Task_2

Shrinath Rajeshirke

26/02/2022

```
## Load required libraries
library(data.table)
library(ggplot2)
library(tidyr)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Load the dataset
filepath = "C:/Users/ASUS/Desktop/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))
```

The client has selected store numbers 77, 86 & 88 as trial stores and want control stores to be established stores that are operational for the entire 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:

- monthly overall sales revenue
- Monthly number of customers
- Monthly number of transactions per customer

```
#### Create a month ID
data[, YEARMONTH := year(DATE)*100 + month(DATE)]

#### Define the measure calculations
```

```
measure <- 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(measure[, .N, STORE_NBR][N == 12,
                                                     STORE_NBR])
preTrialMeasures <- measure[YEARMONTH < 201902 & STORE_NBR %in%
                              storesWithFullObs, ]
```

Now, we have to find out how similar each potential control stores is to the trial store for this calculate the correlation between performance of each store to the trial store.

```
#### Create a function to calculate correlation for a measure,
#### looping through each control 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,

eval(metricCol)]))
calcCorrTable <- rbind(calcCorrTable, calculatedMeasure)
}
return(calcCorrTable)
}
```

We can also calculate a standardised metric based on the absolute difference betweeen the trial's score performance and each control store's performance.

```
#### Create a function to calculate a standardised magnitude distance for
ameasure,
#### looping through each control store
calculateMagnitudeDistance <- function(inputTable, metricCol,
storeComparison){
  calcDistTable = data.table(Store1 = numeric(), Store2 = numeric(),
```

```
                                YEARMONTH = numeric(), measure = numeric())
  storeNumbers <- unique(inputTable[, STORE_NBR])
  for (i in storeNumbers) {
    calculatedMeasure = data.table("Store1" = storeComparison,
                                   "Store2" = i,
                                   "YEARMONTH" = inputTable[STORE_NBR ==
storeComparison, YEARMONTH],
                                   "measure" = abs(inputTable[STORE_NBR ==
storeComparison,

eval(metricCol)] - inputTable[STORE_NBR == i,

eval(metricCol)]))
calcDistTable <- rbind(calcDistTable, calculatedMeasure)
  }

#### 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)
}
```

We 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 get 4 scores, 2 for each of total sales and total customers.

```
#### Use the functions for calculating correlation
trial_store <- 77
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales),
                                    trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers),
                                        trial_store)
#### 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
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by =
c("Store1","Store2"))[, scoreNSales := corr_measure * corr_weight +
    mag_measure * (1- corr_weight)]
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by =
c("Store1", "Store2"))[, scoreNCust := corr_measure * corr_weight +
mag_measure * (1- corr_weight)]
```

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
score_Control <- merge(score_nSales, score_nCustomers, by =
c("Store1","Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]
```

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

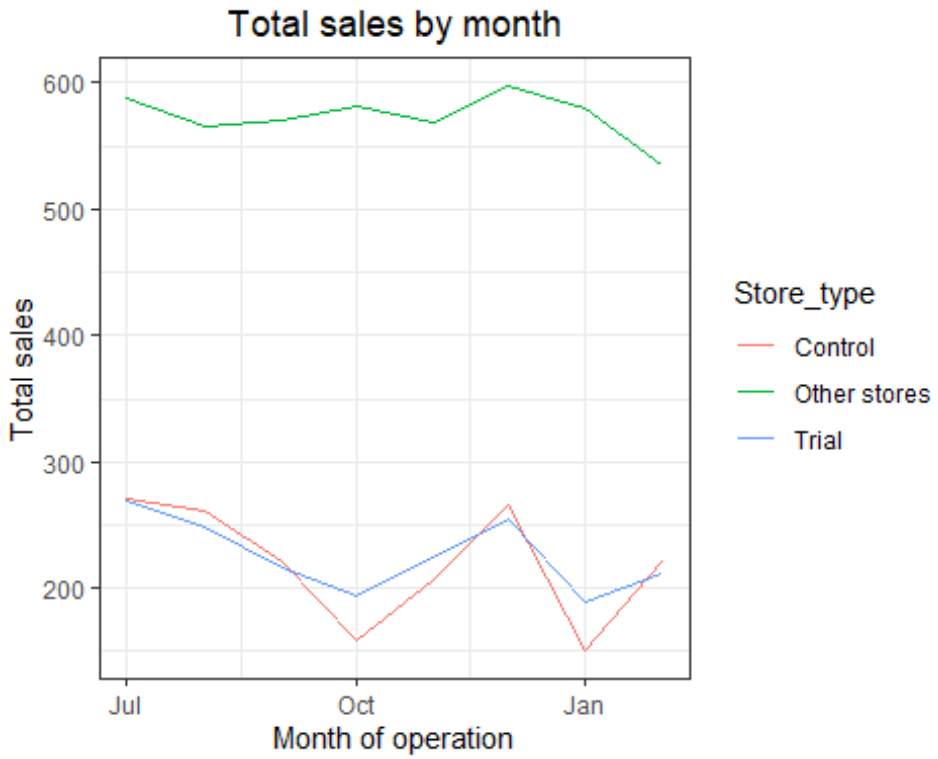
```
#### Select control stores based on the highest matching store (closest to 1
but
#### not the store itself, i.e. the second ranked highest store)
#### Select control store for trial store 77
control_store <- score_Control[Store1 == trial_store, ][order(-
finalControlScore)][2, Store2]
control_store
```

