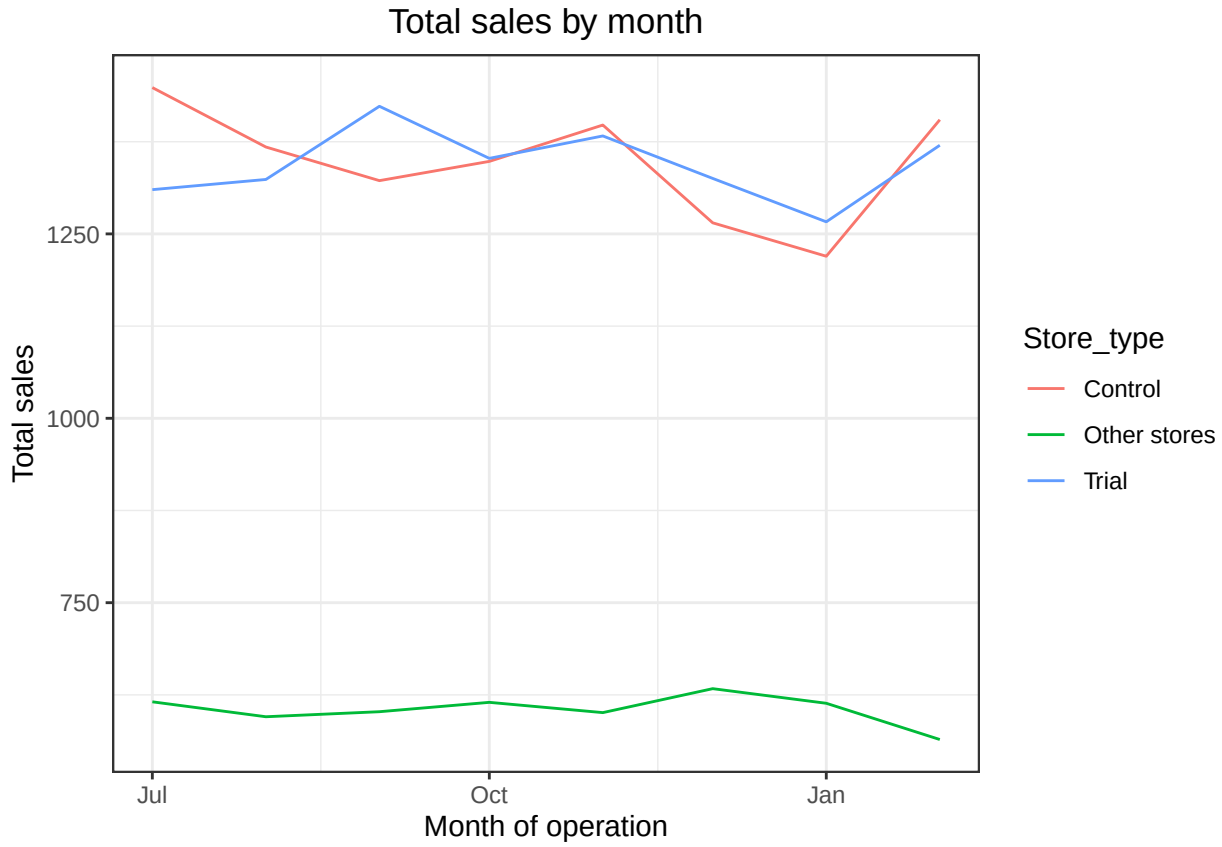
```
## [1] 237
```

We've now found store 237 to be a suitable control store for trial store 88.Again, let's check visually if the drivers are indeed similar in the period before the trial.

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

```
#### Visual checks on trends based on the drivers
measureOverTimeSales <- as.data.table(measureOverTime)
```

