Quantium Virtual Internship - Retail Strategy and Analytics - Task 2

Upasana Sharma

10/23/2020

Solution for Task 2

Load required libraries and datasets

```
library(data.table)
library(ggplot2)
library(tidyr)
```

Load Datasets

```
filePath <- "C:/Quantium/"
data <- fread(paste0(filePath,"QVI_data.csv"))

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

Select control stores

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
#### Add a new month ID column in the data with the format yyyymm.
data[, YEARMONTH := year(DATE)*100 + month(DATE)]
data
```

LYLTY_CARD_… <int>	DATE <date>	STORE_… <int>	TXN… <int>	PROD_… <int>	PROD_NAME <chr>
1000	2018-10-17	1	1	5	Natural Chip Compny SeaSalt175g
1002	2018-09-16	1	2	58	Red Rock Deli Chikn&Garlic Aioli 150g
1003	2019-03-07	1	3	52	Grain Waves Sour Cream&Chives 210G

LYLTY_CARD_... <int>	DATE <date>	STORE_... <int>	TXN... <int>	PROD_... <int>	PROD_NAME <chr>
1003	2019-03-08	1	4	106	Natural ChipCo Hony Soy Chckn175g
1004	2018-11-02	1	5	96	WW Original Stacked Chips 160g
1005	2018-12-28	1	6	86	Cheetos Puffs 165g
1007	2018-12-04	1	7	49	Infuzions SourCream&Herbs Veg Strws 110g
1007	2018-12-05	1	8	10	RRD SR Slow Rst Pork Belly 150g
1009	2018-11-20	1	9	20	Doritos Cheese Supreme 330g
1010	2018-09-09	1	10	51	Doritos Mexicana 170g

1-10 of 10,000 rows | 1-6 of 13 columns Previous **1** 2 3 4 5 6 ... 1000 Next

```
#### 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.
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("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.

```
#### 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())

  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_NBR == i,
                                                                                              ev
al(metricCol)]))
    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.

```
#### Create a function to calculate a standardized magnitude distance for a measure, looping throu
gh each control store
calculateMagnitudeDistance <- function(inputTable, metricCol, storeComparison) {
calcDistTable = data.table(Store1 = numeric(), Store2 = numeric(), YEARMONTH =
numeric(), measure = numeric())
  storeNumbers <- unique(inputTable[, STORE_NBR])
  for (i in storeNumbers) {
  calculatedMeasure = data.table("Store1" = storeComparison
                                 , "Store2" = i
                                 , "YEARMONTH" = inputTable[STORE_NBR ==
storeComparison, YEARMONTH]
                                 , "measure" = abs(inputTable[STORE_NBR ==
storeComparison, eval(metricCol)]
                                                   - inputTable[STORE_NBR == i,
eval(metricCol)])
                                  )
    calcDistTable <- rbind(calcDistTable, calculatedMeasure)
  }

#### Standardise the magnitude distance so that the measure ranges from 0 to 1
  minMaxDist <- calcDistTable[, .(minDist = min(measure), maxDist = max(measure)),
by = c("Store1", "YEARMONTH")]
  distTable <- merge(calcDistTable, minMaxDist, by = c("Store1", "YEARMONTH"))
  distTable[, magnitudeMeasure := 1 - (measure - minDist)/(maxDist - minDist)]
  finalDistTable <- distTable[, .(mag_measure = mean(magnitudeMeasure)), by =
.(Store1, Store2)]
  return(finalDistTable)
}
```

Now let's use the functions to find the control stores! We'll select control stores based on how similar monthly total sales in dollar amounts and monthly number of customers are to the trial stores. So we will need to use our functions to get four scores, two for each of total sales and total customers.

```
#### Use the function you created to calculate correlations against store 77 using total sales and
number of customers.
trial_store <- 77
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales), trial_store)
corr_nSales[order(-corr_measure)]
```

Store1 <dbl>	Store2 <dbl>	corr_measure <dbl>
77	77	1.000000000
77	71	0.914105965
77	233	0.903774188
77	119	0.867664404
77	17	0.842668360

Store1 <dbl>	Store2 <dbl>	corr_measure <dbl>
77	3	0.806643637
77	41	0.783231868
77	50	0.763865842
77	157	0.735893224
77	162	0.729740138

1-10 of 260 rows Previous **1** 2 3 4 5 6 ... 26 Next

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

Store1 <dbl>	Store2 <dbl>	corr_measure <dbl>
77	77	1.000000000
77	233	0.990357788
77	119	0.983266608
77	254	0.916208390
77	113	0.901347988
77	84	0.858571239
77	41	0.844219490
77	3	0.834207432
77	35	0.774647081
77	88	0.765047968

1-10 of 260 rows Previous **1** 2 3 4 5 6 ... 26 Next

```
#### 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.

```
#### Create a combined score composed of correlation and magnitude, by first merging the correlati
ons table with the magnitude table.
#### A simple average on the scores: 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)/2 ]
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by =
        c("Store1", "Store2"))[, scoreNCust := (corr_measure + mag_measure)/2]
```

```
score_nSales[order(-scoreNSales)]
```

Store1 <dbl>	Store2 <dbl>	corr_measure <dbl>	mag_measure <dbl>	scoreNSales <dbl>
77	77	1.000000000	1.00000000	1.000000000
77	233	0.903774188	0.98526489	0.944519541
77	41	0.783231868	0.96514010	0.874185985
77	50	0.763865842	0.97312929	0.868497568
77	17	0.842668360	0.88068824	0.861678301
77	115	0.689158820	0.93283212	0.810995469
77	167	0.657110366	0.95913323	0.808121796
77	265	0.639759375	0.96266286	0.801211116
77	234	0.696324778	0.89033921	0.793331995
77	84	0.684347845	0.83008517	0.757216508

1-10 of 260 rows Previous **1** 2 3 4 5 6 ... 26 Next

```
score_nCustomers[order(-scoreNCust)]
```

Store1 <dbl>	Store2 <dbl>	corr_measure <dbl>	mag_measure <dbl>	scoreNCust <dbl>
77	77	1.000000000	1.00000000	1.000000e+00
77	233	0.990357788	0.99277331	9.915655e-01
77	254	0.916208390	0.93713119	9.266698e-01
77	41	0.844219490	0.97463924	9.094294e-01
77	84	0.858571239	0.92418181	8.913765e-01
77	17	0.747307760	0.96249530	8.549015e-01
77	115	0.718881754	0.96591604	8.423989e-01

Store1 <dbl>	Store2 <dbl>	corr_measure <dbl>	mag_measure <dbl>	scoreNCust <dbl>
77	35	0.774647081	0.90692675	8.407869e-01
77	167	0.717912623	0.94934912	8.336309e-01
77	111	0.685925661	0.96606414	8.259949e-01

1-10 of 260 rows Previous 1 2 3 4 5 6 ... 26 Next

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 single 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)]
```