```
## [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.

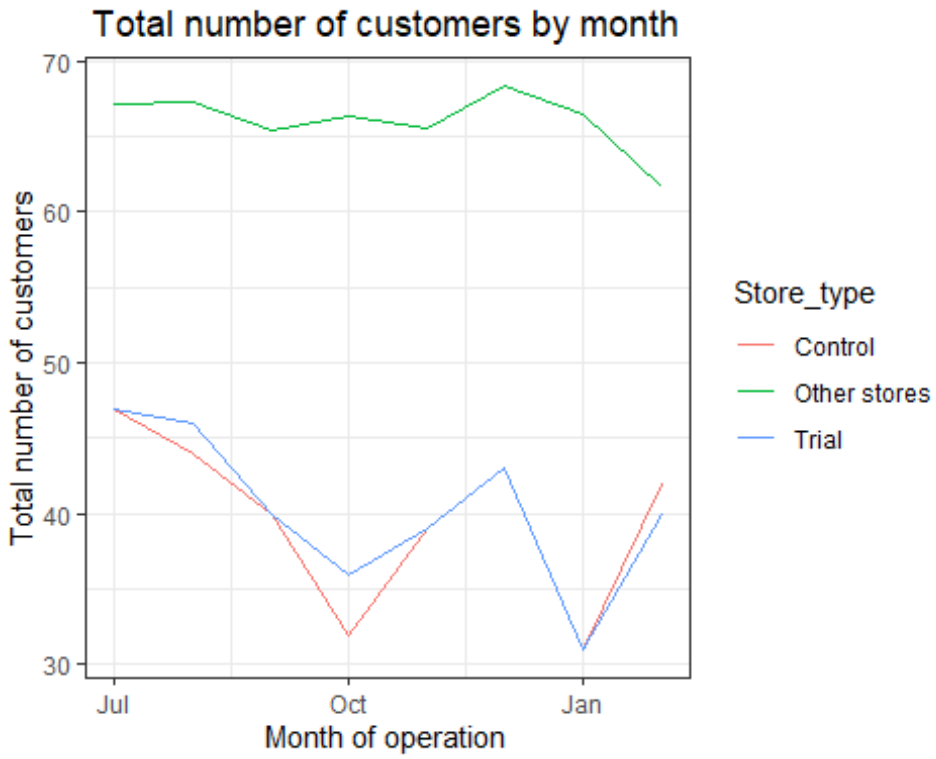
```
#### Visual checks on trends based on the drivers
measureOverTimeSales <- measure
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR ==
                                                           trial_store,
"Trial",
ifelse(STORE_NBR == control_store,
       "Control", "Other stores"))
][, 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.

```
#### Visual checks on trends based on the drivers
measureOverTimeCusts <- measure
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 customers by month")
```



Assessment of trial

The trial period goes from the start of March 2019 to June 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 <- measure
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")],
                        measure[STORE_NBR == trial_store,
c("totSales","YEARMONTH")],
                        by = "YEARMONTH")[, percentageDiff :=
```

```
                                                    abs(controlSales-
totSales)/controlSales]
```

