Task_2

Burhanudin Badiuzaman

2024-08-28

Setting Rmarkdown

Solution template for Task 2

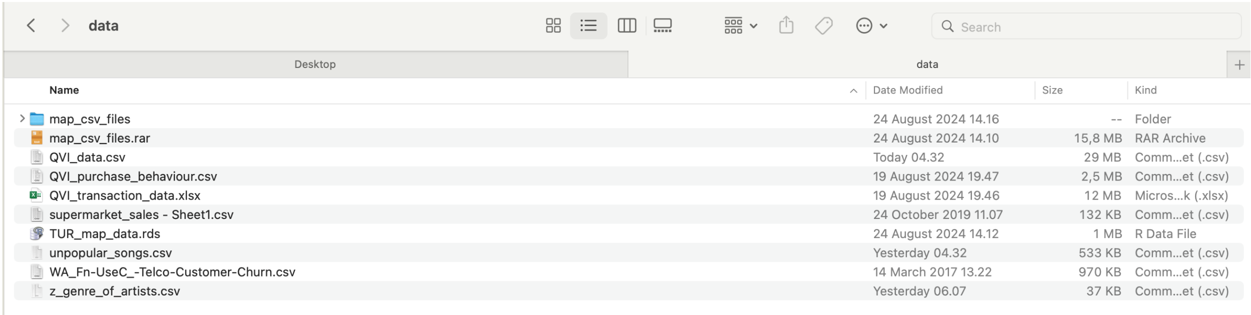
This file is a solution template for the Task 2 of the Quantium Virtual Internship. It will walk you through the analysis, providing the scaffolding for your solution with gaps left for you to fill in yourself. Look for comments that say "over to you" for places where you need to add your own code! Often, there will be hints about what to do or what function to use in the text leading up to a code block - if you need a bit of extra help on how to use a function, the internet has many excellent resources on R coding, which you can find using your favourite search engine.

Load required libraries and datasets

- Note that you will need to install these libraries if you have never used these before and make sure it works in the right work directory.

```r
getwd()
```

```
## [1] "/Users/burhanudin/Study_burhanudin_6/R/Code-R/Data_Analyst_in_R"
```



Name	Date Modified	Size	Kind
map_csv_files	24 August 2024 14.16	--	Folder
map_csv_files.rar	24 August 2024 14.10	15,8 MB	RAR Archive
QVI_data.csv	Today 04.32	29 MB	Comm...et (.csv)
QVI_purchase_behaviour.csv	19 August 2024 19.47	2,5 MB	Comm...et (.csv)
QVI_transaction_data.xlsx	19 August 2024 19.46	12 MB	Micros...k (.xlsx)
supermarket_sales - Sheet1.csv	24 October 2019 11.07	132 KB	Comm...et (.csv)
TUR_map_data.rds	24 August 2024 14.12	1 MB	R Data File
unpopular_songs.csv	Yesterday 04.32	533 KB	Comm...et (.csv)
WA_Fn-UseC_-Telco-Customer-Churn.csv	14 March 2017 13.22	970 KB	Comm...et (.csv)
z_genre_of_artists.csv	Yesterday 06.07	37 KB	Comm...et (.csv)

Figure 1: data.table

Point the filePath to where you have downloaded the datasets to and assign the data files to data.tables

```r
# Over to you! Fill in the path to your working directory. If you are on a Windows machine, you will ne
data <- fread("data/QVI_data.csv")
#### Set themes for plots
theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = 0.5))
```

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

```r
#### Calculate these measures over time for each store
#### Over to you! Add a new month ID column in the data with the format yyyymm.
#monthYear <- format(as.Date(data$DATE, "%Y%m"))
data[, YEARMONTH := year(DATE)*100 + month(DATE)]
#### Next, we define the measure calculations to use during the analysis.
# Over to you! For each store and month calculate total sales, number of customers, transactions per cu
## Hint: you can use uniqueN() to count distinct values in a column
measureOverTime <- data[, .(totSales = sum(TOT_SALES),
                            nCustomers = uniqueN(LYLTY_CARD_NBR),
                            nTxnPerCust = uniqueN(TXN_ID)/uniqueN(LYLTY_CARD_NBR),
                            nChipsPerTxn = uniqueN(PROD_QTY)/uniqueN(TXN_ID),
                            avgPricePerUnit = sum(TOT_SALES)/ sum(PROD_QTY))
                        ,by = c("STORE_NBR","YEARMONTH")][order(STORE_NBR,YEARMONTH)]
#### Filter to the pre-trial period and stores with full observation periods
storesWithFullObs <- unique(measureOverTime[, .N,
                                            STORE_NBR][N == 12, STORE_NBR])
preTrialMeasures <- measureOverTime[YEARMONTH < 201902 & STORE_NBR %in%
storesWithFullObs, ]
```

- Now we need to work out a way of ranking how similar each potential control store is to the trial store. We can calculate how correlated the performance of each store is to the trial store. Let's write a function for this so that we don't have to calculate this for each trial store and control store pair.

```r
#### 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 store Comparison
#### as the store number of the trial store.
calculateCorrelation <- function(inputTable, metricCol, storeComparison) {
  calcCorrTable = data.table(Store1 = numeric(),
                             Store2 = numeric(), corr_measure =
  numeric())
  storeNumbers <- unique(inputTable[, STORE_NBR])

  for (i in storeNumbers) {
    calculatedMeasure = data.table("Store1" = storeComparison,
                                   "Store2" = i,
                                   "corr_measure" = cor( inputTable[STORE_NBR == storeComparison,
                                                                     eval(metricCol)], inputTable[STORE_
                                                                                                eval(me
    calcCorrTable <- rbind(calcCorrTable, calculatedMeasure)
  }
  return(calcCorrTable)
}
```

- Apart from correlation, we can also calculate a standardised metric based on the absolute difference between the trial store's performance and each control store's performance. Let's write a function for this.

```
#### 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)
}
#### Standardize 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.

```
#### Over to you! Use the function you created to calculate correlations against store 77 using total s
#### Hint: Refer back to the input names of the functions we created.
trial_store <- 77
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales),trial_store)
corr_nSales[order(-corr_measure)]
```

```
##      Store1 Store2 corr_measure
##       <num>  <num>        <num>
##    1:     77     77    1.0000000
##    2:     77     71    0.9141060
##    3:     77    233    0.9037742
##    4:     77    119    0.8676644
##    5:     77     17    0.8426684
##   ---
## 256:     77    158   -0.7093194
## 257:     77     24   -0.7181123
## 258:     77    244   -0.7745129
## 259:     77     75   -0.8067514
## 260:     77    186   -0.8202139
```

```r
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers),trial_store )
corr_nCustomers[order(-corr_measure)]
```

```
##      Store1 Store2 corr_measure
##       <num>  <num>        <num>
##   1:     77     77    1.0000000
##   2:     77    233    0.9903578
##   3:     77    119    0.9832666
##   4:     77    254    0.9162084
##   5:     77    113    0.9013480
##  ---
## 256:     77    102   -0.6525273
## 257:     77    147   -0.6569333
## 258:     77    169   -0.6663911
## 259:     77     54   -0.7606047
## 260:     77      9   -0.7856990
```

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

```r
#### Over to you! Create a combined score composed of correlation and magnitude, by first merging the c
#### Hint: 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 := 0.5 *
score_nSales[order(-scoreNSales)]
```

