Quantium Virtual Internship - Retail Strategy and Analytics - Task 2

Solution for Task 2

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

```
#read csv with data.table
data <- fread("~/Quantium Virtual Internship/QVI_data.csv")

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

Assign the data files to data.tables

Select control stores

The client has selected store numbers 77, 86 and 88 as trial stores and wants 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

data[, YEARMONTH :=  strftime(DATE, "%Y%m")]
# For each store and month calculate total sales, number of customers, transactions per customer, chips

measureOverTime <- data[, list(totSales = sum(TOT_SALES),
                              nCustomers = uniqueN(LYLTY_CARD_NBR),
                              nTxn = .N,
                              nTxnPerCust=.N/uniqueN(LYLTY_CARD_NBR),
                              nChipsPerTxn = sum(PROD_QTY)/.N,
                              avgPricePerUnit = mean(price_per_qty)),
                              by = .(YEARMONTH, STORE_NBR)][order(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.

```
#### Let's define inputTable as a metric table with potential comparison stores, metricCol as the store

calculateCorrelation <- function(inputTable, metricCol, storeComparison) {

  #define empty corrtable
  calcCorrTable = data.table(Store1 = numeric(),
                            Store2 = numeric(),
                           corr_measure = numeric())

 storeNumbers <- storesWithFullObs

 for (i in storeNumbers) {

calculatedMeasure = data.table("Store1" = storeComparison,"Store2" = i,"corr_measure" = cor(inputTable[S


calcCorrTable <- rbind(calcCorrTable,calculatedMeasure)
 }
 return(calcCorrTable)
}
```

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

```
#### Create a function to calculate a standardized magnitude distance for a measure,
#### looping through each control store
calculateMagnitudeDistance <- function(inputTable, metricCol, storeComparison) {
calcDistTable = data.table(Store1 = numeric(), Store2 = numeric(), YEARMONTH =
numeric(), measure = numeric())
 storeNumbers <- unique(inputTable[, STORE_NBR])

 for (i in storeNumbers) {
 calculatedMeasure = data.table("Store1" = storeComparison
 , "Store2" = i
 , "YEARMONTH" = inputTable[STORE_NBR ==
storeComparison, YEARMONTH]
 , "measure" = abs(inputTable[STORE_NBR ==
storeComparison, eval(metricCol)]
 - inputTable[STORE_NBR == i,
eval(metricCol)])
 )
 calcDistTable <- rbind(calcDistTable, calculatedMeasure)
 }

#### 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, na.rm = TRUE)), 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.

```
trial_store <- 77

corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales), trial_store)

corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(totSales), trial_store)

#### 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 correlations tabl


corr_weight <- 0.5

score_nSales <- merge(corr_nSales, magnitude_nSales , by = "Store2")[, scoreNSales := 0.5 * corr_measur

score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = "Store2")[, scoreNCust := 0.5 * c
```

Now we have a score for each of total number of sales and number of customers. Let's combine the two via a simple average.

```
#### Combine scores across the drivers by first merging our sales scores and customer scores into a sin

score_Control <- merge(score_nSales, score_nCustomers, by = "Store2")

score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]
```