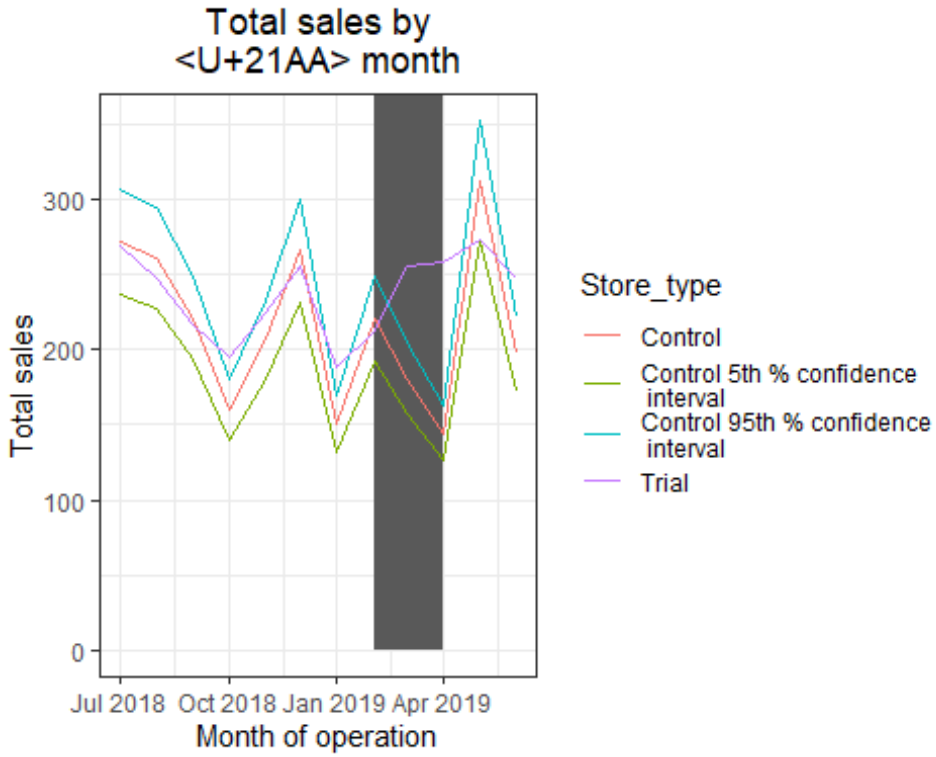
Let's see if the difference is significant

```
#### 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
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    tValue
## 1:       2019-02-01  1.223912
## 2:       2019-03-01  5.633494
## 3:       2019-04-01 11.336505

### 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 <- measure
#### Trial and control store total sales
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")
```



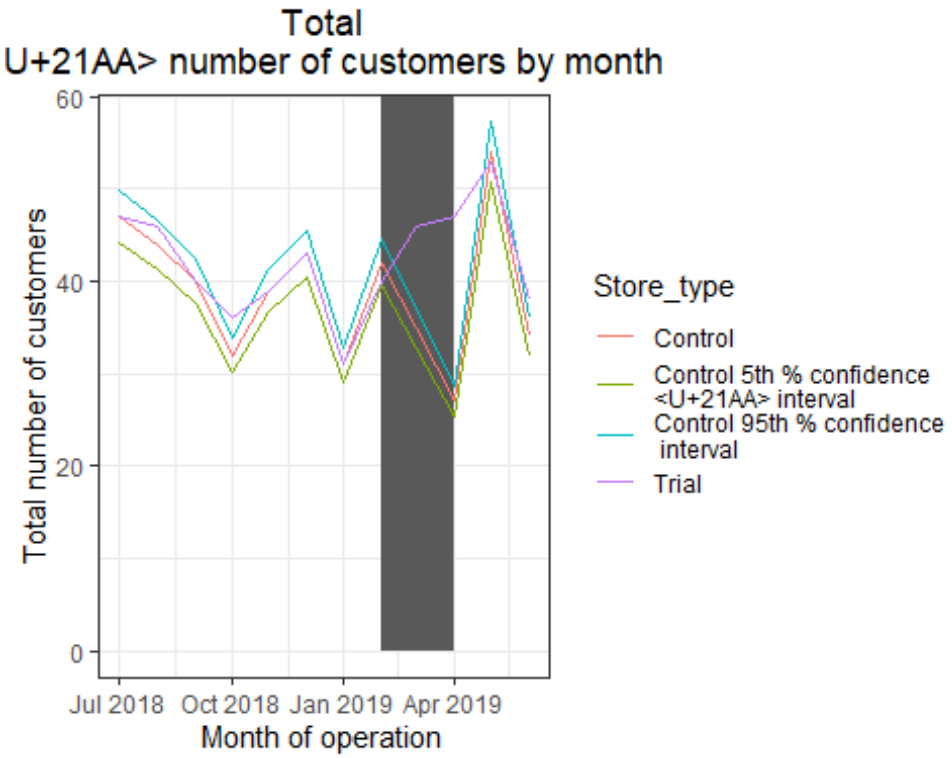
The results show that the trial in store 77 is significantly different to its control store in the trial period as the trial store performance lies outside the 5% to 95% confidence interval of the control store in two of the three trial months. Let's have a look at assessing this for number of customers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

#### Apply the scaling factor
measureOverTimeCusts <- measure
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")],
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 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(nCustomers), by =
 c("YEARMONTH", "Store_type")
][Store_type %in% c("Trial", "Control"), ]
#### Control store 95th percentile
pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",
][, nCusts := nCusts * (1 + stdDev * 2)
][, Store_type := "Control 95th % confidence
 interval"]
#### Control store 5th percentile
```