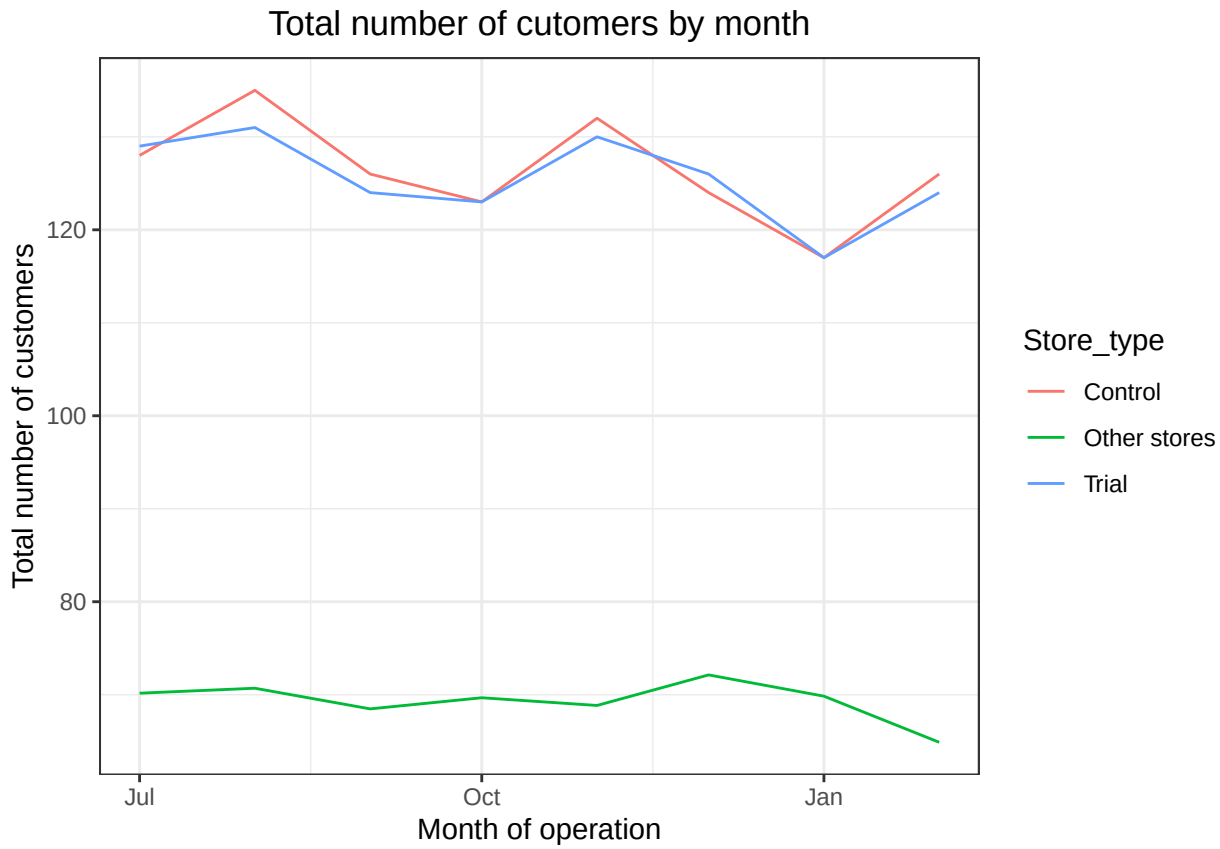
```
#### Note: %/% indicates integer division, %% indicates x mod y
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",ifelse(STORE_

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



Great, the trial and control stores have similar total sales.

Next, number of customers.

```
#### Conduct visual checks on customer count trends by comparing the trial store to the control store a
measureOverTimeCusts <- as.data.table(measureOverTime)
pastCustomers <- measureOverTimeCusts[, Store_type:=ifelse(STORE_NBR == trial_store, "Trial", ifelse(ST

#### Plot
ggplot(pastCustomers, aes(TransactionMonth, numberCustomers, color =Store_type )) +
 geom_line() +
 labs(x = "Month of operation", y = "Total number of customers", title = "Total number of cutomers by m
```



Total number of customers of the control and trial stores are also similar.

Let's now assess the impact of the trial on sales.

```
#### Scale pre-trial control sales to match pre-trial trial store sales
preTrialMeasures <- as.data.table(preTrialMeasures)

scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum(tot

#### Apply the scaling factor
measureOverTimeSales <- as.data.table(measureOverTime)
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][ ,controlSales := totSales*scal
```

Assessment of trial (Store 88) 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.

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

Let's see if the difference is significant!

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

```
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
#### Note that there are 8 months in the pre-trial period
#### hence 7 - 1 = 6 degrees of freedom
degreesOfFreedom <- 6

#### 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. After that, find the 95th percentile of the t distrib

#### to check whether the hypothesis is statistically significant.
#### Hint: The test statistic here is (x - u)/standard deviation

percentageDiff[ , tvalue := (percentageDiff - 0)/stdDev][ , TransactionMonth := as.Date(paste(as.numeri
```

```
##    TransactionMonth    tvalue
## 1:       2019-02-01 0.7812695
## 2:       2019-03-01 6.5956678
## 3:       2019-04-01 5.7685269
```

