Quantium Virtual Internship - Retail Strategy and Analytics - Task 2

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.

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.

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

```
# If you are on a Windows machine, you will need to use forward slashes (/) instead of backshashes (\)
#file.choose()
data <- fread(paste0("D:\\UMP\\Extra Program\\Virtual Internship (Forage)\\Quantium\\Task 2\\QVI_data.csv"))
head(data)
```

```
tail(data)
```

```
str(unique(data))
```

```
## Classes 'data.table' and 'data.frame': 264833 obs. of 12 variables:
## $ LYLTY_CARD_NBR : int 1000 1002 1003 1003 1004 1005 1007 1007 1009 1010 ...
## $ DATE : IDate, format: "2018-10-17" "2018-09-16" ...
## $ 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 150g" "Grain Waves Sour Cream&Chives 210G" "Natural ChipCo
Hony Soy Chckn175g" ...
## $ PROD_QTY : int 2 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 FAMILIES" ...
## $ PREMIUM_CUSTOMER: chr "Premium" "Mainstream" "Budget" "Budget" ...
## - attr(*, ".internal.selfref")=<externalptr>
```

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

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 (totSales) - Monthly number of customers (nCust) - Monthly number of transactions per customer (nTxnPerCust)

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
data$DATE <- as.Date(data$DATE, origin = "1899-12-30")
setDT(data)

#### Add a new month ID column in the data with the format yyyymm.
data[, YEARMONTH := format(DATE, "%Y%m")]
data
```

```
#### Next, we define the measure calculations to use during the analysis.
# For each store and month calculate total sales, number of customers, transactions per customer, chips
per customer and the average price per unit.
## Hint: you can use uniqueN() to count distinct values in a column
measureOverTime <- data[, .(totSales = sum(TOT_SALES, na.rm = TRUE),
                            nCust = uniqueN(LYLTY_CARD_NBR),
                            nTxnPerCust = round(uniqueN(TXN_ID) / uniqueN(LYLTY_CARD_NBR), 2),
                            nChipsPerTxn = round(sum(PROD_QTY, na.rm = TRUE) / uniqueN(TXN_ID), 2),
                            avgPricePerUnit = round(sum(TOT_SALES) / sum(PROD_QTY), 2)),
                        by = .(STORE_NBR, YEARMONTH)][order(STORE_NBR, YEARMONTH)]
measureOverTime
```

```
#### Filter to the pre-trial period and stores with full observation periods
#### [count entry per store][check full observations (12 months)]
storesWithFullObs <- unique(measureOverTime[, .N, STORE_NBR][N == 12, STORE_NBR])
preTrialMeasures <- measureOverTime[YEARMONTH < 201902 & STORE_NBR %in% storesWithFullObs, ]
preTrialMeasures
```

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.

```
#### 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.
calcCorr <- function(inputTable, metricCol, storeComparison) {
  # Initialize an empty data.table for storing results
  calcCorrTable = data.table(Store1 = numeric(),
                             Store2 = numeric(),
                             corr_measure = numeric())

  # Get unique store numbers
  storeNum <- unique(inputTable[, STORE_NBR])

  # Loop through each store number to calculate correlation
  for (i in storeNum) {
    calcMeasure = data.table("Store1" = storeComparison,
                             "Store2" = i,
                             "corr_measure" = cor(inputTable[STORE_NBR == storeComparison, eval(metricCo
l)],
                                                  inputTable[STORE_NBR == i, eval(metricCol)]))

    # Append the results to the results table
    calcCorrTable <- rbind(calcCorrTable, calcMeasure, fill = TRUE)
    }
  return(calcCorrTable)
}
```

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

```
#### Create a function to calculate a standardised magnitude distance for a measure, looping through eac
h control store
calcMagnitudeDist <- function(inputTable, metricCol, storeComparison) {
  # Initialize an empty data.table for storing results
  calcDistTable = data.table(Store1 = numeric(),
                             Store2 = numeric(),
                             YEARMONTH = numeric(),
                             measure = numeric())

  # Get unique store numbers
  storeNum <- unique(inputTable[, STORE_NBR])

  # Loop through each store number to calculate magnitude distance
  for (i in storeNum) {
    calcMeasure = 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)]))

    # Append the results to the results table
    calcDistTable <- rbind(calcDistTable, calcMeasure, fill = TRUE)
    }

  #### Standardise the magnitude distance so that the measure ranges from 0 to 1
  # Calculate min and max of 'measure' for each (Store1, YEARMONTH) group
  minMaxDist <- calcDistTable[ ,
                               .(minDist = min(measure),
                                 maxDist = max(measure)),
                               by = c("Store1", "YEARMONTH")]

  # Merge the min-max data back with the original distances
  distTable <- merge(calcDistTable,
                     minMaxDist,
                     by = c("Store1", "YEARMONTH"))

  # Calc the standardized magnitude measure (scaled to 0-1)
  distTable[, magnitudeMeasure := 1 - (measure - minDist)/(maxDist - minDist)]

  # Calc the average standardized magnitude measure for each store pair
  finalDistTable <- distTable[ ,
                               .(mag_measure = mean(magnitudeMeasure)),
                               by = .(Store1, Store2)]

  # Return the final result
  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.

```
trial_store <- 77

corr_nSales <- calcCorr(preTrialMeasures, quote(totSales), trial_store)
corr_nCust <- calcCorr(preTrialMeasures, quote(nCust), trial_store)

#### Then, use the functions for calculating magnitude.
magnitude_nSales <- calcMagnitudeDist(preTrialMeasures, quote(totSales), trial_store)
magnitude_nCust <- calcMagnitudeDist(preTrialMeasures, quote(nCust), 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 ta
ble with the magnitude table.

#### 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 := (corr_
measure + mag_measure) * 0.5]
score_nSales[order(-scoreNSales)]
```

```
score_nCustomers <- merge(corr_nCust, magnitude_nCust, by = c("Store1", "Store2"))[, scoreNCust := (corr
_measure + mag_measure) * 0.5]
score_nCustomers[order(-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
le table
score_Control <- merge(score_nSales, score_nCustomers, by = c("Store1", "Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]

score_Control[order(-finalControlScore)]
```