```
pastCustomers_Controls5 <- pastCustomers[Store_type == "Control",
][, nCusts := nCusts * (1 - stdDev * 2)
][, Store_type := "Control 5th % confidence
 interval"]
trialAssessment <- rbind(pastCustomers, pastCustomers_Controls95,
 pastCustomers_Controls5)
#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, nCusts, color = Store_type)) +
geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0 ,
ymax = Inf, color = NULL), show.legend = FALSE) +

geom_line() +
labs(x = "Month of operation", y = "Total number of customers", title =
"Total
 number of customers by month")
```



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

```
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 for calculating correlation
trial_store <- 86
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales),
 trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers),
 trial_store)
#### Use the functions for calculating magnitude
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures,
quote(totSales), trial_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures,
 quote(nCustomers), trial_store)
#### Create a combined score composed of correlation and magnitude
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1",
"Store2"))[, scoreNSales := corr_measure * corr_weight + mag_measure * (1-
corr_weight)]

score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by =
c("Store1", "Store2"))[, scoreNCust := corr_measure * corr_weight +
mag_measure * (1- corr_weight)]

#### Combine scores across the drivers
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 86
control_store <- score_Control[Store1 == trial_store,
][order(-finalControlScore)][2, Store2]
control_store

## [1] 155
```

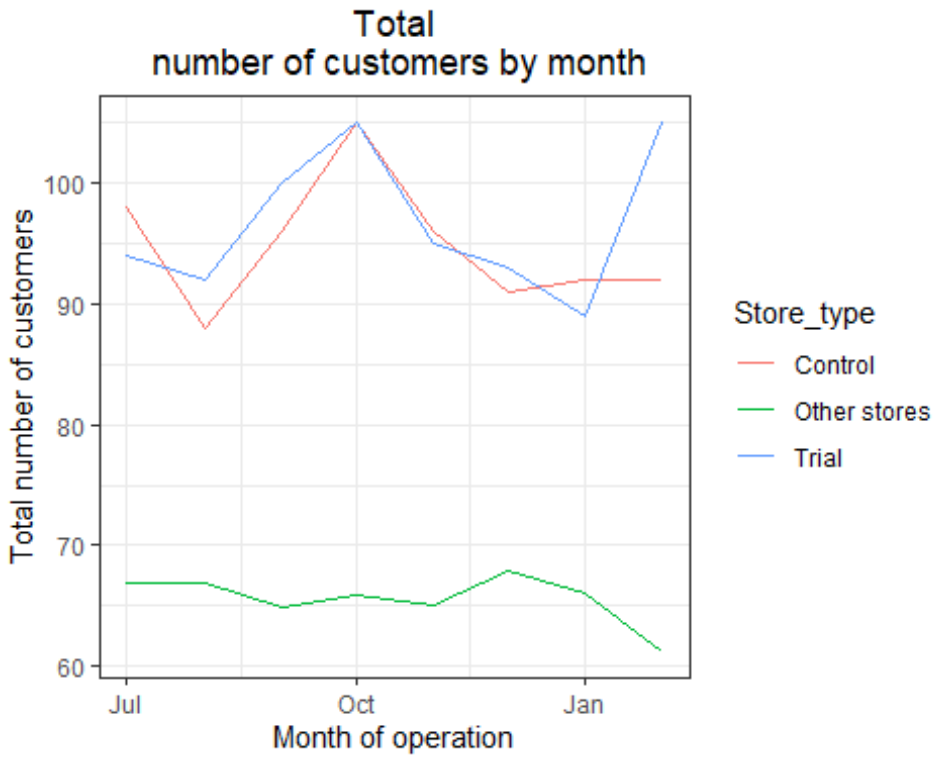
Looks like store 155 will be a control store for trial store 86. Again, let's check visually if the drivers are indeed similar in the period before the trial. We'll look at total sales first.