```
qt(0.95, df = degreesOfFreedom)
```

```
## [1] 1.94318
```

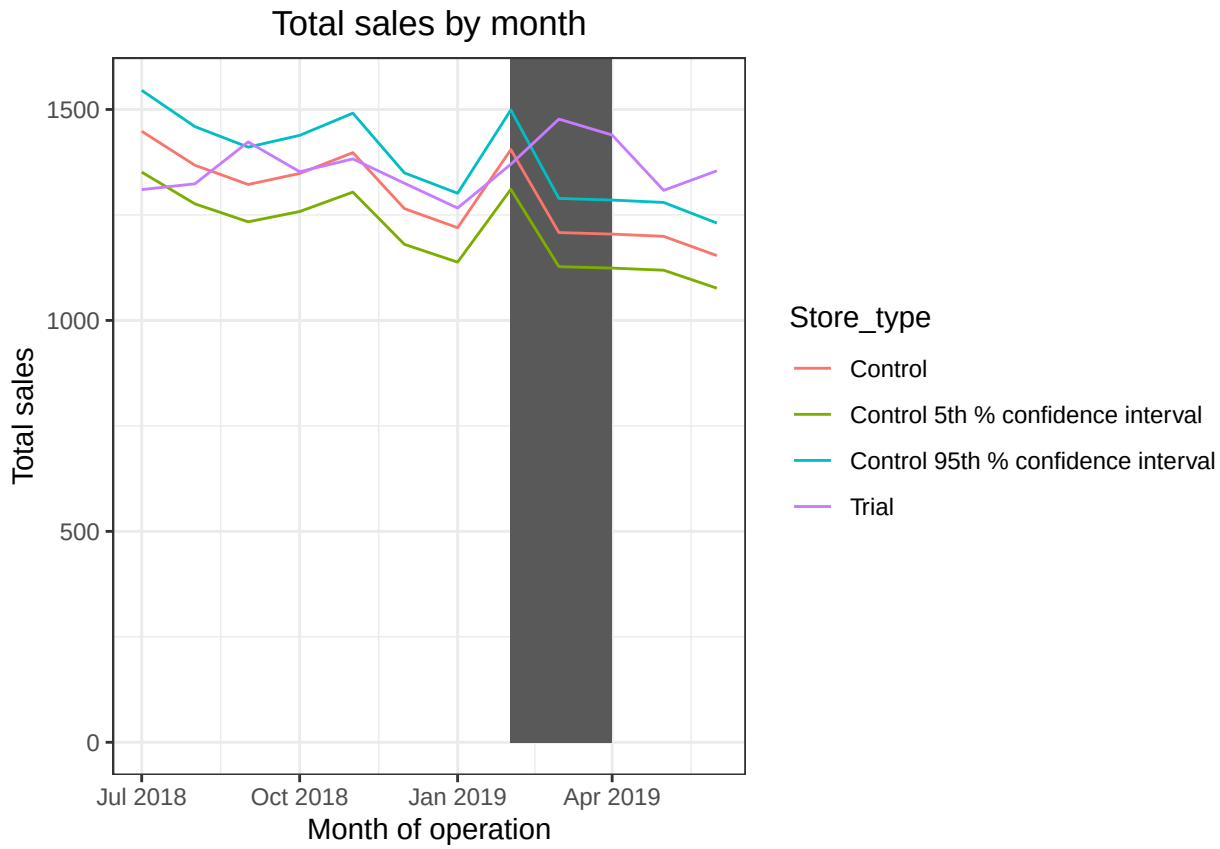
We can observe that the t-value is much larger than the 95th percentile value of the t-distribution for March and April - i.e. the increase in sales in the trial store in March and April is statistically greater than in the control store.

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 <- as.data.table(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_

#### Control store 95th percentile
#### Note: 95% percent within two standard deviations
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)
```