Stor... <dbl>	Stor... <dbl>	corr_measure.x <dbl>	mag_measur... <dbl>	scoreNSales <dbl>	corr_measure.y <dbl>	mag_measur... <dbl>
77	77	1.000000000	1.00000000	1.000000000	1.000000000	1.00000000
77	233	0.903774188	0.98526489	0.944519541	0.990357788	0.99277331
77	41	0.783231868	0.96514010	0.874185985	0.844219490	0.97463924
77	17	0.842668360	0.88068824	0.861678301	0.747307760	0.96249530
77	254	0.577108489	0.92277135	0.749939920	0.916208390	0.93713119
77	115	0.689158820	0.93283212	0.810995469	0.718881754	0.96591604
77	84	0.684347845	0.83008517	0.757216508	0.858571239	0.92418181
77	167	0.657110366	0.95913323	0.808121796	0.717912623	0.94934912
77	50	0.763865842	0.97312929	0.868497568	0.607390794	0.92507621
77	111	0.519472674	0.96546566	0.742469166	0.685925661	0.96606414

1-10 of 260 rows | 1-8 of 9 columns Previous 1 2 3 4 5 6 ... 26 Next

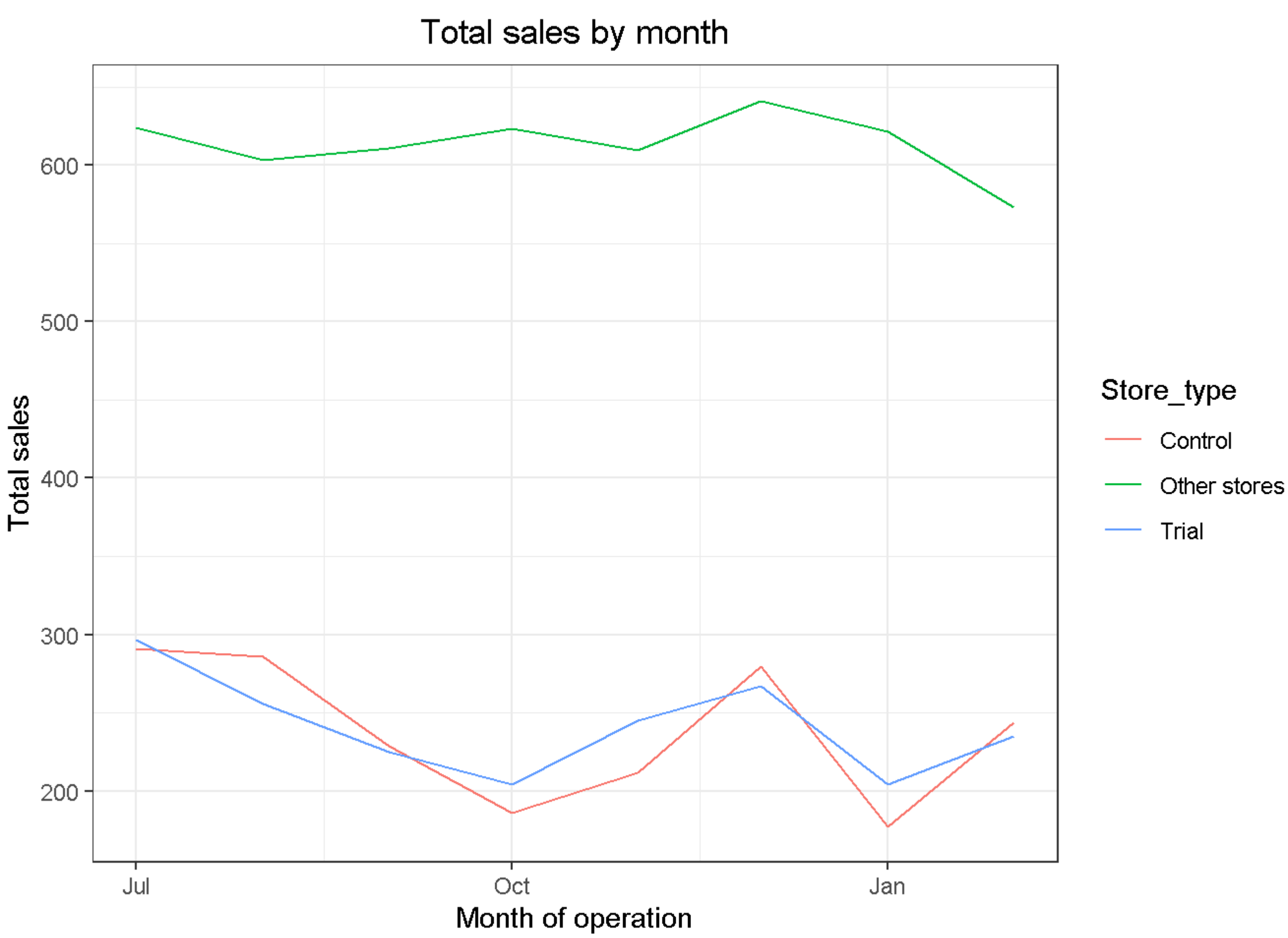
The store with the highest score is then selected as the control store since it is most similar to the trial store.

```
#### 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 hi
ghest 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")
                              ][, 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.

```
#### Conduct visual checks on customer count trends by comparing the trial store
#### 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"
))
][, 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 = "Total number of customers", title = "Total number of custome
rs 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(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.

```
#### 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)/controlSa
les]
```

```
percentageDiff # between control store sales and trial store sales
```