```
#### 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(aes(linetype = Store_type)) +
labs(x = "Month of operation", y = "Total sales", title = "Total sales by
 month")
```

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

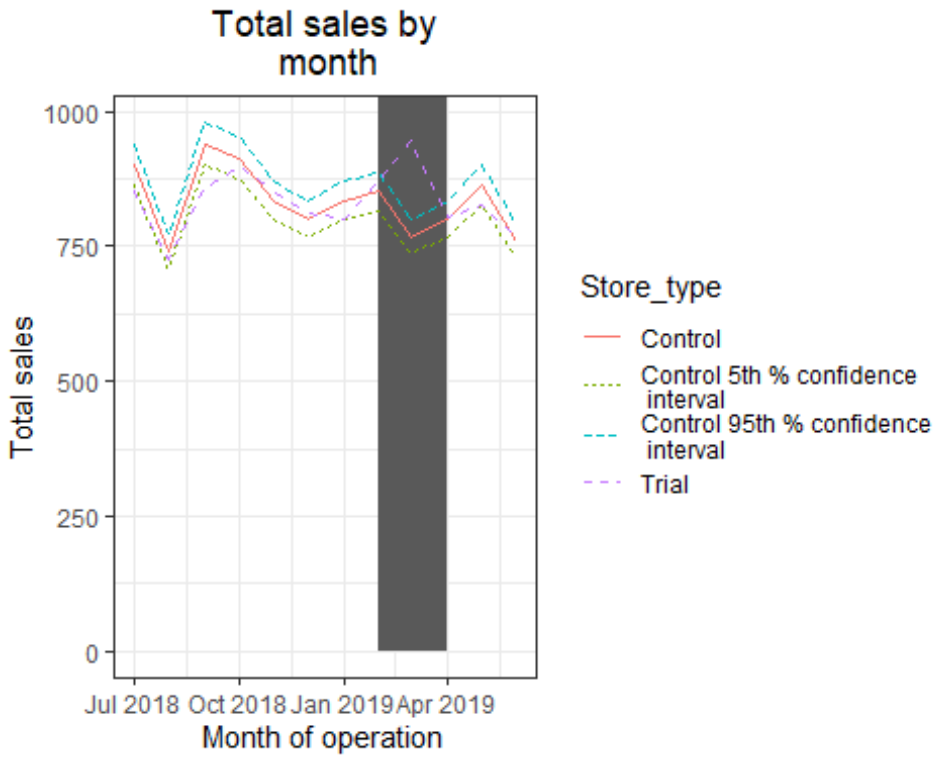
```
#### Visual checks on trends based on the drivers
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR ==
 trial_store, "Trial",
ifelse(STORE_NBR == control_store,
 "Control", "Other stores"))
][, numberCustomers := mean(nCustomers), by =
 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 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(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
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"]
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")
```

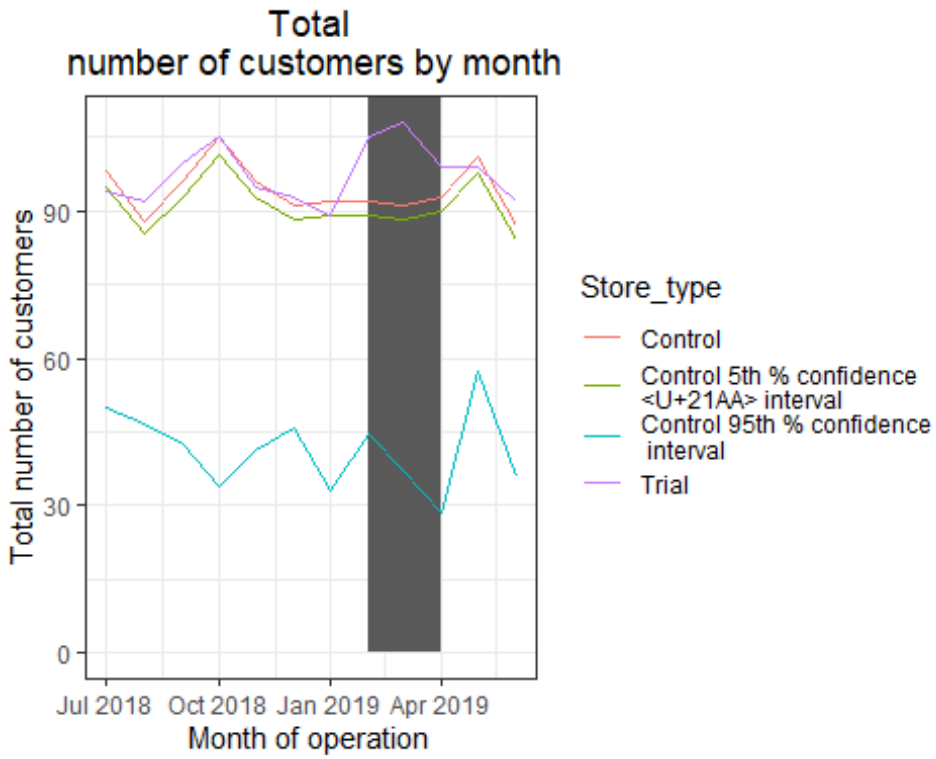


The results show that the trial in store 86 is not significantly different to its control store in the trial period as the trial store performance lies inside the 5% to 95% confidence interval of the control store in two of the three trial months. Let's have a look at assessing this for number of customers 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 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(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"]

##       STORE_NBR YEARMONTH totSales nCustomers nTxnPerCust nChipsPerTxn
##  [1,]     FALSE     FALSE     TRUE       TRUE        TRUE         TRUE
##  [2,]     FALSE     FALSE     TRUE       TRUE        TRUE         TRUE
##  [3,]     FALSE     FALSE     TRUE       TRUE        TRUE         TRUE
##  [4,]     FALSE     FALSE     TRUE       TRUE        TRUE         TRUE
##  [5,]     FALSE     FALSE     TRUE       TRUE        TRUE         TRUE
##  [6,]     FALSE     FALSE     TRUE       TRUE        TRUE         TRUE
##  [7,]     FALSE     FALSE     TRUE       TRUE        TRUE         TRUE
##  [8,]     FALSE     FALSE     TRUE       TRUE        TRUE         TRUE
##  [9,]     FALSE     FALSE     TRUE       TRUE        TRUE         TRUE
## [10,]     FALSE     FALSE     TRUE       TRUE        TRUE         TRUE
## [11,]     FALSE     FALSE     TRUE       TRUE        TRUE         TRUE
## [12,]     FALSE     FALSE     TRUE       TRUE        TRUE         TRUE
##       avgPricePerUnit Store_type TransactionMonth numberCustomers nCusts
##  [1,]            TRUE      FALSE            FALSE            TRUE   TRUE
##  [2,]           FALSE      FALSE            FALSE            TRUE   TRUE
##  [3,]            TRUE      FALSE            FALSE            TRUE   TRUE
##  [4,]            TRUE      FALSE            FALSE            TRUE   TRUE
##  [5,]           FALSE      FALSE            FALSE            TRUE   TRUE
##  [6,]           FALSE      FALSE            FALSE            TRUE   TRUE
##  [7,]           FALSE      FALSE            FALSE            TRUE   TRUE
##  [8,]            TRUE      FALSE            FALSE            TRUE   TRUE
##  [9,]           FALSE      FALSE            FALSE            TRUE   TRUE
## [10,]           FALSE      FALSE            FALSE            TRUE   TRUE
## [11,]           FALSE      FALSE            FALSE            TRUE   TRUE
## [12,]           FALSE      FALSE            FALSE            TRUE   TRUE

#### 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