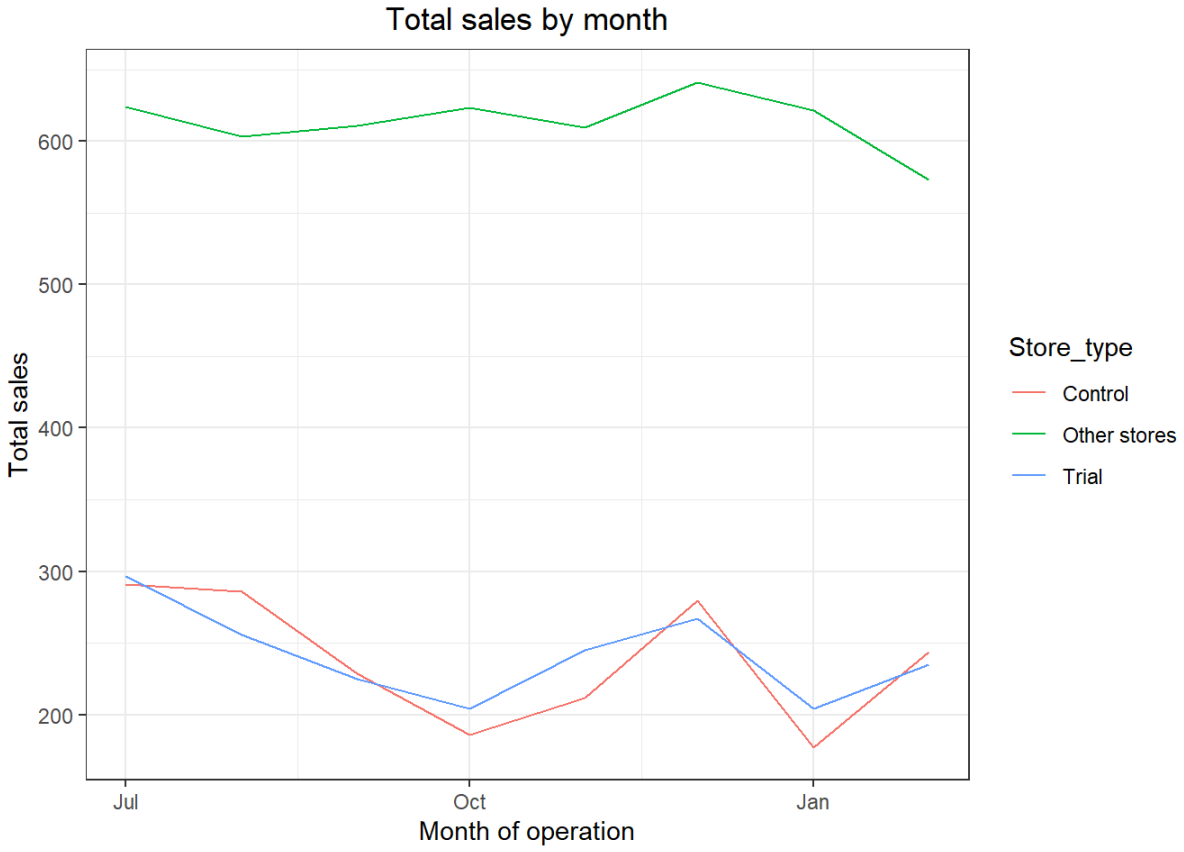
The store with the highest score is then selected as the control store since it is most similar to the trial store.

```
#### 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[Store1 == trial_store, ][order(-finalControlScore)][2, Store2]
control_store
```

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

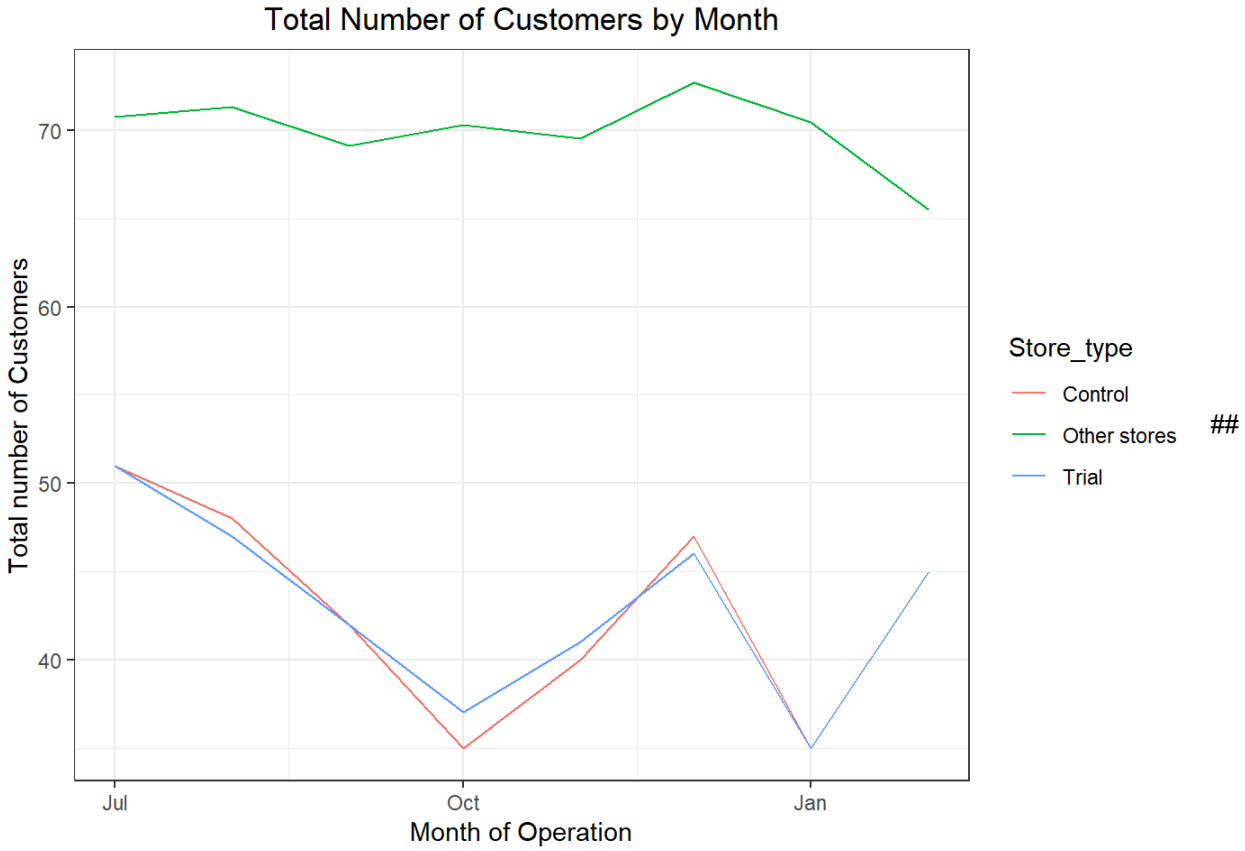
Now that we have found a control store, let's check visually if the drivers are indeed similar in the period before the trial. We'll look at total sales first.

```
#### Visual checks on trends based on the drivers
measureOverTimeSales <- measureOverTime

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

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



Next, number of customers.

```
#### visual checks on customer count trends by comparing the trial store to the control store and other
stores.

#### Hint: Look at the previous plot.
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                             ifelse(STORE_NBR == control_store, "Control",
"Other stores"))
                                      ][, numCust := mean(nCust), by = c("YEARMONTH", "Store_type")
                                        ][, YEARMONTH := as.numeric(YEARMONTH)  # Convert YEARMONTH to n
umeric
                                          ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
                                                                                YEARMONTH %% 100,
                                                                                1,
                                                                                sep = "-"),
                                                                          "%Y-%m-%d")
                                            ][YEARMONTH < 201903, ]


ggplot(pastCustomers, aes(TransactionMonth, numCust, color = Store_type)) +
  geom_line() +
  labs(x = "Month of Operation",
       y = "Total number of Customers",
       title = "Total Number of Customers by Month")
```



##

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(totS
ales)] /
  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.

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

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


#### We will test with a null hypothesis of there being 0 difference between trial and control stores.
#### Calculate the t-values for the trial months. After that, find the 95th percentile of the t-distribu
tion with the appropriate degrees of freedom to check whether the hypothesis is statistically significan
t.
#### Hint: The test statistic here is (x - u)/standard deviation
percentageDiff[, tValue := (percentageDiff - 0) / stdDev
               ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"),
                                               "%Y-%m-%d")
               ][YEARMONTH < 201905 & YEARMONTH > 201901, .(TransactionMonth, tValue)]
```