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 ta
ke 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 sto
res.
#### Calculate the t-values for the trial months. After that, find the 95th percentile of the t di
stribution with the appropriate degrees of freedom
#### to check whether the hypothesis is statistically significant.
#### 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)]
```

TransactionMonth <date>	tValue <dbl>
2019-02-01	1.183534
2019-03-01	7.339116
2019-04-01	12.476373

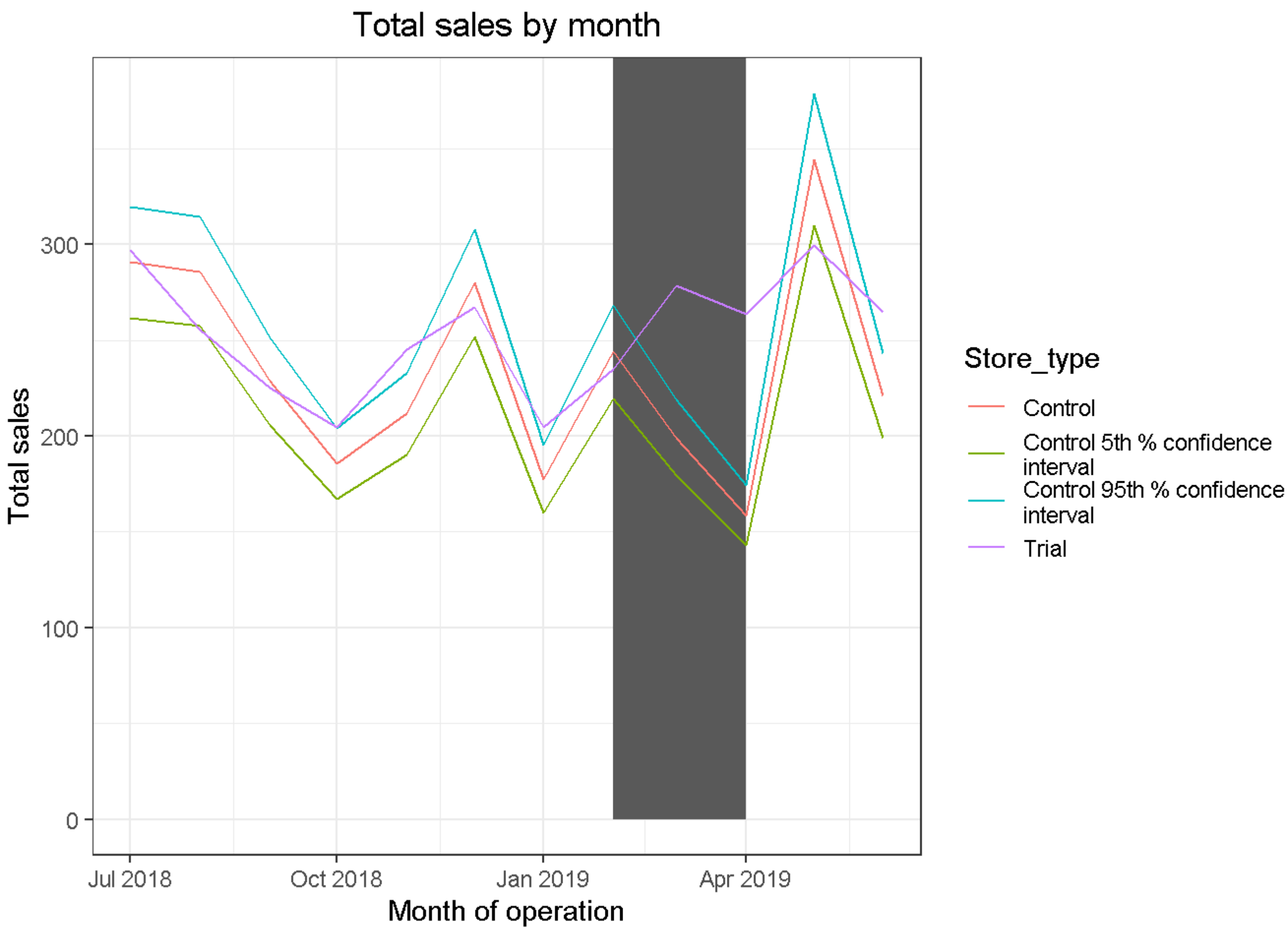
3 rows

```
#### Find the 95th percentile of the t distribution with the appropriate degrees of freedom to com
pare 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 graph
ggplot(trialAssessment, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0 , ymax =
Inf, color = NULL), show.legend = FALSE) +
  geom_line() +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by month")
```

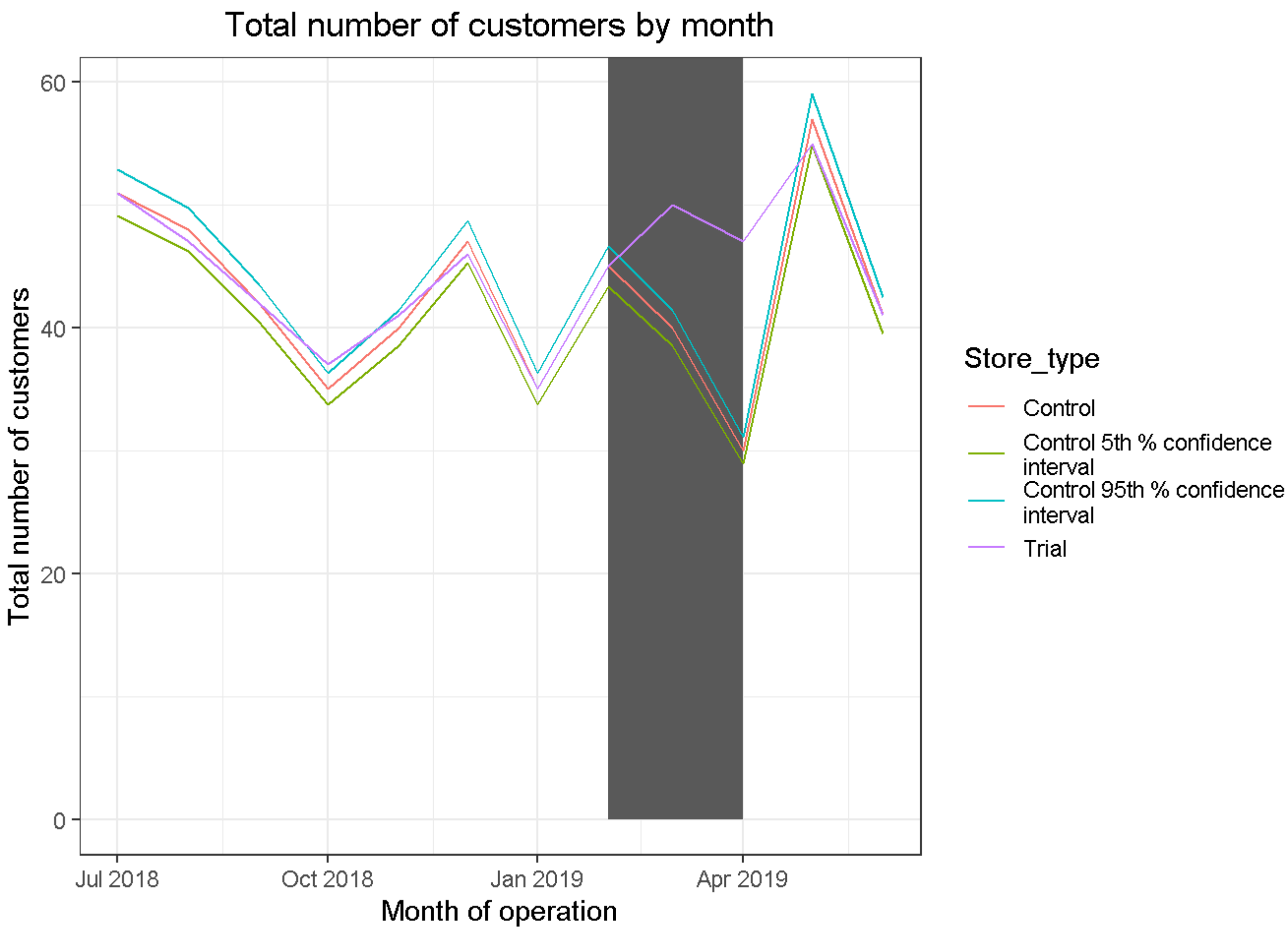


The results show that the trial in store 77 is significantly different to its control store in the trial period as the trial store performance lies outside the 5% to 95% confidence interval of the control store in two of the three trial months. Let's have a look at assessing this for number of customers 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 customers.
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(nCustomers)] / preTrialMeasures[STORE_NBR ==
control_store & YEARMONTH < 201902, sum(nCustomers)]
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"))]
percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH", "controlCustomers")],
measureOverTimeCusts[STORE_NBR == trial_store,c("nCustomers", "YEARMONTH")],
by = "YEARMONTH"
)[, percentageDiff := abs(controlCustomers-nCustomers)/controlCustomers]
```

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 ta
ke 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(nCustomers), by =
c("YEARMONTH", "Store_type")
                            ][Store_type %in% c("Trial", "Control"), ]

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

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

trialAssessment <- rbind(pastCustomers, pastCustomers_Controls95,
pastCustomers_Controls5)

#### Plot everything into one grap, (geom_rect creates a rectangle in the plot. Use this to highli
ght the trial period in our graph).
ggplot(trialAssessment, aes(TransactionMonth, nCusts, color = Store_type)) +
  geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0 ,
ymax = Inf, color = NULL), show.legend = FALSE) +
  geom_line() + labs(x = "Month of operation", y = "Total number of customers", title = "Total num
ber 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.