Plot

```
#### 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 88 is significantly different to its control store in the trial period (March and April) as the trial store performance lies outside of the 5% to 95% confidence interval of the control store in two of the three trial months.

Let's have a look at assessing this for number of customers as well.

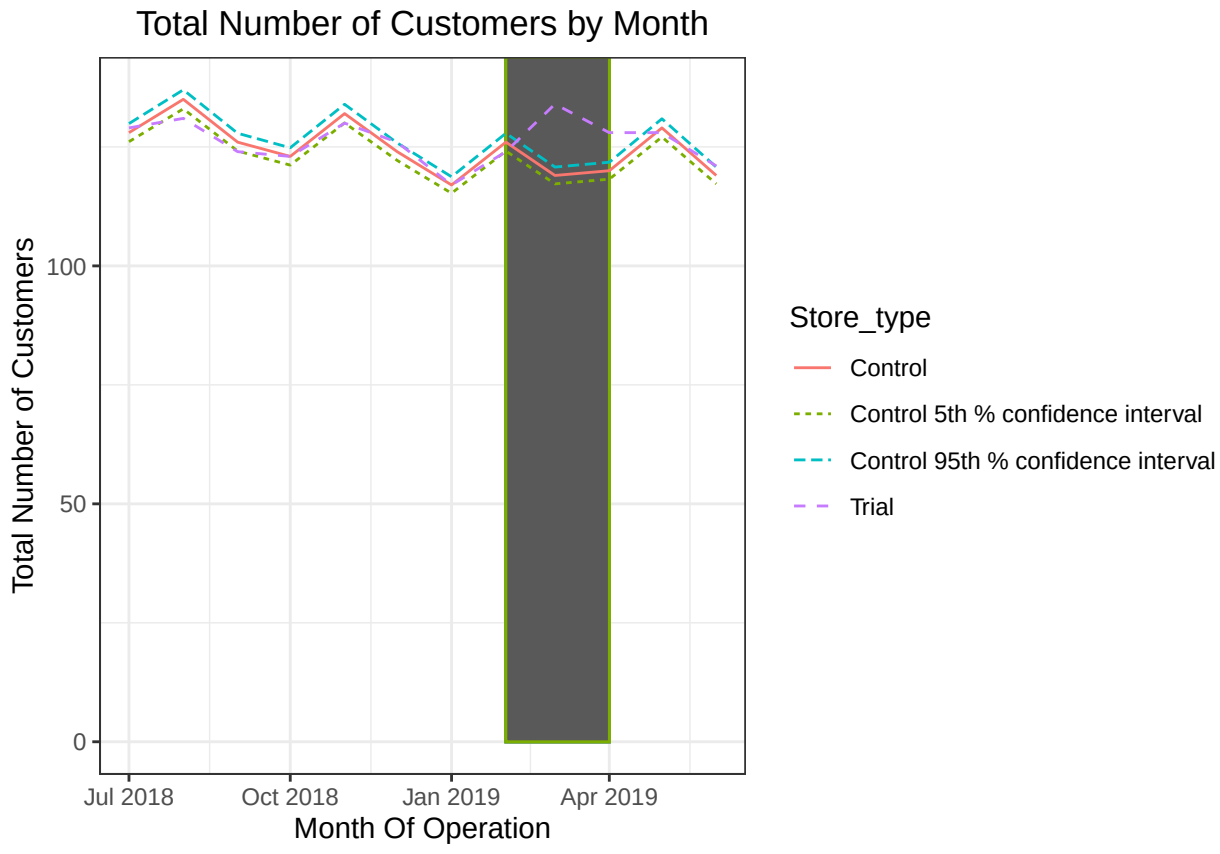
```
#### This would be a repeat of the steps before for total sales
#### Scale pre-trial control customers to match pre-trial trial store customers
#### Compute a scaling factor to align control store customer counts to our trial store.
#### Then, apply the scaling factor to control store customer counts.
#### Finally, calculate the percentage difference between scaled control store customers and trial cust
#### Scale pre-trial control customer counts to match pre-trial trial store customer counts
preTrialMeasures <- as.data.table(preTrialMeasures)

scalingFactorForControlCust  <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum(nCus

#### Apply the scaling factor
measureOverTimeCusts <- as.data.table(measureOverTime)
scaledControlCustomers <- measureOverTimeSales[STORE_NBR == control_store, ][ ,controlCustomers := nCust
```

```
#### Calculate the percentage difference between scaled control store customers and trial customers.
percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH", "controlCustomers")], measureOverTimeCu
```

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

```
#### As our null hypothesis is that the trial period is the same as the pre-trial period, let's take th
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
```

```r
degreesOfFreedom <- 6

#### Trial and control store number of customers
measureOverTimeCusts <- as.data.table(measureOverTime)
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store, "Trial", ifelse(S
][Store_type %in% c("Trial", "Control"), ]

###Control 95th percentile
pastCustomers_Control95 <- pastCustomers[Store_type == "Control",][, nCusts := nCusts*(1 + stdDev*2)][,

###Control 5th percentile
pastCustomers_Control5 <- pastCustomers[Store_type == "Control",][, nCusts := nCusts*(1 - stdDev*2)][, S

trialAssessment <- rbind(pastCustomers,pastCustomers_Control95,pastCustomers_Control5)

###Visualize
ggplot(trialAssessment, aes(TransactionMonth, nCusts, color = Store_type)) +
  geom_rect(data = trialAssessment[YEARMONTH < 201905 & YEARMONTH > 201901 , ], aes(xmin = min(Transacti
  geom_line(aes(linetype = Store_type)) +
  labs(x = "Month Of Operation",
       y = "Total Number of Customers",
       title = "Total Number of 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.

Conclusion

- 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.