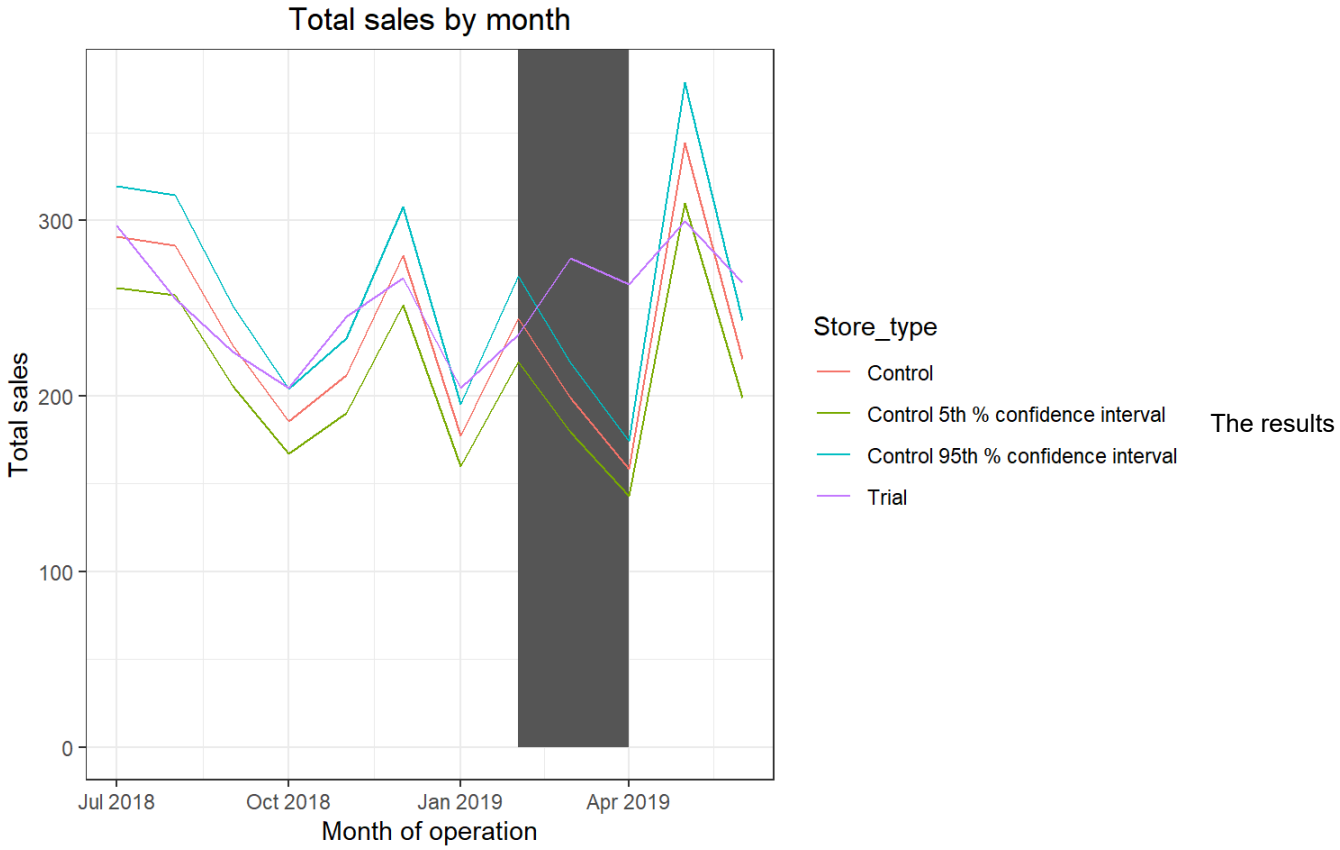
```
#### Find the 95th percentile of the t-distribution with the appropriate
#### degrees of freedom to compare against
qt(0.95, df = degreesOfFreedom)
```

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

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

Let's create a more visual version of this by plotting the sales of the control store, the sales of the trial stores and the 95th percentile value of sales of the control store.

```
measureOverTimeSales <- measureOverTime

#### Trial and control store total sales
#### Create new variables Store_type, totSales and TransactionMonth in the data table.
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                         ifelse(STORE_NBR == control_store, "Control",
"Other stores"))
                              ][, totSales := mean(totSales), by = c("YEARMONTH", "Store_type")
                              ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100,
1, sep = "-"),
                                                              "%Y-%m-%d")
                              ][Store_type %in% c("Trial", "Control"), ]


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


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


#### Plotting these in one nice graph
ggplot(trialAssessment,
       aes(TransactionMonth, totSales, color = Store_type)) +
  geom_rect(data = trialAssessment[YEARMONTH < 201905 & YEARMONTH > 201901 ,],
            aes(xmin = min(TransactionMonth),
                xmax = max(TransactionMonth),
                ymin = 0 , ymax = Inf,
                color = NULL),
            show.legend = FALSE) +
  geom_line() +
  labs(x = "Month of operation",
       y = "Total sales",
       title = "Total sales by month")
```



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.

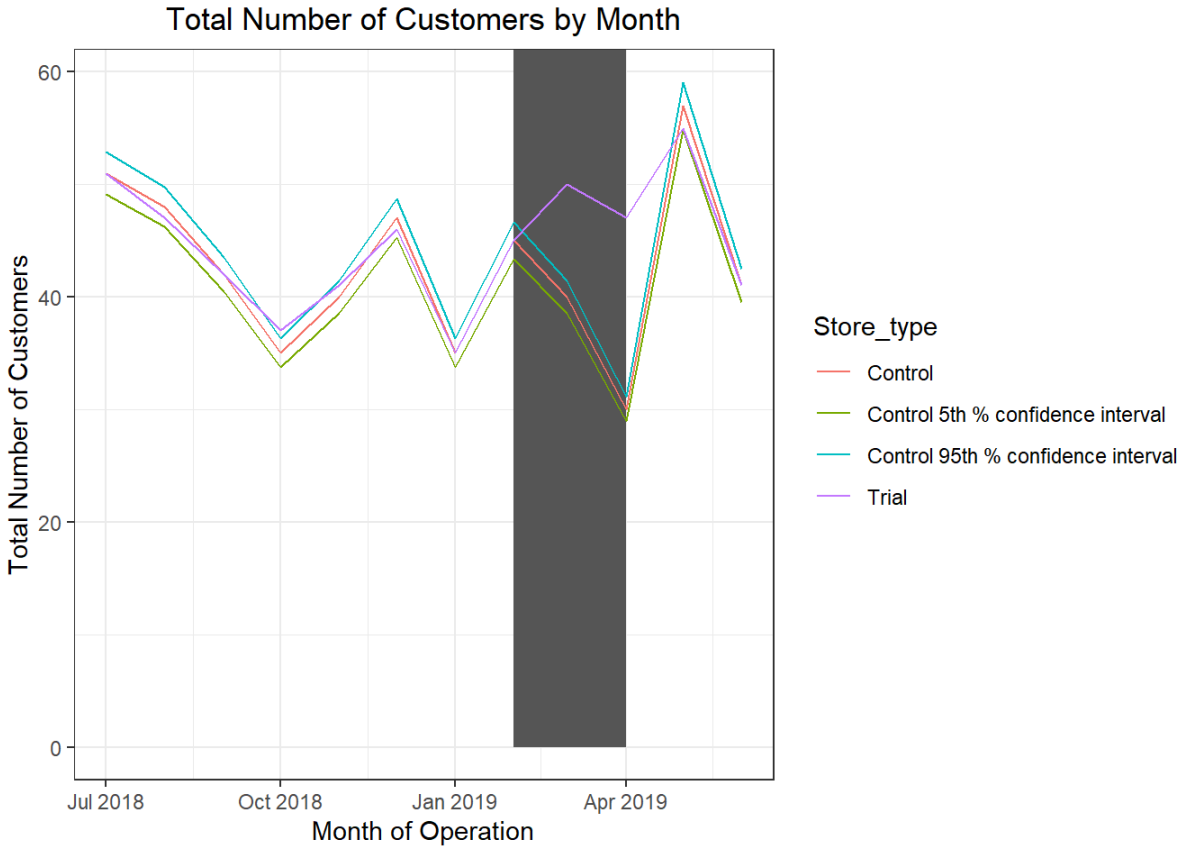
```
#### This would be a repeat of the steps before for total sales scale pre-trial control customers to mat
ch pre-trial trial store customers
#### Compute a scaling factor to align control store customer counts to our trial store.
#### Then, apply the scaling factor to control store customer counts.
#### Finally, calculate the percentage difference between scaled control store customers and trial custo
mers.
scalingFactorForControlCust <- preTrialMeasures[
  STORE_NBR == trial_store & YEARMONTH < 201902, sum(nCust)] /
  preTrialMeasures[STORE_NBR == control_store & YEARMONTH < 201902, sum(nCust)]


measureOverTimeCusts <- measureOverTime
measureOverTime
```

```
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store,
                                               ][, controlCustomers := nCust * scalingFactorForControlCu
st
                                                 ][, Store_type := ifelse(STORE_NBR == trial_store, "Tri
al",
                                                                          ifelse(STORE_NBR == control_st
ore, "Control", "Other stores"))]


percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH", "controlCustomers")],
                        measureOverTime[STORE_NBR == trial_store, c("nCust", "YEARMONTH")], by = "YEARMO
NTH")[, percentageDiff := abs(controlCustomers - nCust) / controlCustomers]
percentageDiff
```