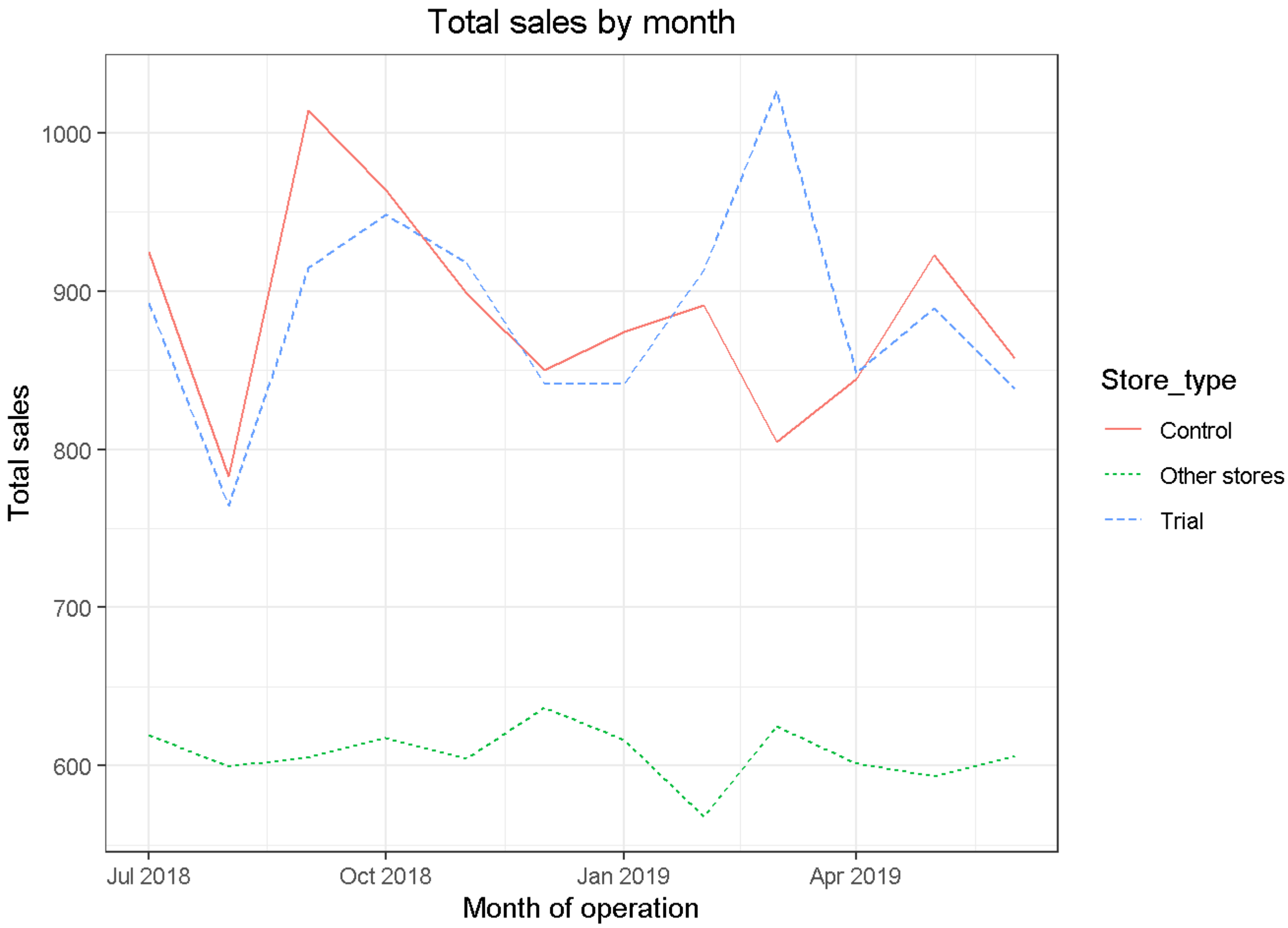
Trial store 86

```
#### 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("STORE_NBR",
"YEARMONTH")][order(STORE_NBR, YEARMONTH)]

#### Use the functions we created earlier to calculate correlations and magnitude for each potenti
al control store
trial_store <- 86
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_stor
e)

#### 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_measure + mag_measure)/2]
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = c("Store1", "Store2"))[ , sc
oreNCust := (corr_measure + mag_measure)/2]

#### 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 + scoreNCust * 0.5]

#### Select control stores based on the highest matching store (closest to 1 but not the store its
elf, 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
```