```
##      Store1 Store2 corr_measure mag_measure scoreNSales
##       <num>  <num>        <num>       <num>       <num>
##   1:     77     77    1.0000000   1.0000000  1.00000000
##   2:     77    233    0.9037742   0.9852649  0.94451954
##   3:     77     41    0.7832319   0.9651401  0.87418598
##   4:     77     50    0.7638658   0.9731293  0.86849757
##   5:     77     17    0.8426684   0.8806882  0.86167830
##  ---
## 256:     77    247   -0.6310496   0.5263807 -0.05233446
## 257:     77     24   -0.7181123   0.5908516 -0.06363035
## 258:     77    201   -0.4109081   0.2809523 -0.06497786
## 259:     77     55   -0.6667816   0.4693768 -0.09870241
## 260:     77     75   -0.8067514   0.3061880 -0.25028171
```

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

```
##      Store1 Store2 corr_measure mag_measure  scoreNCust
##       <num>  <num>        <num>       <num>       <num>
##   1:     77     77    1.0000000   1.0000000  1.00000000
##   2:     77    233    0.9903578   0.9927733  0.99156555
##   3:     77    254    0.9162084   0.9371312  0.92666979
##   4:     77     41    0.8442195   0.9746392  0.90942936
##   5:     77     84    0.8585712   0.9241818  0.89137652
##  ---
## 256:     77    147   -0.6569333   0.4991028 -0.07891525
## 257:     77    247   -0.6210342   0.4278646 -0.09658482
## 258:     77    227   -0.6237974   0.3923204 -0.11573851
## 259:     77     75   -0.5907354   0.3360498 -0.12734284
## 260:     77    102   -0.6525273   0.3968462 -0.12784056
```

- 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.

```
#### Over to you! Combine scores across the drivers by first merging our sales scores and customer scor
score_Control <- merge(score_nSales,score_nCustomers , by =c("Store1", "Store2") )
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]
```

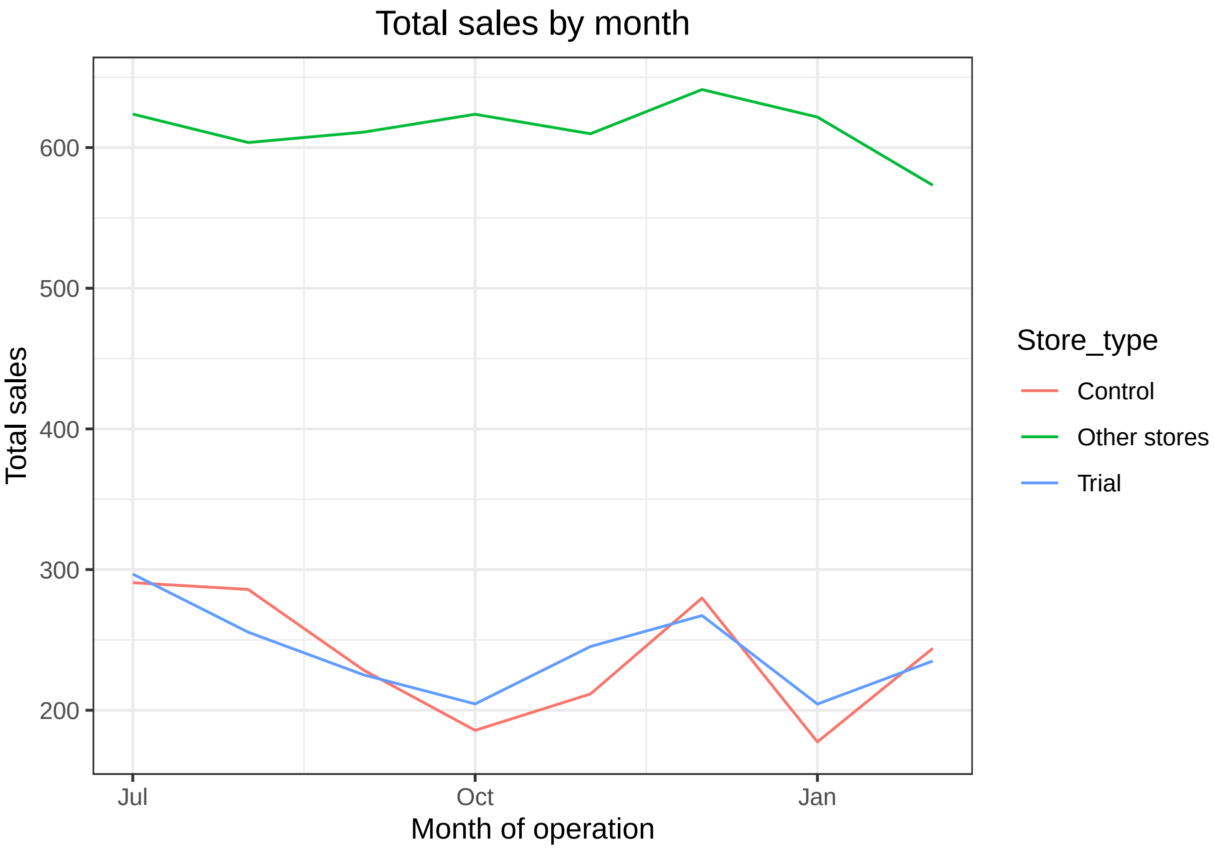
- The store with the highest score is then selected as the control store since it is most similar to the trial store.

```
#### Select control stores based on the highest matching store (closest to 1 but not the store itself,
control_store <- score_Control[Store1 == trial_store, ][order(-finalControlScore)][2, Store2]
control_store
```

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

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

```
#### Visual checks on trends based on the drivers
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial", ifelse(STOR

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

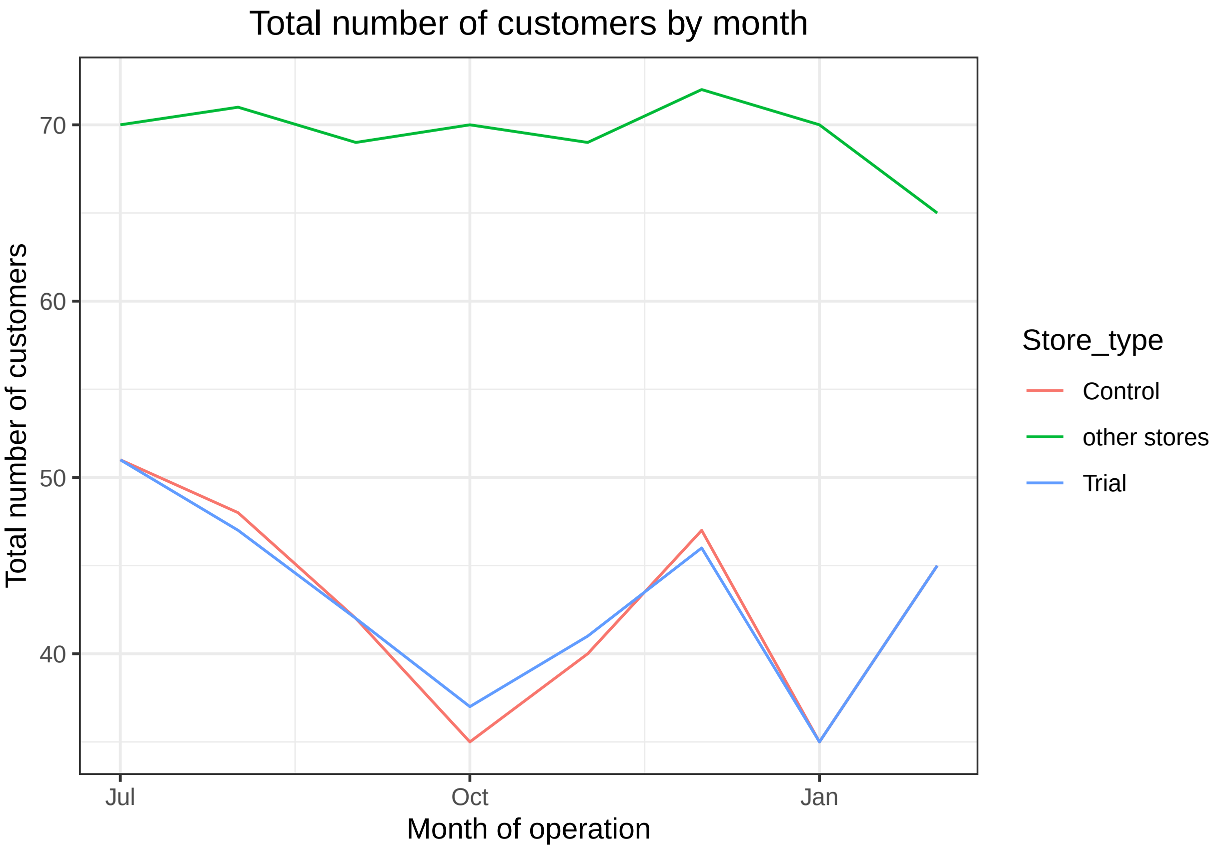


```
#### Over to you! Conduct visual checks on customer count trends by comparing the trial store to the co
#### Hint: Look at the previous plot.
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[,Store_type := ifelse(STORE_NBR == trial_store, "Trial",ifelse(ST
```