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
standard deviation based on the scaled percentage difference in the pre-trial period
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])

degreesOfFreedom <- 7

#### Trial and control store number of customers
pastCustomers <- measureOverTimeCusts[, nCusts := mean(nCust),
                                      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)
trialAssessment
```

```
#### Plot everything into one nice graph.
#### Hint: geom_rect creates a rectangle in the plot. Use this to highlight the trial period in our grap
h.

ggplot(trialAssessment, aes(TransactionMonth, nCusts, colour = Store_type)) +
  geom_rect(data = trialAssessment[YEARMONTH > 201901 & YEARMONTH < 201905, ],
                       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")
```



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),
                            nCust = 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("STORE_NBR", "YEARMONTH")][order(STORE_NBR, YEARMONTH)]


#### Use the functions we created earlier to calculate correlations and magnitude for each potential con
trol store
trial_store <- 86

corr_nSales <- calcCorr(preTrialMeasures, quote(totSales), trial_store)
corr_nCustomers <- calcCorr(preTrialMeasures, quote(nCust), trial_store)

magnitude_nSales <- calcMagnitudeDist(preTrialMeasures, quote(totSales), trial_store)
magnitude_nCustomers <- calcMagnitudeDist(preTrialMeasures, quote(nCust), 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")
                      )[, score_NSales := (corr_measure + mag_measure) * 0.5]
score_nCustomers <- merge(corr_nCustomers,
                          magnitude_nCustomers,
                          by = c("Store1", "Store2")
                          )[, score_NCustomers := (corr_measure + mag_measure) * 0.5]


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

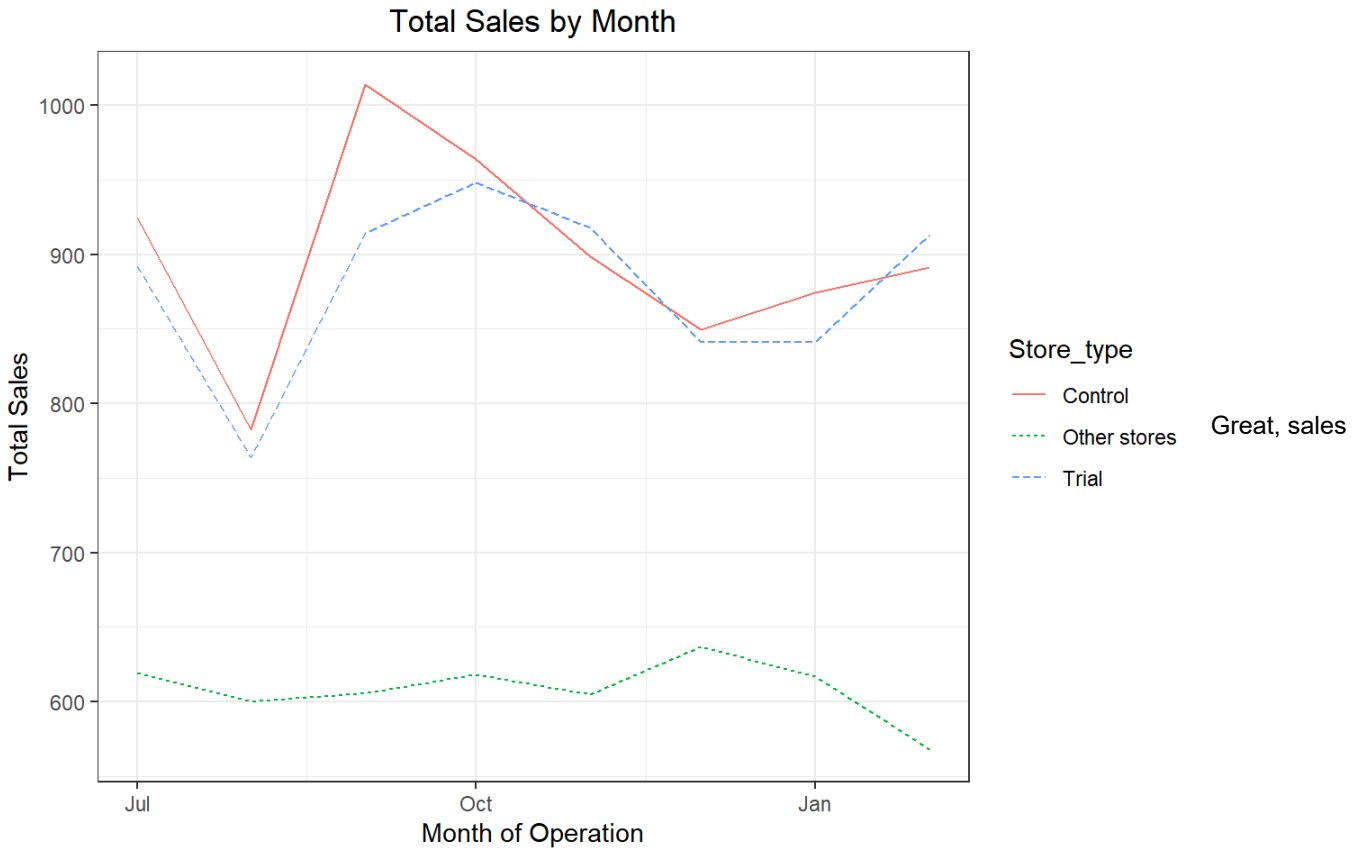
```
## [1] 155
```

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.