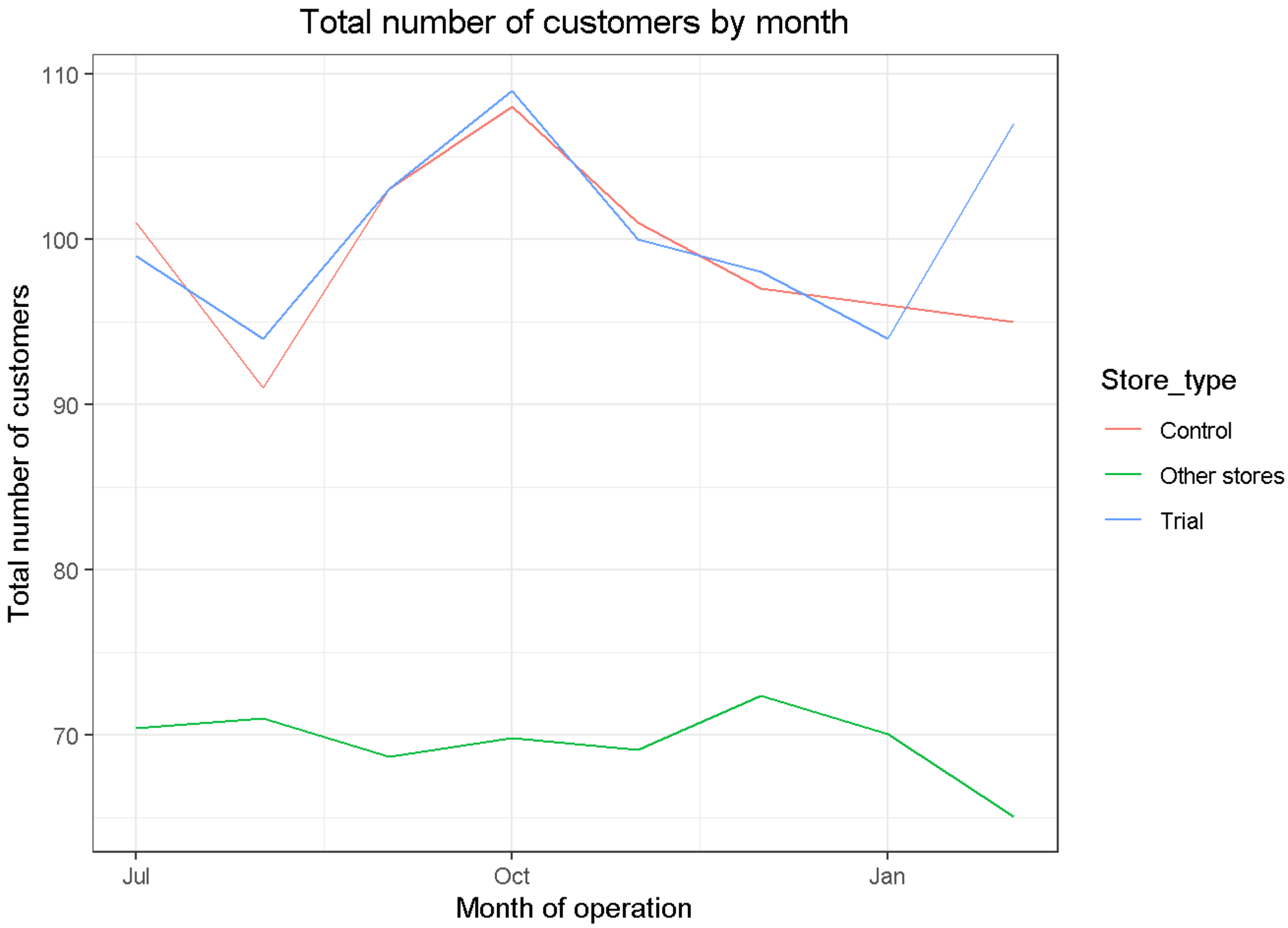
Looks like store 155 will be a control store for trial store 86. Again, let’s check visually if the drivers are indeed similar in the period before the trial. We’ll look at total sales first.

```
#### Conduct visual checks on trends based on the drivers
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type:= ifelse(STORE_NBR == trial_store, "Trial", ifelse
(STORE_NBR== control_store, "Control", "Other stores"))][, totSales := mean(totSales), by = c("YEA
RMONTH", "Store_type")][, TransactionMonth:= as.Date(paste(YEARMONTH%/%100, YEARMONTH%% 100, 1, se
p = "-"), "%Y-%m-%d")][YEARMONTH <210903]

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



Sales are trending in a similar way. Next, number of customers.

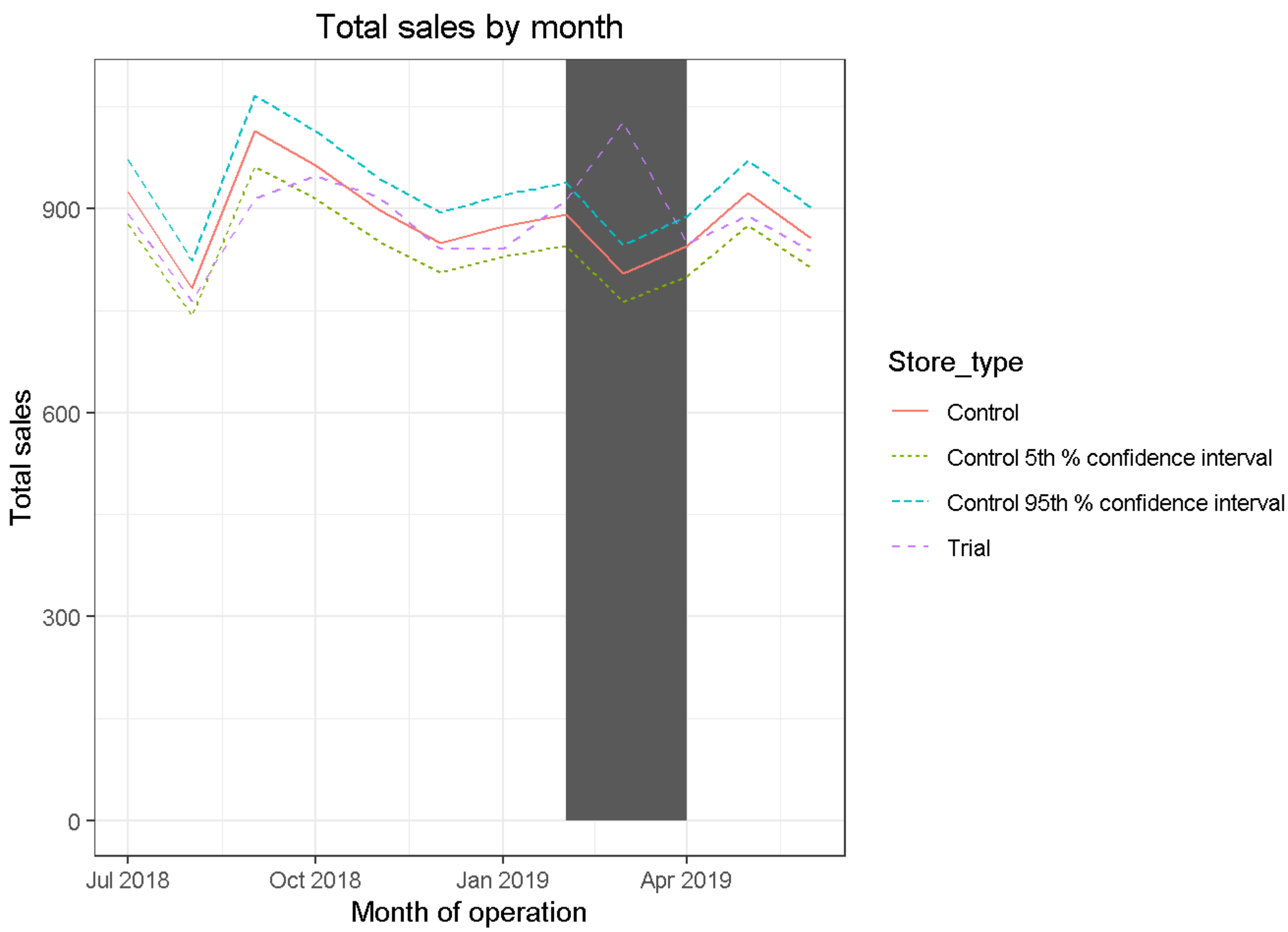
```
#### Conduct visual checks on trends based on the drivers
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
ifelse(STORE_NBR == control_store, "Control", "Other stores"))
][, numberCustomers := mean(nCustomers), 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 = "Total number of customers", title = "Total number of custome
rs by month")
```



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]

#### Calculate the percentage difference between scaled control sales and trial sales When calcula
ting percentage difference, remember to use absolute difference
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 ta
ke the standard deviation based on the scaled percentage difference
#### in the pre-trial period
#### Calculate the standard deviation of percentage differences during the pre-trial period
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7

#### Trial and control store total sales

#### Create a table with sales by store type and month. We only need data for the trial and contro
l 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. The 5th and 95th percentiles c
an be approximated by using two standard deviations away from the mean.
#### Recall that the variable stdDev earlier calculates standard deviation in percentages, and not
dollar sales.
#### Control store 95th percentile
pastSales_Controls95 <- pastSales[Store_type == "Control",
][, totSales := totSales * (1 + stdDev * 2)
][, Store_type := "Control 95th % confidence interval"]

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

#### Then, create a combined table with columns from pastSales, pastSales_Controls95 and pastSales
_Controls5
```