```
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 for calculating correlation
trial_store <- 88
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales),
 trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers),
 trial_store)
#### Use the functions for calculating magnitude
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures,
 quote(totSales), trial_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures,
 quote(nCustomers), trial_store)
#### Create a combined score composed of correlation and magnitude
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1",
"Store2"))[, scoreNSales := corr_measure * corr_weight + mag_measure * (1-
corr_weight)]

score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by =
c("Store1", "Store2"))[, scoreNCust := corr_measure * corr_weight +
mag_measure * (1- corr_weight)]

#### Combine scores across the drivers
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.

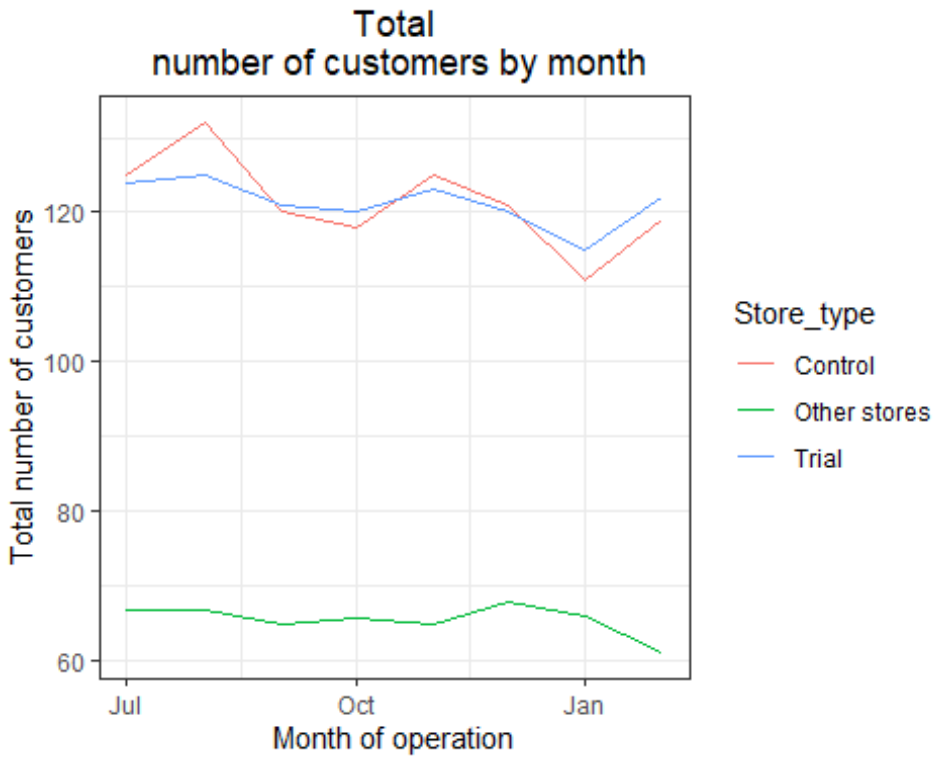
```
#### Visual checks on trends based on the drivers
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR ==
 trial_store, "Trial",
ifelse(STORE_NBR == control_store,
 "Control", "Other stores"))
][, totSales := mean(totSales), by = c("YEARMONTH",
```

```
 "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")
```