```
#### Conduct visual checks on trends based on the drivers
measureOverTimeSales <- measureOverTime[, YEARMONTH := as.numeric(YEARMONTH)]

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

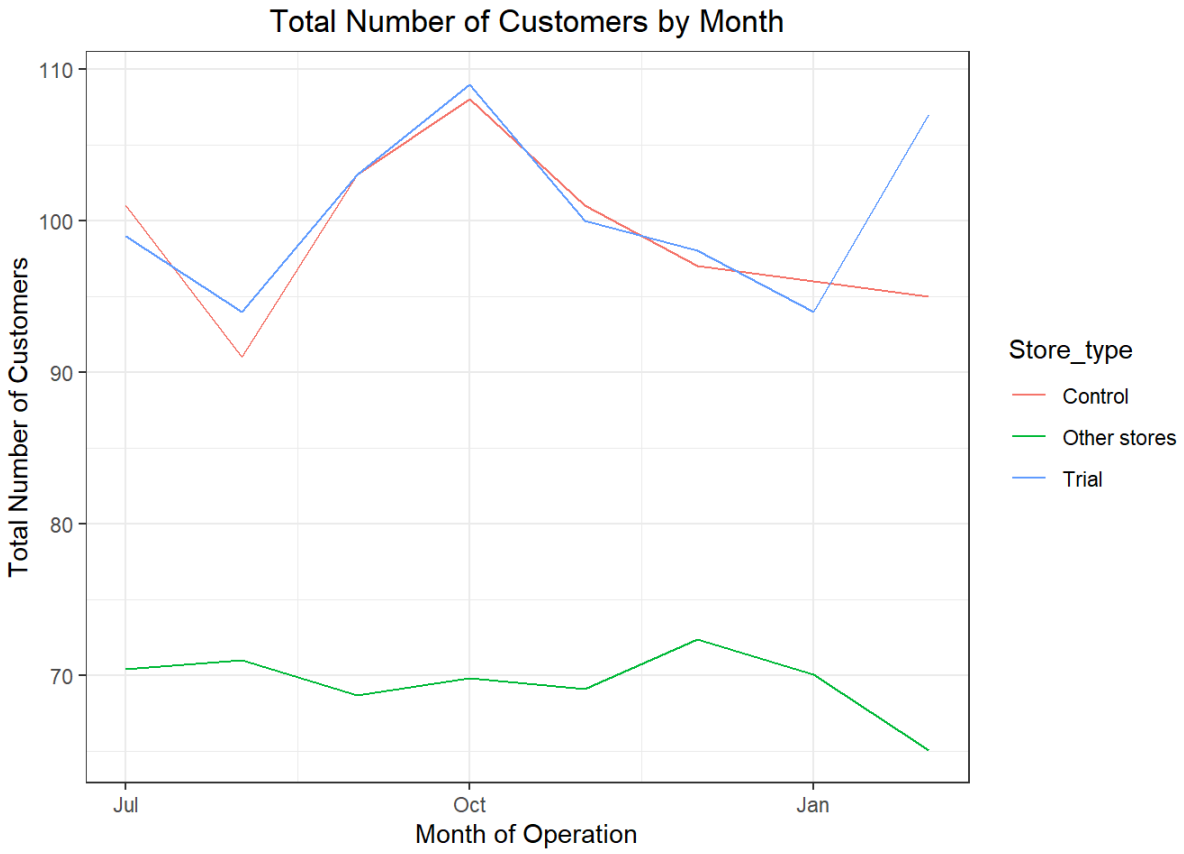
```
ggplot(pastSales,
       aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line(aes(linetype = Store_type)) +
  labs(x = "Month of Operation", y = "Total Sales", title = "Total Sales by Month")
```



are trending in a similar way.

Next, number of customers.

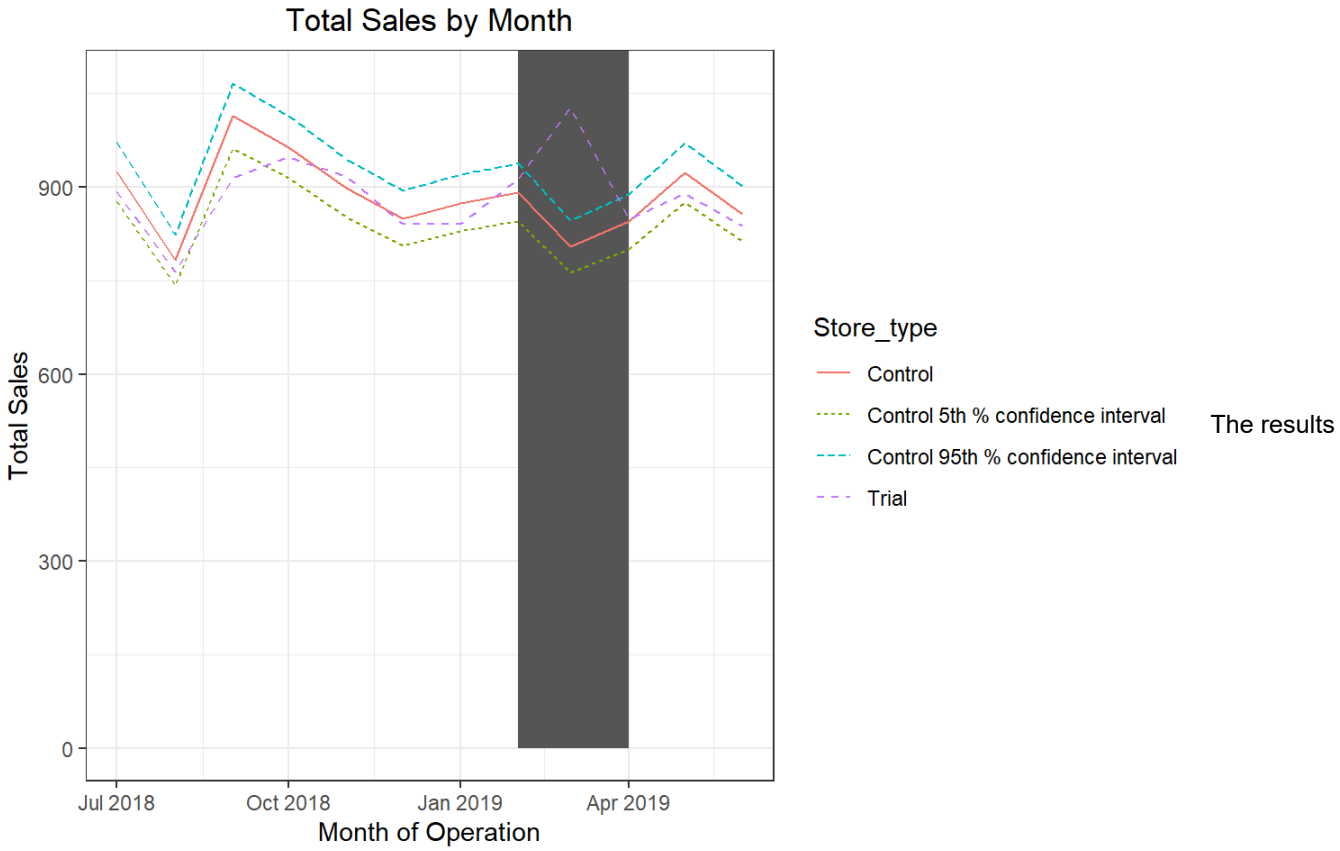
```
#### Conduct visual checks on trends based on the drivers
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                             ifelse(STORE_NBR == control_store, "Contro
l", "Other stores"))
                                      ][, numberCustomers := mean(nCust), by = c("YEARMONTH", "Store_typ
e")
                                        ][, 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 = "Total Number of Customers",
       title = "Total Number of Customers by 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(totS
ales)
                                                 ] / preTrialMeasures[STORE_NBR == control_store & YEARM
ONTH < 201902, sum(totSales)]

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

#### Calculate the percentage difference between scaled control sales and trial sales
#### Hint: When calculating percentage difference, remember to use absolute difference
percentageDiff <- merge(scaledControlSales[, c("YEARMONTH", "controlSales")],
                        measureOverTime[STORE_NBR == trial_store],
                        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 the
standard deviation based on the scaled percentage difference in the pre-trial period
#### Over to you! Calculate the standard deviation of percentage differences during the pre-trial period
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(STORE
_NBR == control_store, "Control", "Other stores"))
                             ][, totSales := mean(totSales), by = c("YEARMONTH", "Store_type")
                             ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100,
1, sep = "-"),
                                                             "%Y-%m-%d")
                             ][Store_type %in% c("Trial", "Control"), ]

#### 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 t
he mean.
#### Hint2: Recall that the variable stdDev earlier calculates standard deviation in percentages, and no
t dollar sales.
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_Contr
ols5
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 ,
```