Next, number of customers.

```
## Warning in `[.data.table`(measureOverTimeCusts[, `:=`(Store_type,
## ifelse(STORE_NBR == : 70.750000 (type 'double') at RHS position 1
## out-of-range(NA) or truncated (precision lost) when assigning to type 'integer'
## (column 4 named 'nCustomers')
```

```
ggplot(pastCustomers, aes(x=TransactionMonth,y=nCustomers , color =Store_type)) +
  geom_line() +
  labs(x ="Month of operation" , y ="Total number of customers" , title ="Total number of customers by r
```



Assessment of trial

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
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(totSales)]/preTrialMeasures[STORE_NBR == control_store &
YEARMONTH < 201902, sum(totSales)]
#### Apply the scaling factor
measureOverTimeSales <- measureOverTime
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][ ,
controlSales := totSales * scalingFactorForControlSales]
```

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

```
#### Over to you! Calculate the percentage difference between scaled control sales and trial sales
percentageDiff <- merge(scaledControlSales[, c("YEARMONTH","controlSales")],measureOverTime[STORE_NBR ==
```

+ 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 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.
```

```
#### Over to you! Calculate the t-values for the trial months. After that, find the 95th percentile of
qt(0.95, df = degreesOfFreedom)
```

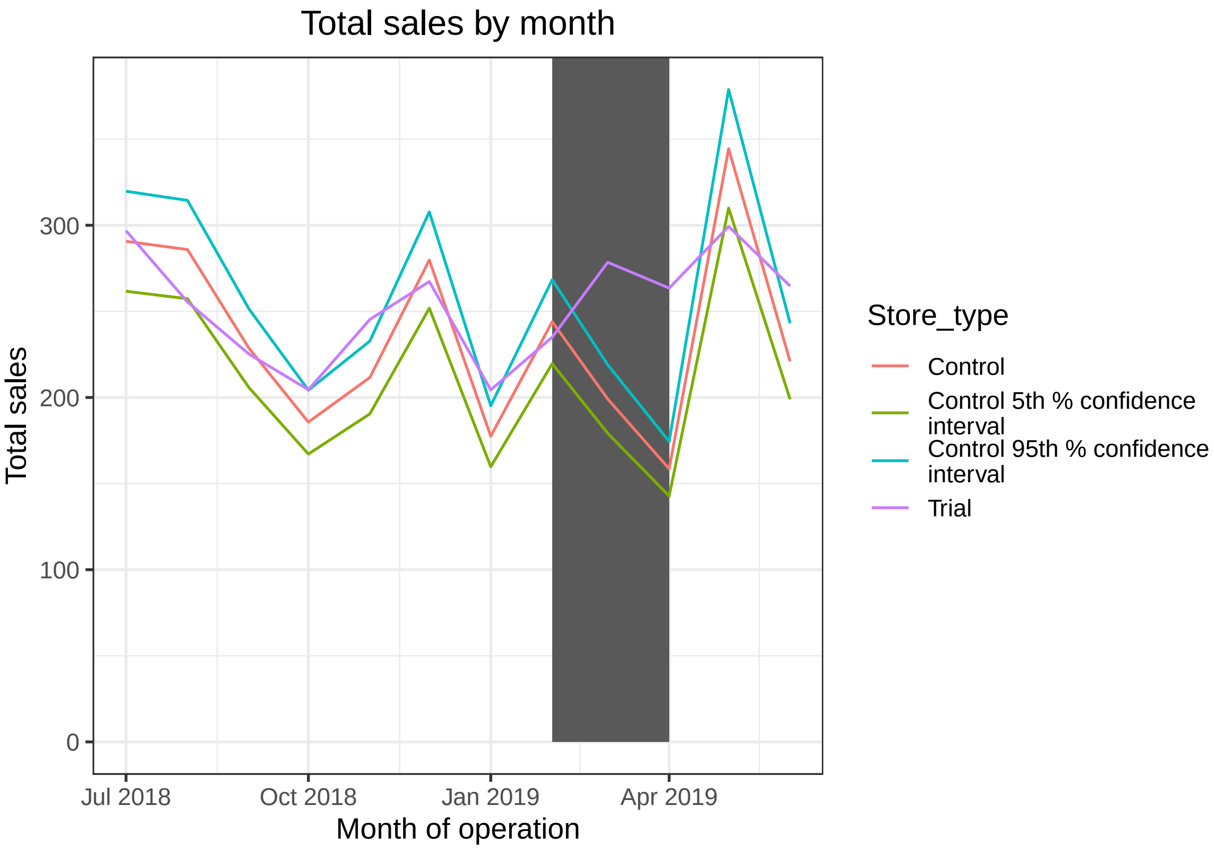
```
## [1] 1.894579
```

```
#### 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(YEARMONTH
```

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

- 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.
- 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_
][, 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, col
  geom_line() +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by month")
```



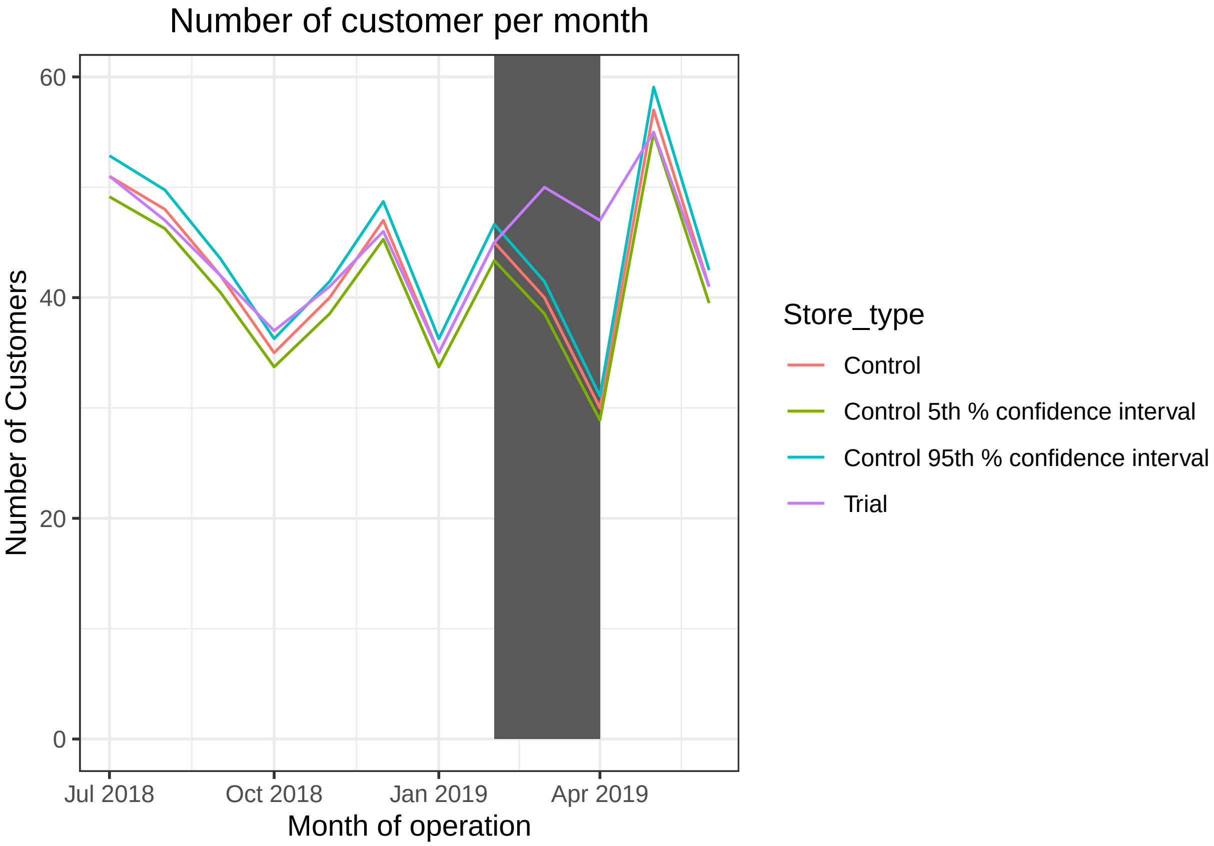
- The results show that the trial in store 77 is significantly different to its control store in the trial period as the trial store performance lies outside the 5% to 95% confidence interval of the control store in two of the three trial months. Let's have a look at assessing this for number of customers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