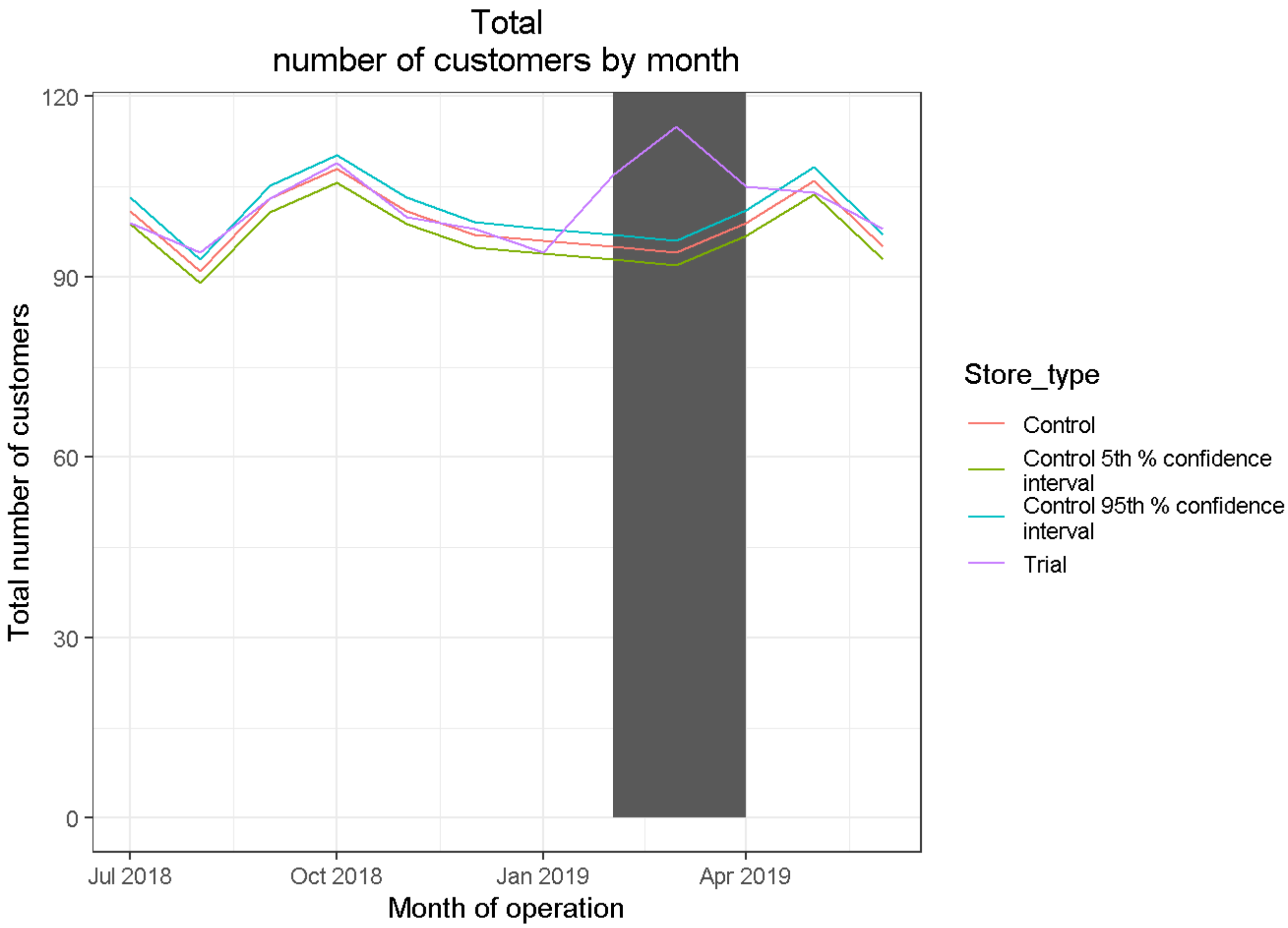
```
trialAssessment <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)

#### Plotting these in one 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(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 ta
ke 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(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 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.

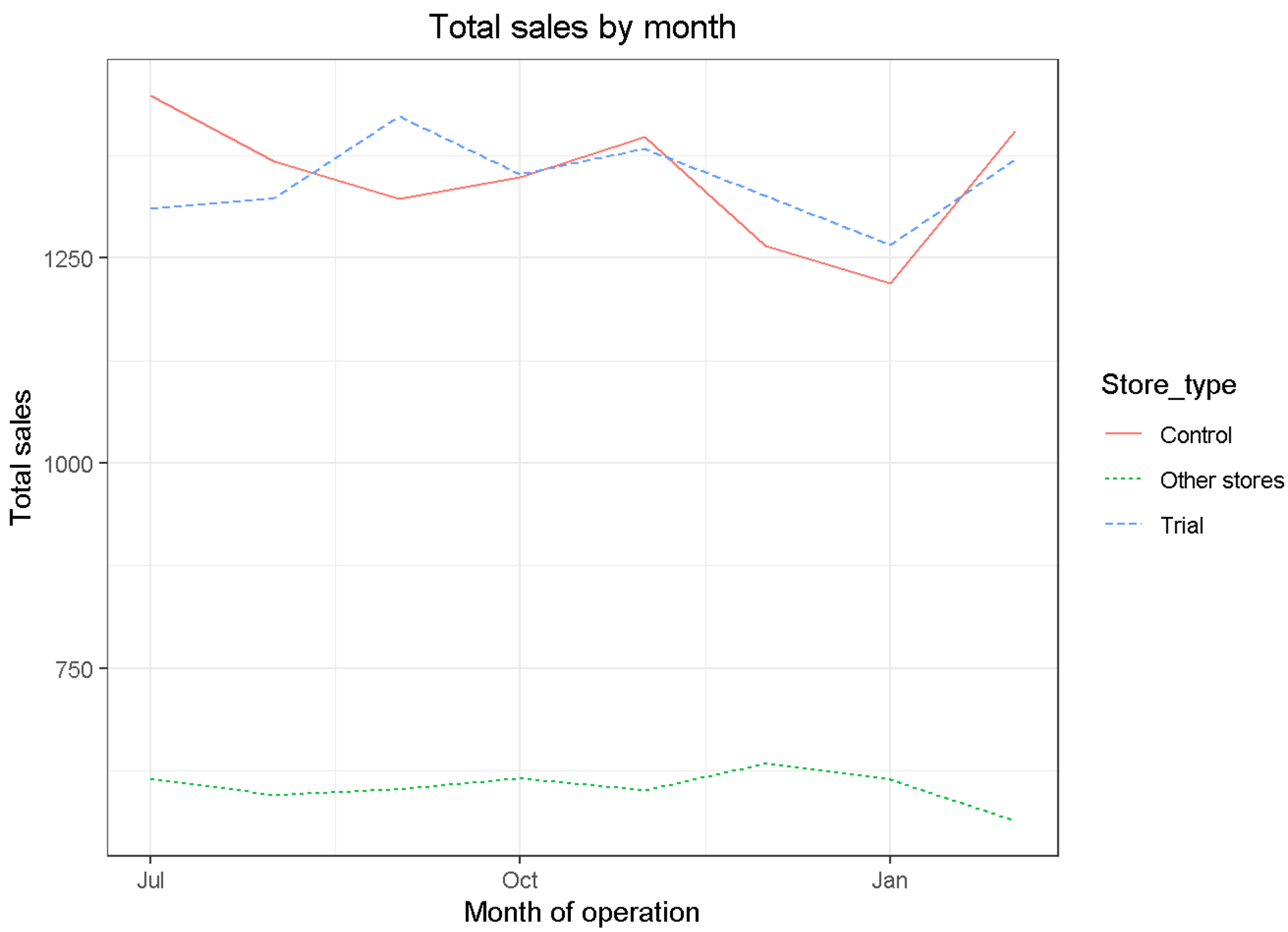
Trial store 88