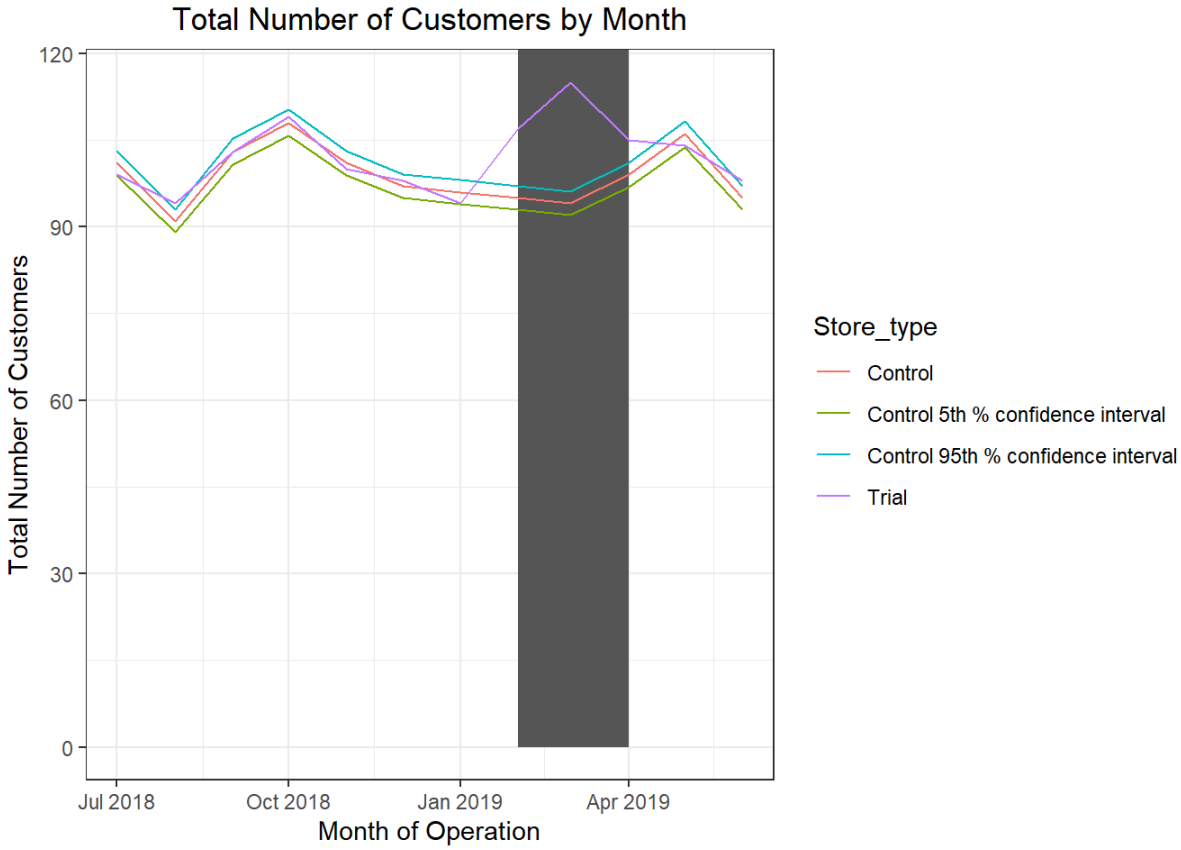
```
              ymax = Inf,
              color = NULL),
          show.legend = FALSE) +
geom_line(aes(linetype = Store_type)) +
labs(x = "Month of Operation",
     y = "Total Sales",
     title = "Total Sales by Month")
```