```
#### This would be a repeat of the steps before for total sales
#### Scale pre-trial control customers to match pre-trial trial store customers
#### Over to you! 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 custo
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum(nCust
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
]
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 the
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7
#### Trial and control store number of customers
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store, "Trial", ifelse(S
c("YEARMONTH", "Store_type")][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH%%100,1, s
#### Control store 95th percentile
pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",][, nCustomers := nCustomers * (1 + st
#### Control store 5th percentile
```

```
pastCustomers_Controls5 <- pastCustomers[Store_type == "Control", ][, nCustomers := nCustomers * (1 - st
trialAssessment <- rbind(pastCustomers, pastCustomers_Controls95,
pastCustomers_Controls5)
#### Over to you! Plot everything into one nice graph.
#### Hint: geom_rect creates a rectangle in the plot. Use this to highlight the trial period in our gra
ggplot(trialAssessment, aes(TransactionMonth,nCustomers, color = Store_type)) +
  geom_rect(data =trialAssessment[YEARMONTH < 201905 & YEARMONTH > 201901,] , aes(xmin =min(Transaction
  geom_line() +
  labs(x = "Month of operation", y = "Number of Customers", title = "Number of customer per month")
```



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

Trial store 86

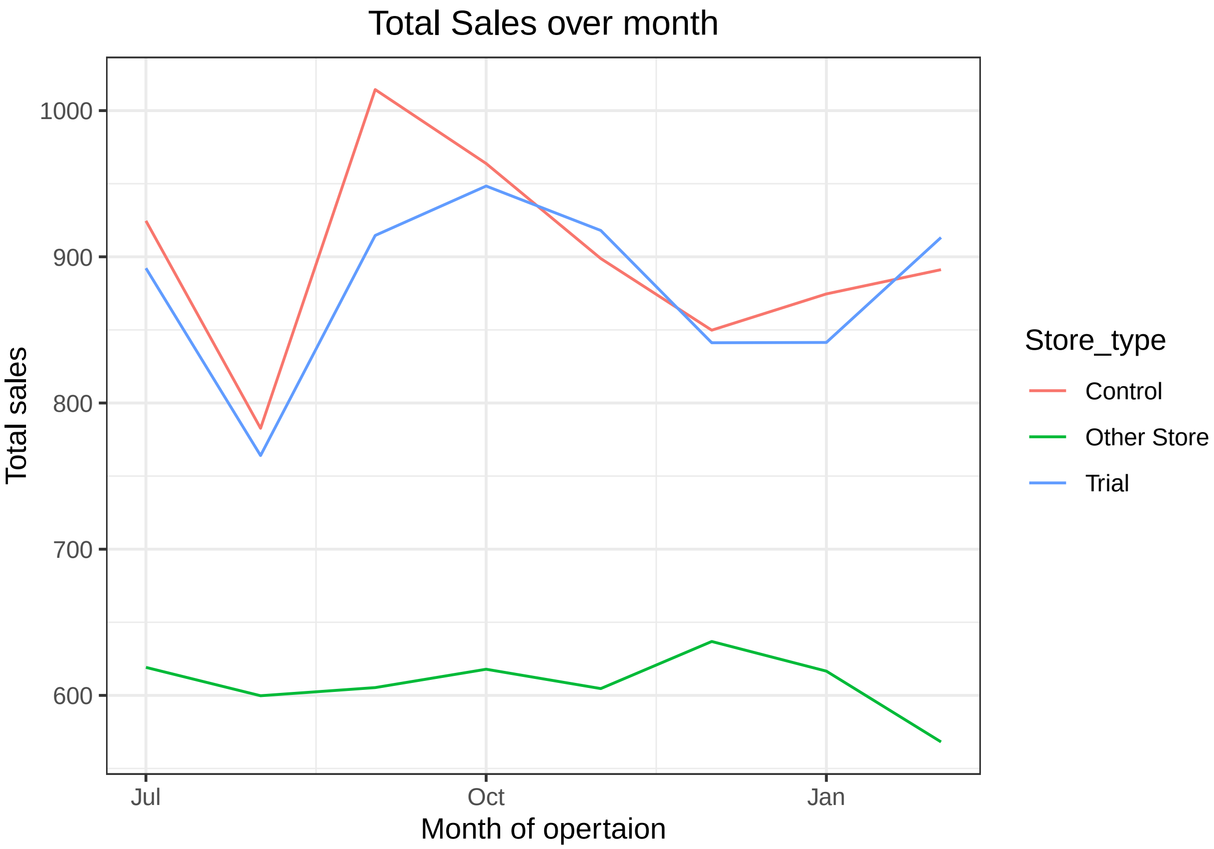
```
#### Over to you! Calculate the metrics below as we did for the first trial store.
measureOverTime <- data[, .(totSales = sum(TOT_SALES),
                            nCustomers = uniqueN(LYLTY_CARD_NBR),
                            nTxnPerCust = uniqueN(TXN_ID)/uniqueN(LYLTY_CARD_NBR),
                            nChipsPerTxn = sum(PROD_QTY)/ uniqueN(TXN_ID),
                            avgPricePerUnit =sum(TOT_SALES)/ sum(PROD_QTY)),
                        by = c("YEARMONTH", "STORE_NBR")][
                          order(YEARMONTH,STORE_NBR )]

#### Over to you! Use the functions we created earlier to calculate correlations and magnitude for each
trial_store <- 86
```

```r
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales), trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers), trial_store)
magnitude_nSales <-calculateMagnitudeDistance(preTrialMeasures, quote(totSales), trial_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures, quote(nCustomers), trial_store)

#### Now, create a combined score composed of correlation and magnitude
corr_weight <- 0.5
score_nSales <-merge(corr_nSales, magnitude_nSales, by = c("Store1", "Store2"))[, scoreNSales := corr_me
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = c("Store1", "Store2"))[,scoreNCust

#### Finally, combine scores across the drivers using a simple average.
score_Control <- merge(score_nSales, score_nCustomers, by = c("Store1","Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCustomers * 0.5]