```
#### 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),
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 from earlier to calculate the correlation of the sales and number of custom
ers of each potential control store to the trial store
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
customers of each potential control store to the trial store
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales), trial_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures, quote(nCustomers), trial_stor
e)

#### Create a combined score composed of correlation and magnitude by merging the correlations tab
le 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)/2]
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = c("Store1", "Store2"))[ , sc
oreNCust := (corr_measure + mag_measure)/2]

#### 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 it
self, 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
```

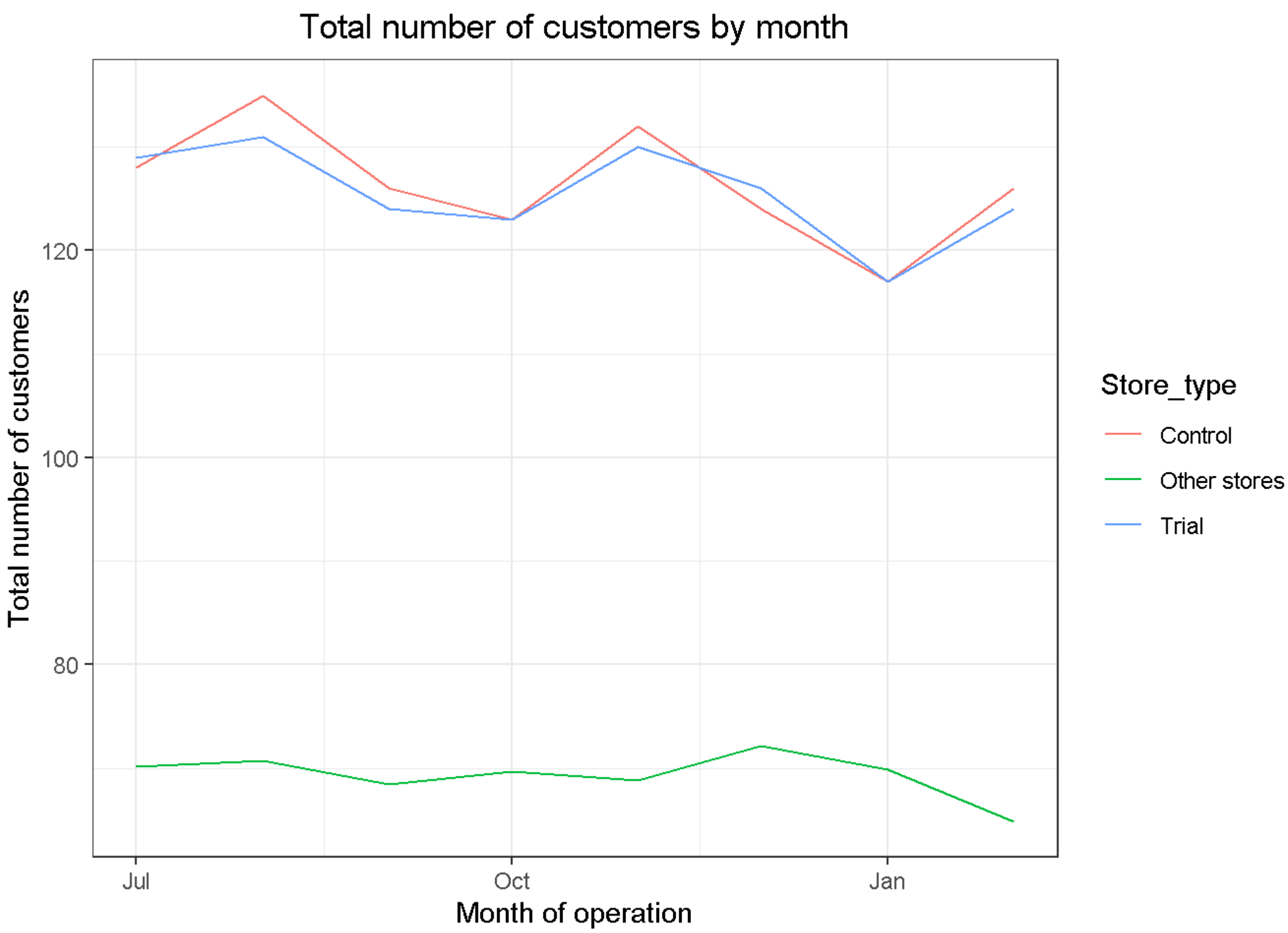
We've now found store 237 to be a suitable control store for trial store 88. Again, let's check visually if the drivers are indeed similar in the period before the trial. We'll look at total sales first.