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.

```
#### 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(nCus
t)
                                                ] / preTrialMeasures[STORE_NBR == control_store & YEARMO
NTH < 201902, sum(nCust)]

#### Apply the scaling factor
measureOverTimeCusts <- measureOverTime
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store,
                                               ][ , controlCustomers := nCust * scalingFactorForControlC
ust
                                                  ][, Store_type := ifelse(STORE_NBR == trial_store, "Tr
ial",
                                                                           ifelse(STORE_NBR == control_s
tore, "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("nCust", "YEARMONTH")], by = "YEARMO
NTH"
                        )[, percentageDiff := abs(controlCustomers-nCust)/controlCustomers]

#### As our null hypothesis is that the trial period is the same as the pre-trial period, let's take the
standard deviation based on the scaled percentage difference in the pre-trial period
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])

degreesOfFreedom <- 7

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

#### 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 tim
e.
#### Conduct the analysis on trial store 88.
measureOverTime <- data[, .(totSales = sum(TOT_SALES),
                            nCust = uniqueN(LYLTY_CARD_NBR),
                            nTxnPerCust = (uniqueN(TXN_ID)) / (uniqueN(LYLTY_CARD_NBR)),
                            avgPricePerUnit = sum(TOT_SALES) / sum(PROD_QTY)),
                        by = c("STORE_NBR", "YEARMONTH")][order(STORE_NBR, YEARMONTH)]

#### Use the functions from earlier to calculate the correlation of the sales and number of customers of
each potential control store to the trial store
trial_store <- 88

corr_nSales <- calcCorr(preTrialMeasures, quote(totSales), trial_store)
corr_nCustomers <- calcCorr(preTrialMeasures, quote(nCust), trial_store)

#### Use the functions from earlier to calculate the magnitude distance of the sales and number of custo
mers of each potential control store to the trial store
magnitude_nSales <- calcMagnitudeDist(preTrialMeasures, quote(totSales), trial_store)
magnitude_nCustomers <- calcMagnitudeDist(preTrialMeasures, quote(nCust), trial_store)

#### Create a combined score composed of correlation and magnitude by merging the correlations table and
the magnitudes table, for each driver.
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1", "Store2"))[, scoreNSales := (corr_
measure + mag_measure) * 0.5]
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = c("Store1", "Store2"))[, scoreNCus
t := (corr_measure + mag_measure) * 0.5]


#### Combine scores across the drivers by merging sales scores and customer scores, and compute a final
combined score.
score_Control <- merge(score_nSales, score_nCustomers, by = c("Store1", "Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 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 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.

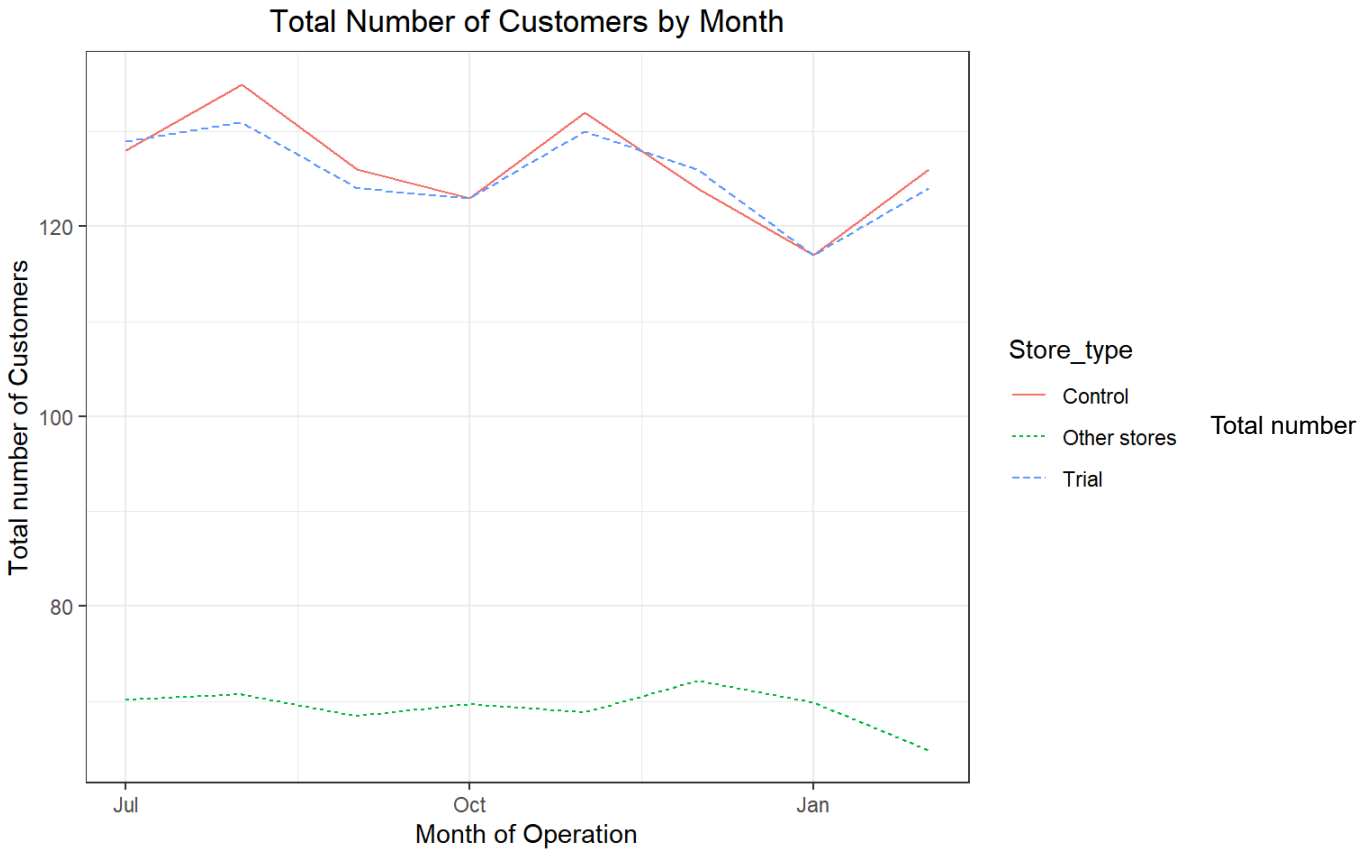
We'll look at total sales first.

```
#### 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, compared to the control store and other stores.
measureOverTimeSales <- measureOverTime[, YEARMONTH := as.numeric(YEARMONTH)]

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

ggplot(pastSales,
       aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line(aes(linetype = Store_type)) +
  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.

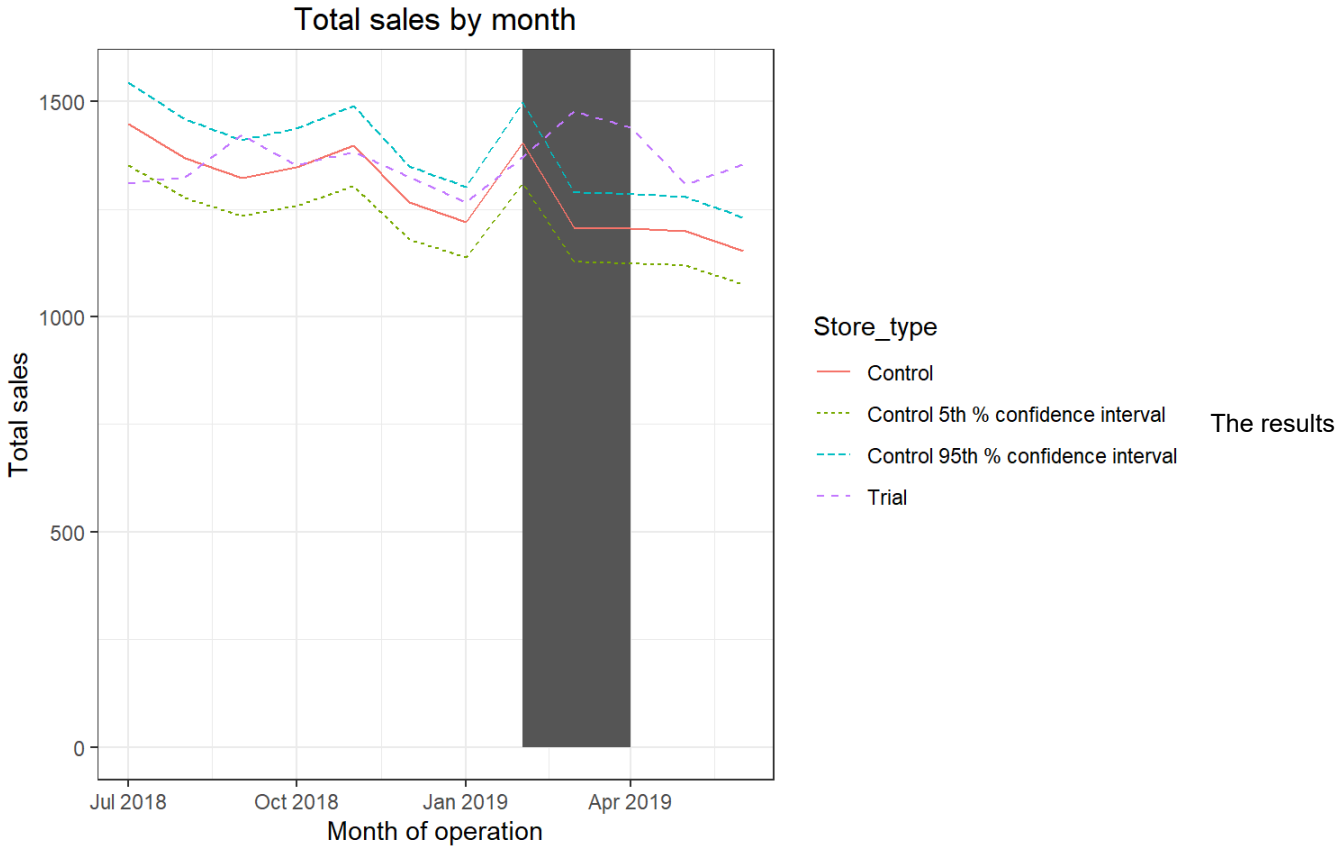
```
#### Visual checks on trends based on the drivers
#### For the period before the trial, create a graph with customer counts of the trial store for each mo
nth, compared to the control store and other stores.
measureOverTimeCusts <- measureOverTime

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



ggplot(pastCustomers, aes(TransactionMonth, numCust, colour = Store_type)) +
  geom_line(aes(linetype = Store_type)) +
  labs(x = "Month of Operation",
       y = "Total number of Customers",
       title = "Total Number of Customers by Month")
```