#### 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 control store for trial store 86
control_store <- score_Control[Store1 == trial_store,
][order(-finalControlScore)][2, Store2]
control_store
```

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

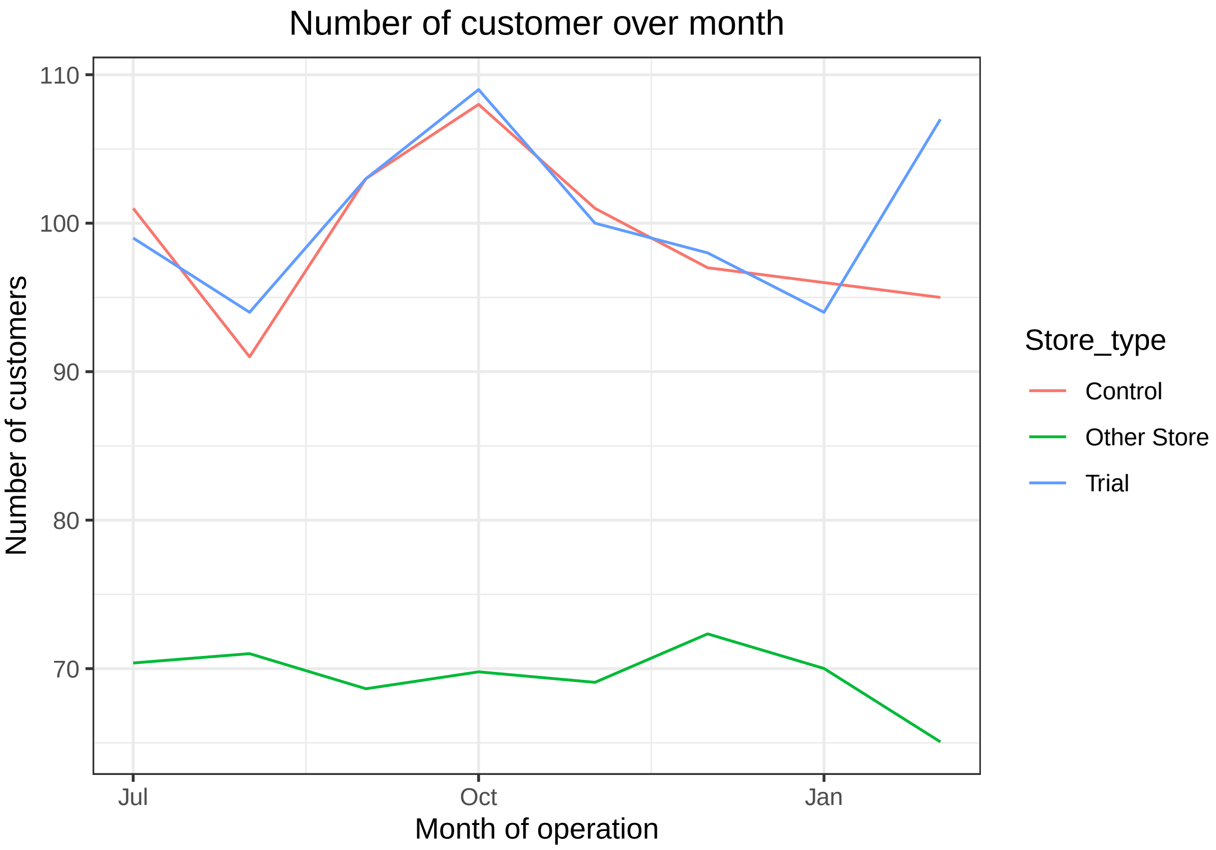
- Looks like store 155 will be a control store for trial store 86. Again, let's check visually if the drivers are indeed similar in the period before the trial. We'll look at total sales first.

```r
#### Over to you! Conduct visual checks on trends based on the drivers
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial", ifelse(STORE
][, 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 opertaion", y= "Total sales", title = "Total Sales over month")
```



- Great, sales are trending in a similar way. Next, number of customers.

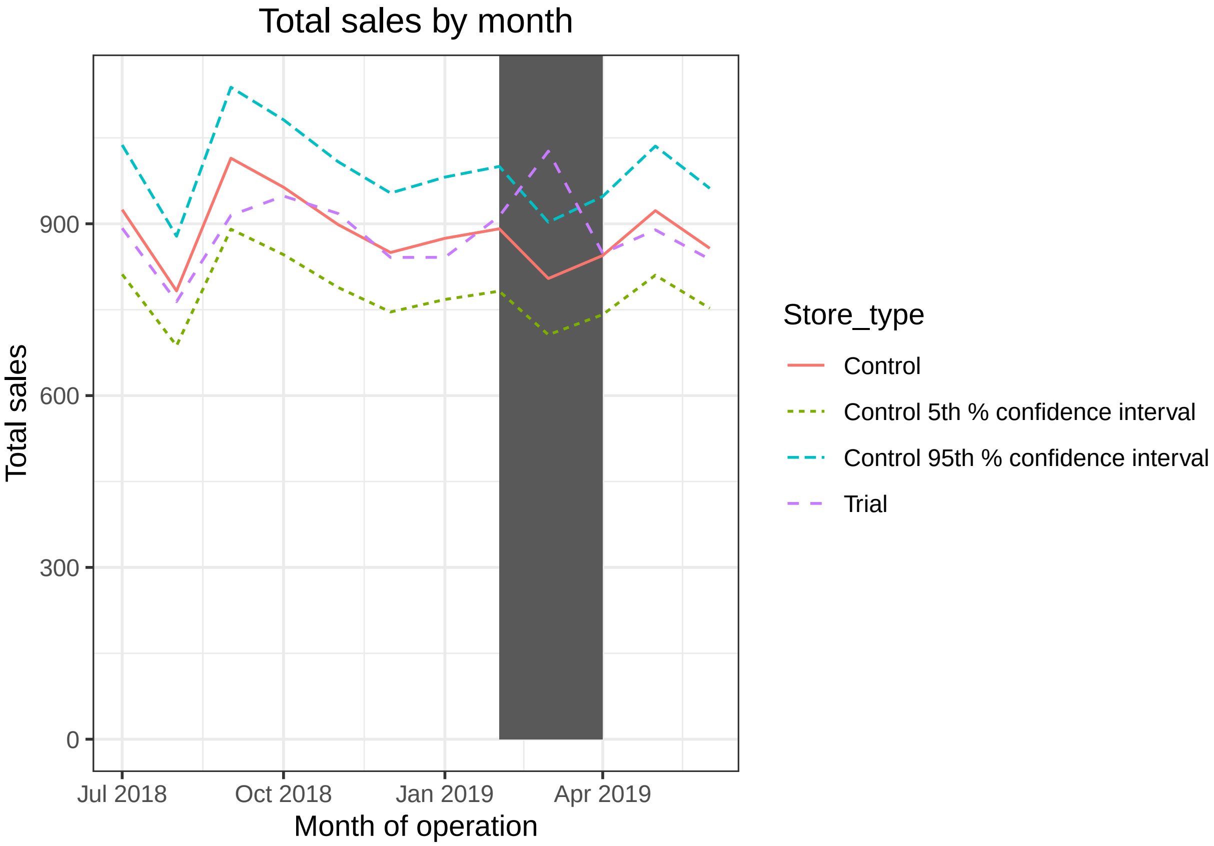
```
#### Over to you again! Conduct visual checks on trends based on the drivers
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store, "Trial", ifelse(S
][, numberCustomers := mean(nCustomers), by = c("YEARMONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"),"%Y-%m-%d")
][YEARMONTH < 201903 , ]
ggplot(pastCustomers, aes(TransactionMonth, numberCustomers, color = Store_type)) +
  geom_line() +
  labs(x ="Month of operation", y = "Number of customers", title = "Number of customer over month")
```



Good, the trend in number of customers is also similar. Let's now assess the impact of the trial on sales.

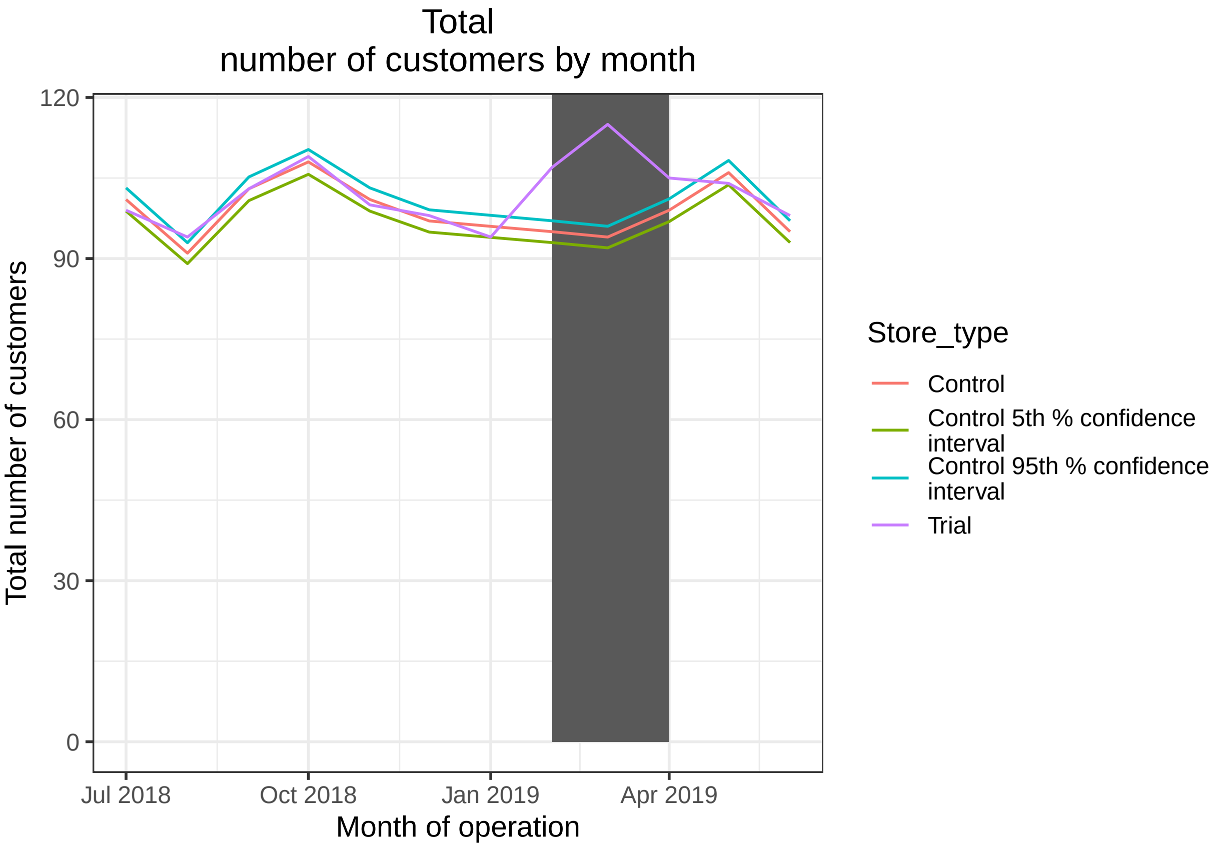
```
#### Scale pre-trial control sales to match pre-trial trial store sales
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(totSales)]/preTrialMeasures[STORE_NBR == control_store &
YEARMONTH < 201902, sum(totSales)]