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

```
#### 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 =
```

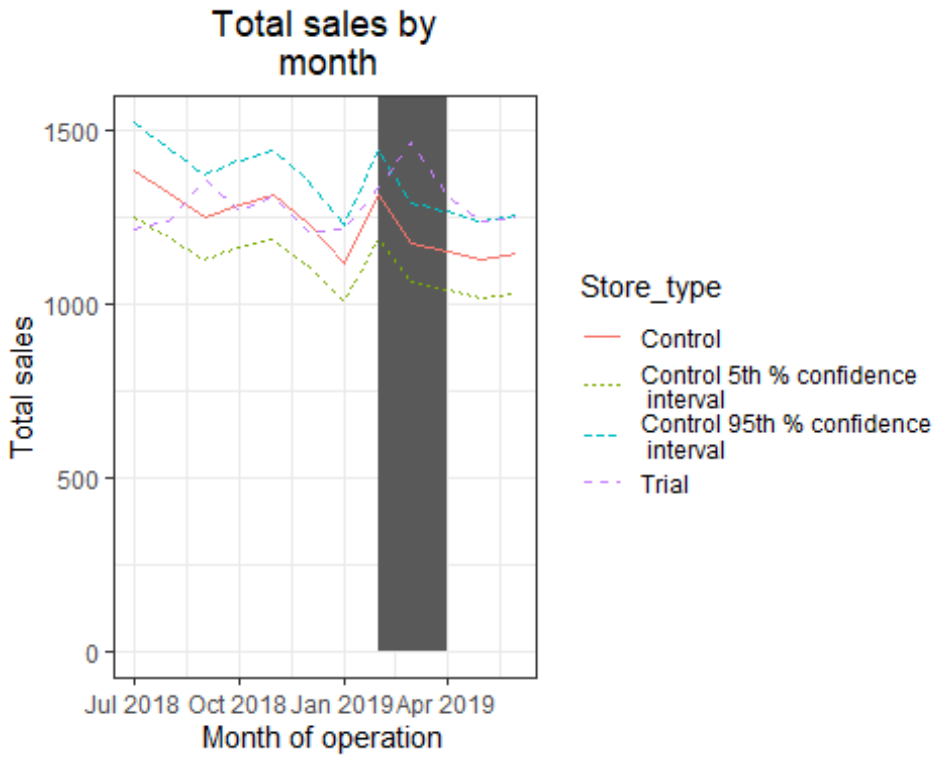
```
"Total
 number of customers by month")
```



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 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
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
```

```r
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 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")
```



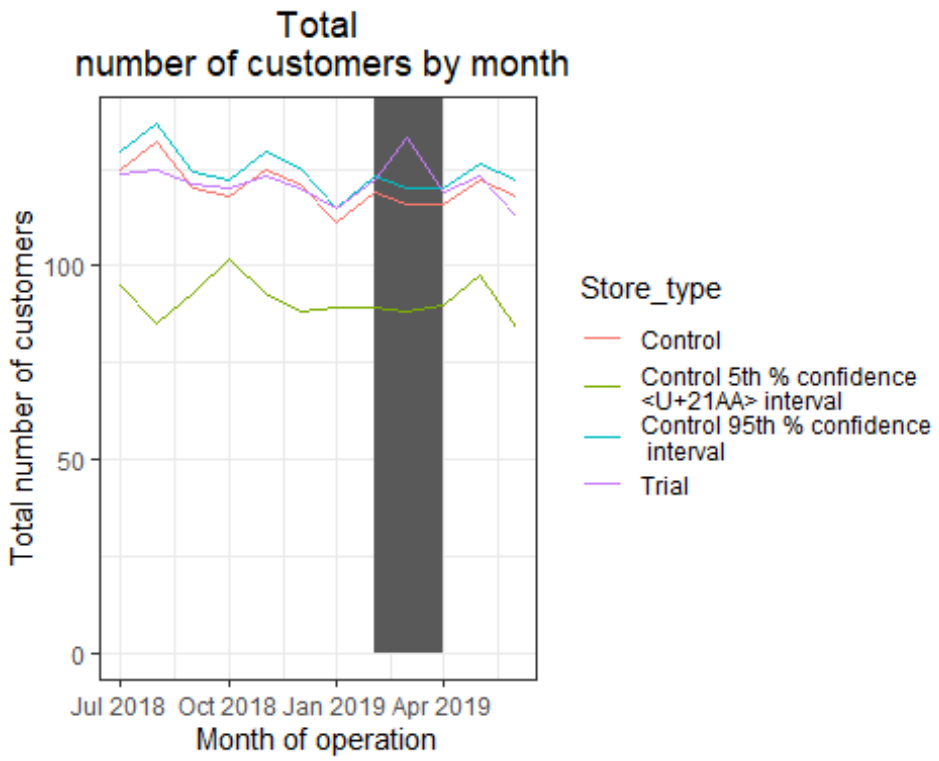
The results show that the trial in store 88 is significantly different to its control store in the trial period as the trial store performance lies outside of the 5% to 95% confidence interval of the control store in two of the three trial months. Let's have a look at assessing this for number of customers as well.