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(totS
ales)
                                                 ] / preTrialMeasures[STORE_NBR == control_store & YEARM
ONTH < 201902, sum(totSales)]

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

#### Calculate the absolute 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]

#### As our null hypothesis is that the trial period is the same as the pre-trial period, let's take the
standard deviation based on the scaled percentage difference in the pre-trial period
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])

degreesOfFreedom <- 7

#### Trial and control store total sales
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                         ifelse(STORE_NBR == control_store, "Control",
"Other stores"))
                              ][, totSales := mean(totSales), by = c("YEARMONTH", "Store_type")
                              ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100,
1, sep = "-"),
                                                              "%Y-%m-%d")
                              ][Store_type %in% c("Trial", "Control"), ]

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


#### Control store 5th percentile
pastSales_Controls5 <- pastSales[Store_type == "Control",
                                 ][, totSales := totSales * (1 - stdDev * 2)
                                 ][, Store_type := "Control 5th % confidence interval"]


#### 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(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")
```

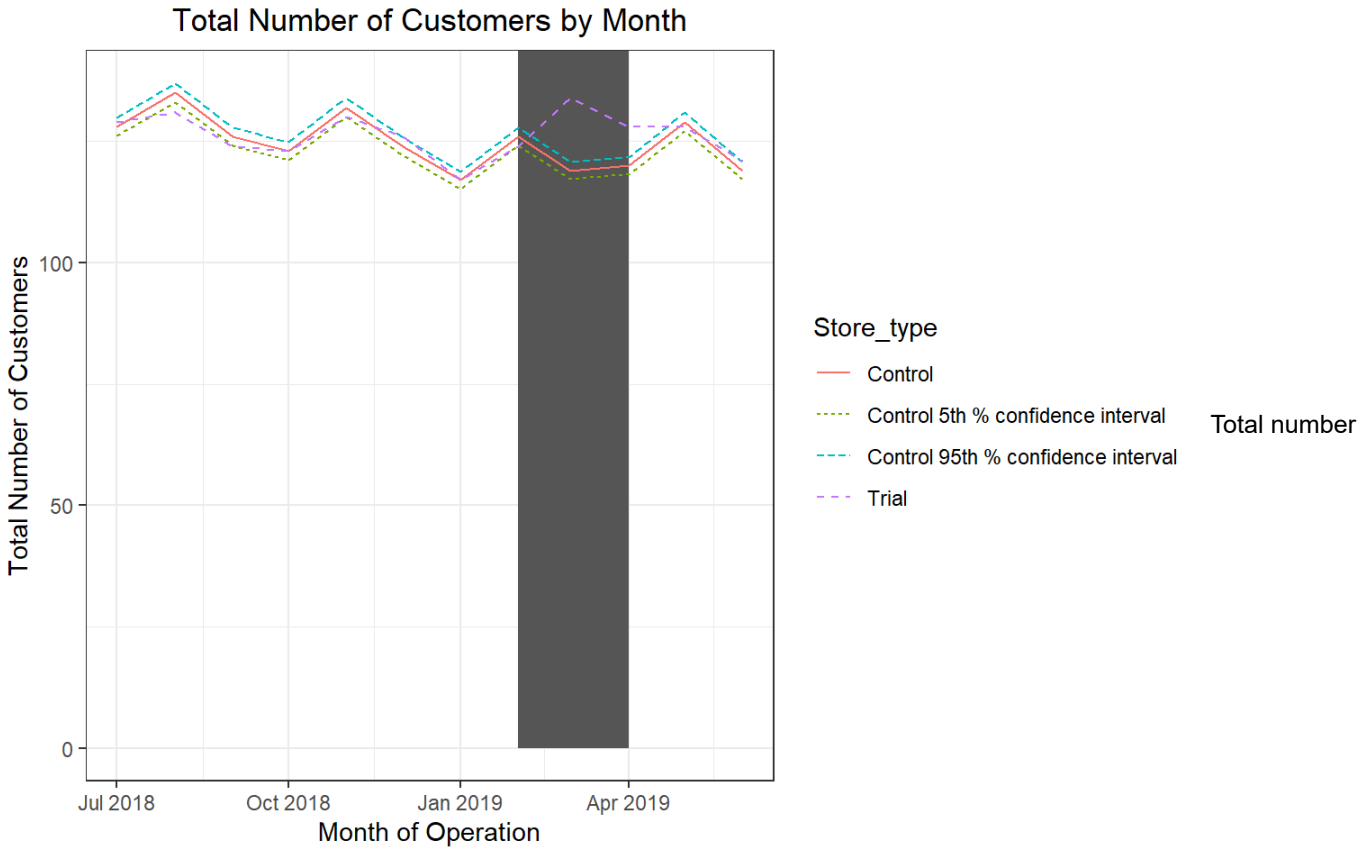


The results show that the trial in store 88 is significantly different to its control store in the trial period as the trial store performance lies outside of the 5% to 95% confidence interval of the control store in two of the three trial months.

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
t)
                                                ] / preTrialMeasures[STORE_NBR == control_store & YEARMO
NTH < 201902, sum(nCust)]

#### Apply the scaling factor
measureOverTimeCusts <- measureOverTime
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store,
                                               ][, controlCustomers := nCust * scalingFactorForControlCu
st
                                                 ][, Store_type := ifelse(STORE_NBR == trial_store, "Tri
al",
                                                                          ifelse(STORE_NBR == control_st
ore, "Control", "Other stores"))]

#### Calculate the absolute percentage difference between scaled control sales and trial sales
percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH", "controlCustomers")],
                        measureOverTime[STORE_NBR == trial_store,
                                        c("nCust", "YEARMONTH")
                                        ],
                        by = "YEARMONTH"
                        )[, percentageDiff := abs(controlCustomers - nCust) / controlCustomers]



#### As our null hypothesis is that the trial period is the same as the pre-trial period, let's take the
standard deviation based on the scaled percentage difference in the pre-trial period
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7 # 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[, nCusts := mean(nCust),
                                      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"]

#### 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, nCusts, colour = Store_type)) +
  geom_rect(data = trialAssessment[YEARMONTH > 201901 & YEARMONTH < 201905, ],
                       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 Number of Customers",
        title = "Total Number of Customers by Month")
```



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.