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

#### Over to you! Calculate the percentage difference between scaled control sales and trial sales
#### Hint: When calculating percentage difference, remember to use absolute difference
percentageDiff <- merge( scaledControlSales[, c("controlSales", "YEARMONTH")],measureOverTimeSales[, c(
by = "YEARMONTH"
)[, percentageDiff := abs(controlSales - totSales)/ controlSales]

#### As our null hypothesis is that the trial period is the same as the pre-trial period, let's take th
#### Over to you! Calculate the standard deviation of percentage differences during the pre-trial perio
stdDev <- sd(percentageDiff[YEARMONTH < 201902,percentageDiff])
degreesOfFreedom <- 7

#### Trial and control store total sales
#### Over to you! 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(STOR
][, 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"),]

#### Over to you! 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
#### Hint2: Recall that the variable stdDev earlier calculates standard deviation in percentages, and n
pastSales_Controls95 <- pastSales[Store_type == "Control",
][, totSales := totSales * (1 + stdDev * 2)][, Store_type := "Control 95th % confidence interval"]
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_Cont
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")
```



- 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.

```r
#### 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")],
measureOverTime[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 th

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

#### 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")
```



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

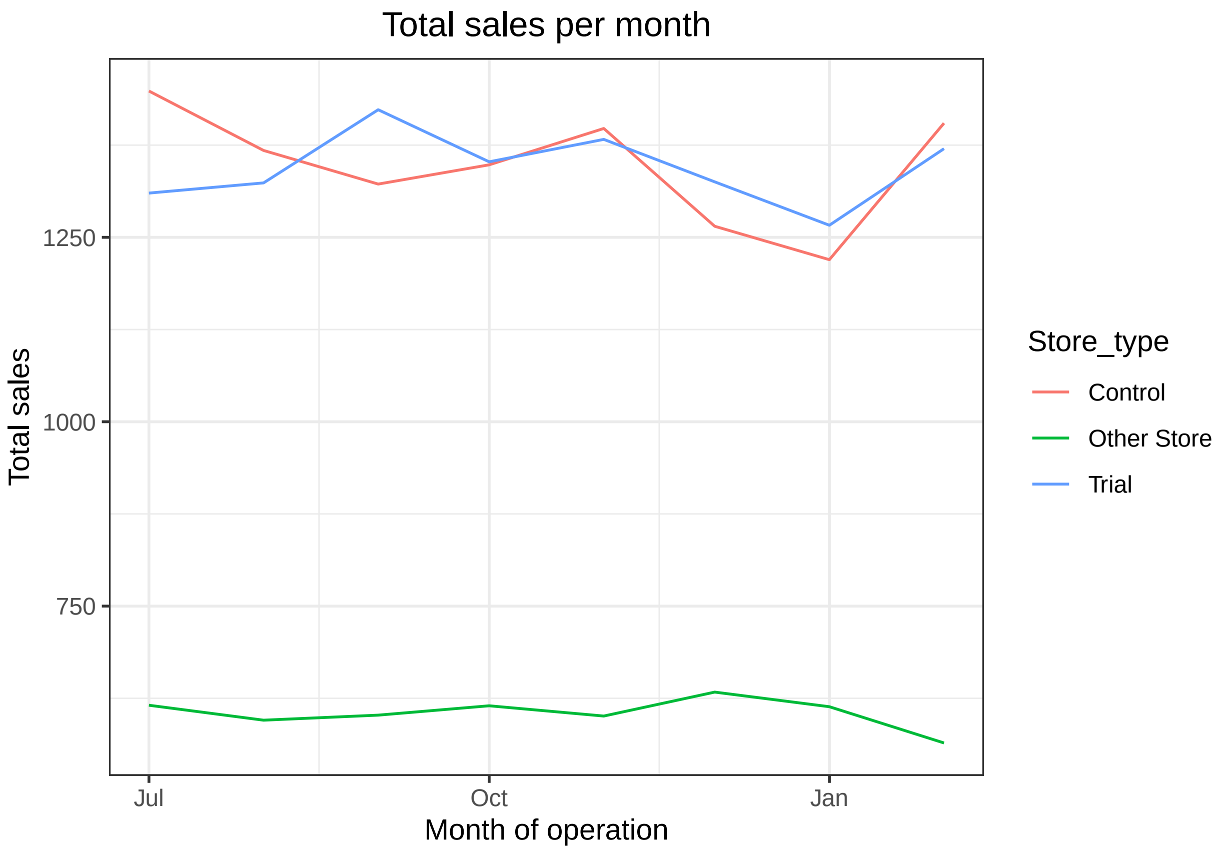
```
#### All over to you now! Your manager has left for a conference call, so you'll be on your own this ti
#### Conduct the analysis on trial store 88.
measureOverTime <- data[, .(totSales = sum(TOT_SALES),
                            nCustomers = uniqueN(LYLTY_CARD_NBR),
                            nTxnPerCust = uniqueN(TXN_ID)/uniqueN(LYLTY_CARD_NBR),
                            nChipesPerTxn = 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
#### Use the functions from earlier to calculate the correlation of the sales and number of customers o
trial_store <- 88
```

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

#### Use the functions from earlier to calculate the magnitude distance of the sales and number of custo
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales), trial_store)
magnitude_nCustomers <-calculateMagnitudeDistance(preTrialMeasures, quote(nCustomers), trial_store)

#### Create a combined score composed of correlation and magnitude by merging the correlations table an
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1", "Store2"))[, scoreNSales := corr_n
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = c("Store1", "Store2"))[, scoreNCus

#### Combine scores across the drivers by merging sales scores and customer scores, and compute a final
score_Control <- merge(score_nSales, score_nCustomers, by = c("Store1", "Store2"))
score_Control[, finalControlScore := scoreNSales * 2 + scoreNCustomers * 2]

#### 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 control store for trial store 88
control_store <- score_Control[Store1 == trial_store][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. We'll look at total sales first.