```
#### 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
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")
```



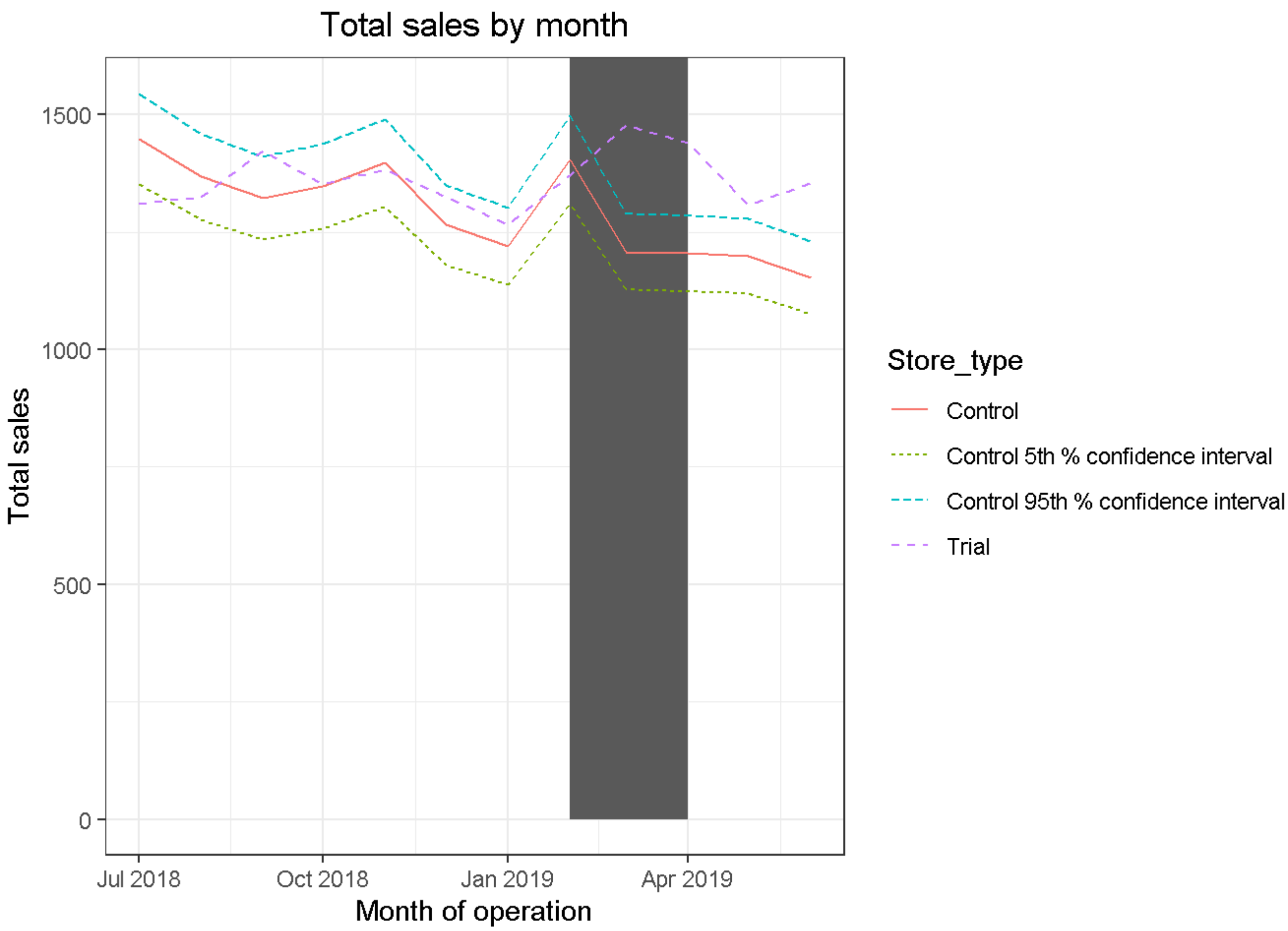
The trial and control stores have similar total sales. Next, number of customers.

```
#### 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 e
ach month, 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"))
][, 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 = "Total number of customers", title = "Total num
ber of customers by month")
```



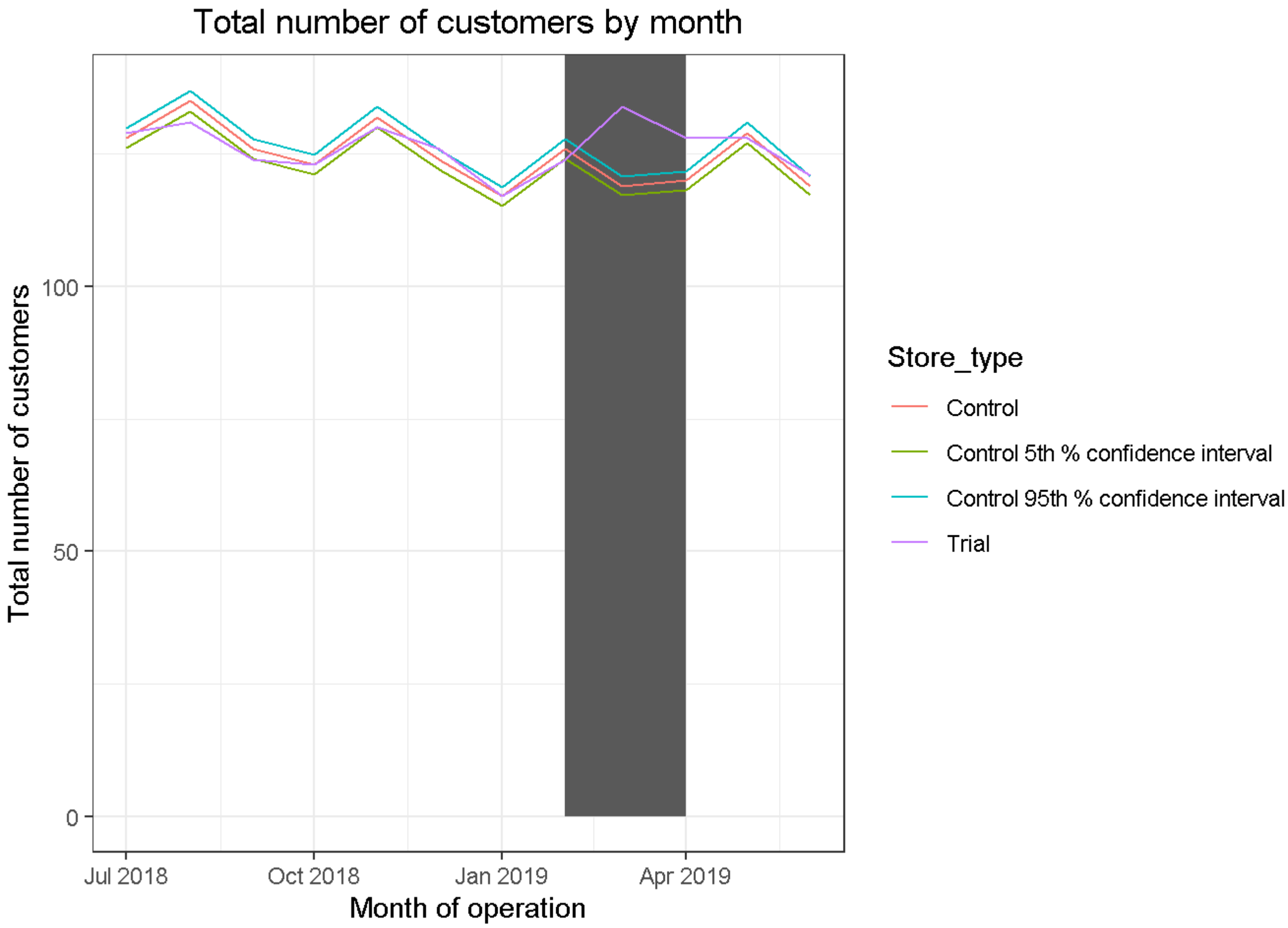
Total number of customers of the control and trial stores are also similar. Let’s now assess the impact of the trial on sales.

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

#### Apply the scaling factor
measureOverTimeSales <- measureOverTime
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][ ,controlSales := totSale
s * 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(control
Sales-totSales)/controlSales]

#### As our null hypothesis is that the trial period is the same as the pre-trial period,  let's t
ake 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"]
trialAssessment <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)

#### Plotting these in one 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(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 absolute percentage difference between scaled control sales and trial sales
percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH","controlCustomers")],measureOverTim
e[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 ta
ke 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 customer-s
pastCustomers <- measureOverTimeCusts[, nCusts := mean(nCustomers), by = c("YEARMONTH", "Store_typ
e")
][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 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")
```



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

I have 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.