```
#### This would be a repeat of the steps before for total sales
#### Scale pre-trial control customers to match pre-trial trial store
customers
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(nCustomers)]/preTrialMeasures[STORE_NBR ==
control_store & YEARMONTH < 201902, sum(nCustomers)]

#### Apply the scaling factor
measureOverTimeCusts <- measureOverTime
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store,
][ , controlCustomers :=
 nCustomers * scalingFactorForControlCust
][, Store_type := ifelse(STORE_NBR
 == trial_store, "Trial",
ifelse(STORE_NBR == control_store,
 "Control", "Other stores"))
]
#### Calculate the percentage difference between scaled control sales and
trial sales
percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH",
"controlCustomers")],
```

```
measureOverTime[STORE_NBR == trial_store,
c("nCustomers", "YEARMONTH")],
by = "YEARMONTH"
)[, percentageDiff :=
 abs(controlCustomers-nCustomers)/controlCustomers]
#### As our null hypothesis is that the trial period is the same as the
pre-trial period, let's take 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(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"]

##       STORE_NBR YEARMONTH totSales nCustomers nTxnPerCust nChipsPerTxn
##  [1,]      TRUE     FALSE     TRUE       TRUE         TRUE        FALSE
##  [2,]      TRUE     FALSE     TRUE       TRUE        FALSE        FALSE
##  [3,]      TRUE     FALSE     TRUE       TRUE        FALSE        FALSE
##  [4,]      TRUE     FALSE     TRUE       TRUE        FALSE         TRUE
##  [5,]      TRUE     FALSE     TRUE       TRUE        FALSE        FALSE
##  [6,]      TRUE     FALSE     TRUE       TRUE        FALSE        FALSE
##  [7,]      TRUE     FALSE     TRUE       TRUE        FALSE        FALSE
##  [8,]      TRUE     FALSE     TRUE       TRUE        FALSE        FALSE
##  [9,]      TRUE     FALSE     TRUE       TRUE        FALSE         TRUE
## [10,]      TRUE     FALSE     TRUE       TRUE        FALSE        FALSE
## [11,]      TRUE     FALSE     TRUE       TRUE        FALSE        FALSE
## [12,]      TRUE     FALSE     TRUE       TRUE        FALSE        FALSE
##       avgPricePerUnit Store_type TransactionMonth numberCustomers nCusts
##  [1,]            TRUE      FALSE            FALSE            TRUE   TRUE
##  [2,]            TRUE      FALSE            FALSE            TRUE   TRUE
##  [3,]            TRUE      FALSE            FALSE            TRUE   TRUE
##  [4,]            TRUE      FALSE            FALSE            TRUE   TRUE
##  [5,]            TRUE      FALSE            FALSE            TRUE   TRUE
##  [6,]            TRUE      FALSE            FALSE            TRUE   TRUE
##  [7,]            TRUE      FALSE            FALSE            TRUE   TRUE
##  [8,]            TRUE      FALSE            FALSE            TRUE   TRUE
##  [9,]            TRUE      FALSE            FALSE            TRUE   TRUE
```

```
## [10,]            TRUE      FALSE            FALSE            TRUE    TRUE
## [11,]            TRUE      FALSE            FALSE            TRUE    TRUE
## [12,]            TRUE      FALSE            FALSE            TRUE    TRUE
```

```
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")
```



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

1) The control stores 233, 155, 237 for trial stores 77, 86 and 88 respectively.
2) 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.