```
#### Visual checks on trends based on the drivers
#### For the period before the trial, create a graph with total sales of the trial store for each month
measureOverTimeSales <- measureOverTime
pastSales <-measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial", ifelse(STORE_
ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line() +
  labs(x ="Month of operation", y ="Total sales", title = "Total sales per month")
```

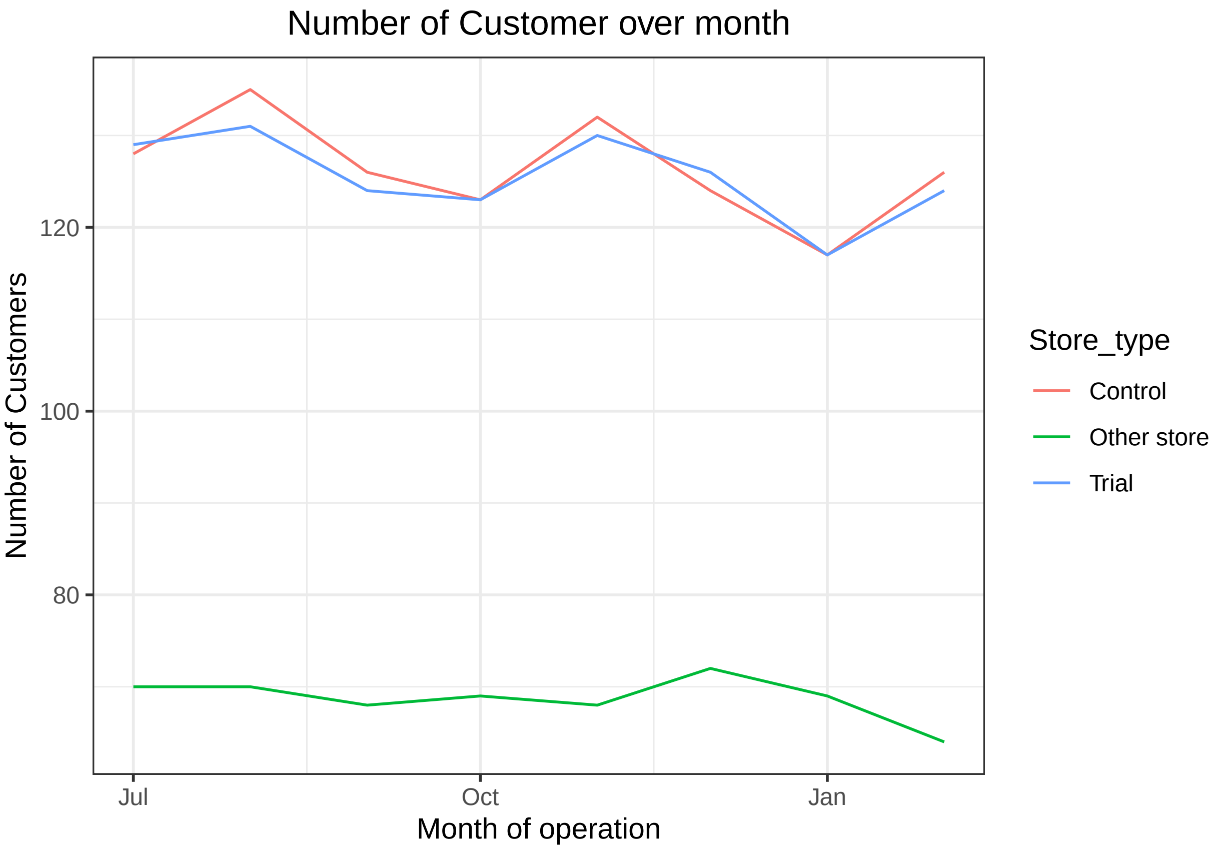


+Great, the trial and control stores have similar total sales. ### Next, number of customers.

```
#### Visual checks on trends based on the drivers
#### For the period before the trial, create a graph with customer counts of the trial store for each m
measureOverTimeCusts <- measureOverTime

pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store,"Trial", ifelse(S
```

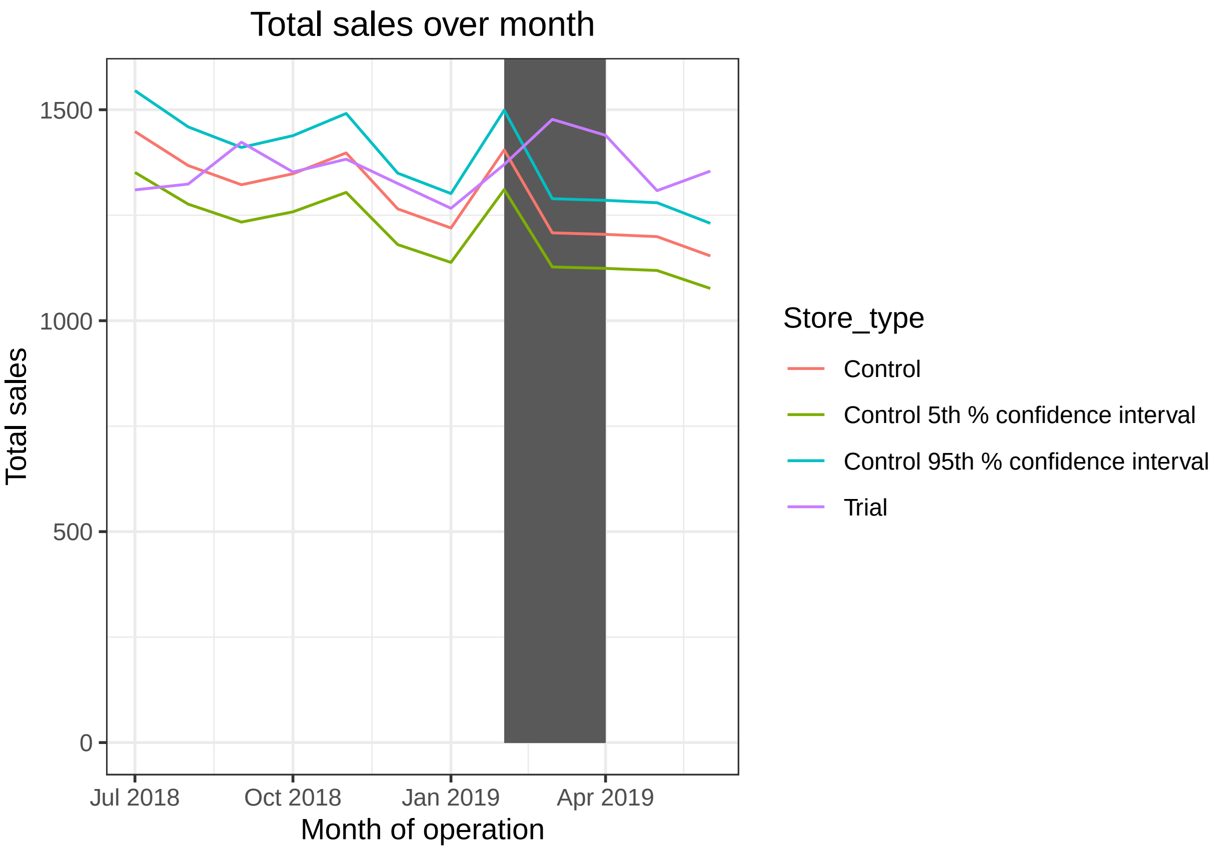
```
## Warning in `[.data.table`(measureOverTimeCusts[, `:=`(Store_type,
## ifelse(STORE_NBR == : 70.162879 (type 'double') at RHS position 1
## out-of-range(NA) or truncated (precision lost) when assigning to type 'integer'
## (column 4 named 'nCustomers')
```

```
ggplot(pastCustomers, aes(TransactionMonth, nCustomers, color = Store_type)) +
  geom_line() +
  labs(x ="Month of operation", y = "Number of Customers", title ="Number of Customer over month")
```



- Total number of customers of the control and trial stores are also similar. Let's now assess the impact of the trial on sales.

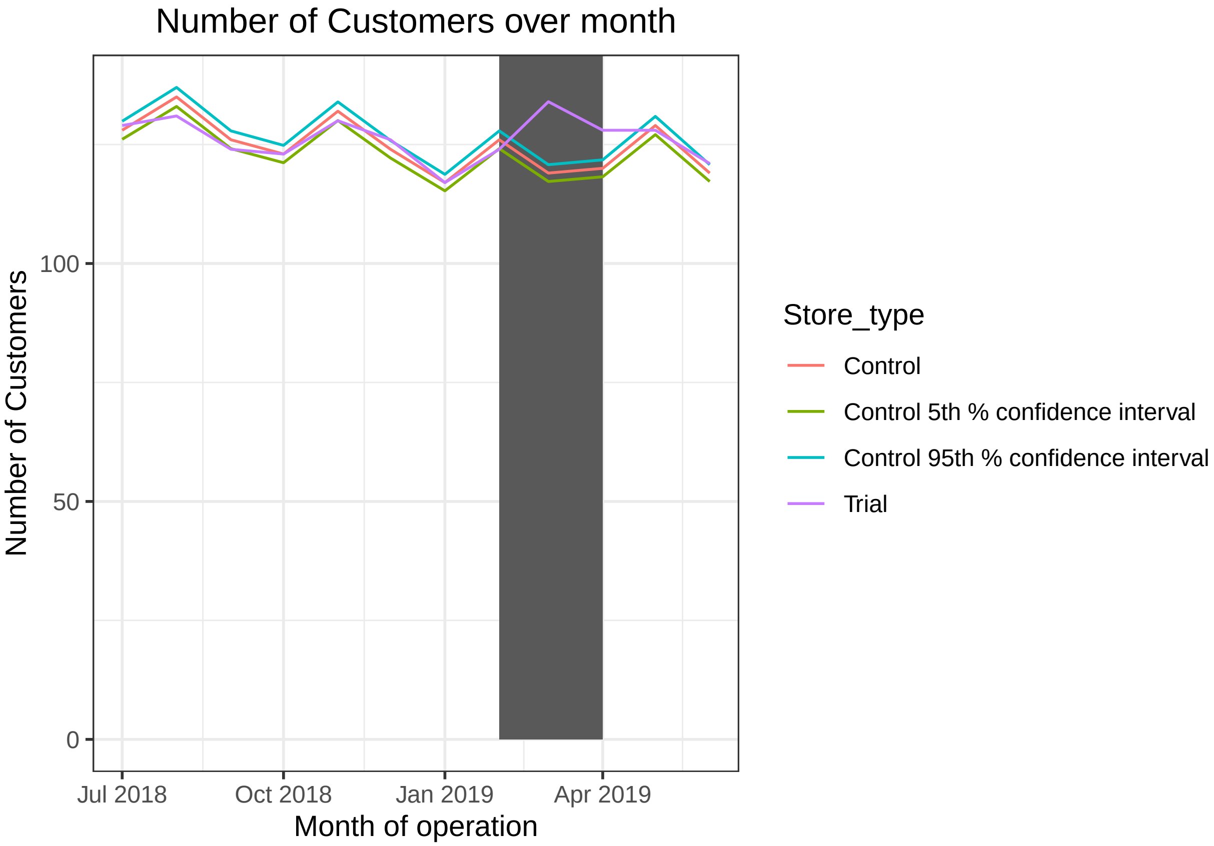
```
#### Scale pre-trial control store sales to match pre-trial trial store sales
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum(tot

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

#### Calculate the absolute percentage difference between scaled control sales and trial sales
percentageDiff <- merge(scaledControlSales[, c("YEARMONTH", "controlSales")], measureOverTimeSales[STOR

#### 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 <- 7

#### Trial and control store total sales

measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial", ifelse(STOR

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

#### Control store 5th percentile
pastSales_Controls5 <- pastSales[Store_type == "Control"][, totSales := totSales * (1 - stdDev * 2)][,
#### Combine the tables pastSales, pastSales_Controls95, pastSales_Controls5
trialAssessment <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)

#### Plot these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_rect(data = trialAssessment[YEARMONTH <201905 & YEARMONTH > 201901,], aes(xmin = min(Transaction
  geom_line() +
  labs(x ="Month of operation", y ="Total sales", title = "Total sales over month")
```



- The results show that the trial in store 88 is significantly different to its control store in the trial period 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 store customers to match pre-trial trial store customers
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum(nCus

#### Apply the scaling factor
measureOverTimeCusts <- measureOverTime
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store,][, controlCustomers := nCust

#### Calculate the absolute percentage difference between scaled control sales and trial sales
percentageDiff <- merge(scaledControlCustomers[, c ("YEARMONTH", "controlCustomers")], measureOverTimeC

#### 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 <- 7
```

```
# note that there are 8 months in the pre-trial period hence 8 - 1 = 7 degrees of freedom
#### Trial and control store number of customers
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store, "Trial", ifelse(

#### Control store 95th percentile
pastCustomers_Controls95 <-pastCustomers[Store_type == "Control"][, nCustomers := nCustomers * (1 + std

#### Control store 5th percentile
pastCustomers_Controls5 <- pastCustomers[Store_type == "Control"][, nCustomers := nCustomers * (1 - std

#### Combine the tables pastSales, pastSales_Controls95, pastSales_Controls5
trialAssessment <- rbind(pastCustomers, pastCustomers_Controls95, pastCustomers_Controls5)

#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, nCustomers, color = Store_type)) +
  geom_rect(data = trialAssessment[YEARMONTH < 201905 & YEARMONTH >201901,], aes( xmin = min(Transactio
  geom_line() +
  labs(x = "Month of operation", y = "Number of Customers", title = "Number of Customers over 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

- 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.