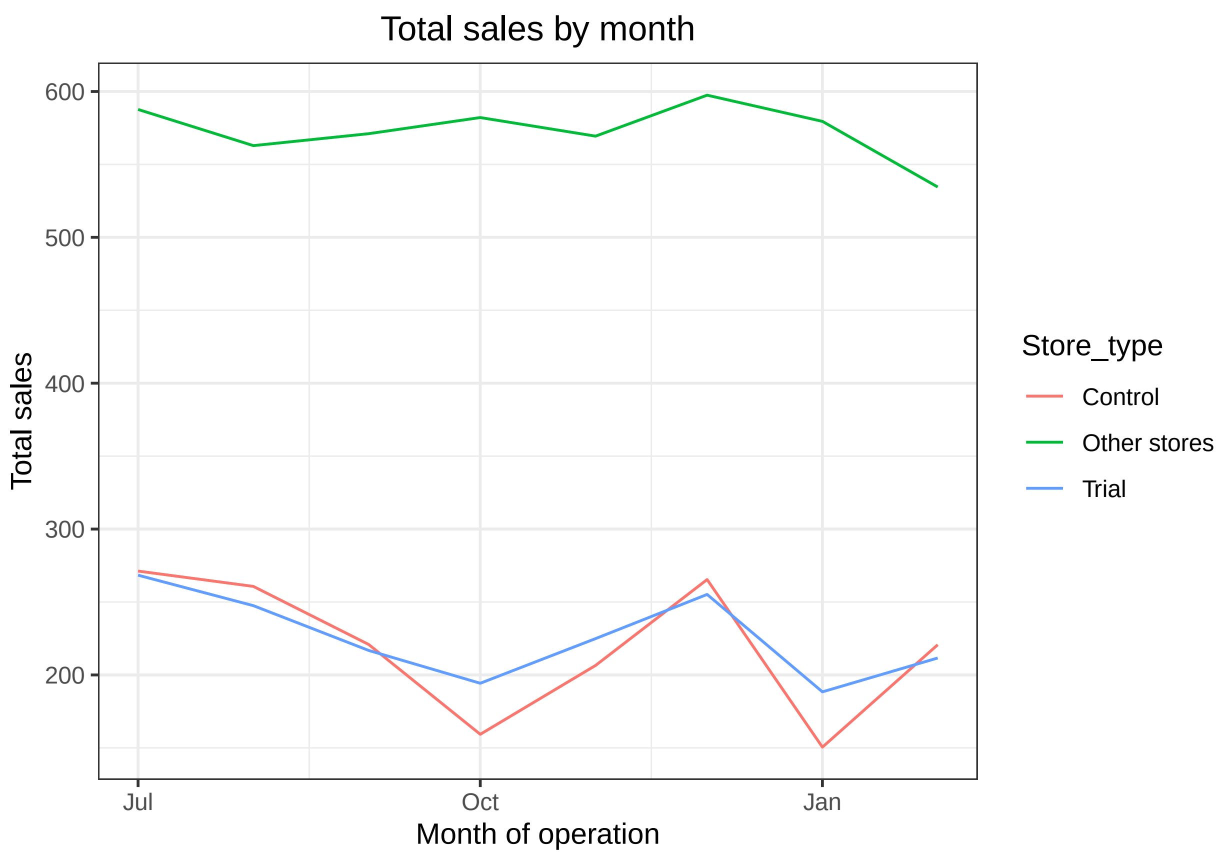
The store with the highest score is then selected as the control store since it is most similar to the trial store.

```
## Select the most appropriate control store for trial store 77 by finding the store with the highest f
control_store <- score_Control[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.

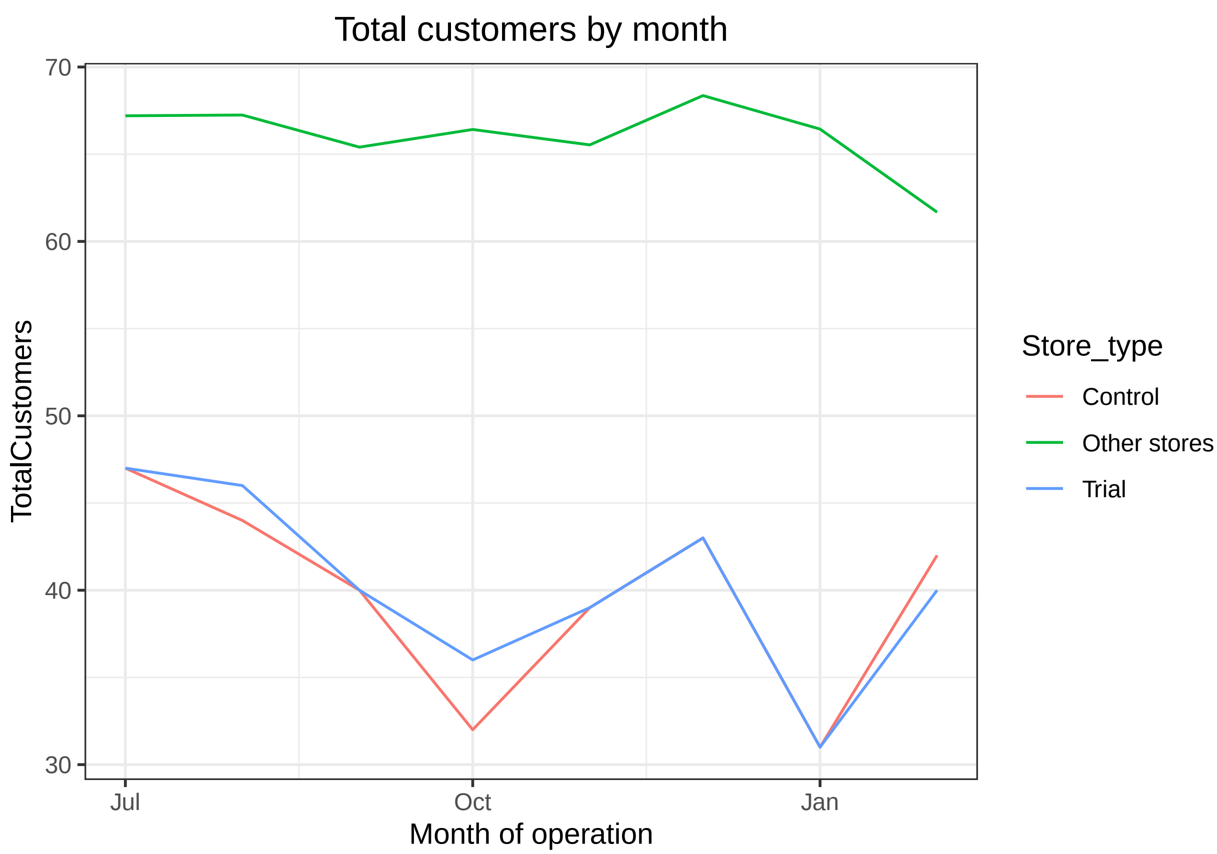
```
#### Visual checks on trends based on the drivers
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
 ifelse(STORE_NBR == control_store,
"Control", "Other stores"))
 ][, totSales := mean(totSales), by = c("YEARMONTH",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

```
## Conducting visual checks on customer count trends by comparing the trial store to the control store
measureOverTimeCusts <- measureOverTime

pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
 ifelse(STORE_NBR == control_store,
"Control", "Other stores"))
```

```
 ][, mean_cust := mean(nCustomers), by = c("YEARMONTH",
"Store_type")
 ][, TransactionMonth := as.Date(paste(as.numeric(YEARMONTH) %/%
100, as.numeric(YEARMONTH) %% 100, 1, sep = "-"), "%Y-%m-%d")
 ][YEARMONTH < 201903 , ]

ggplot(pastCustomers, aes(TransactionMonth, mean_cust, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "TotalCustomers", title = "Total 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(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]
```

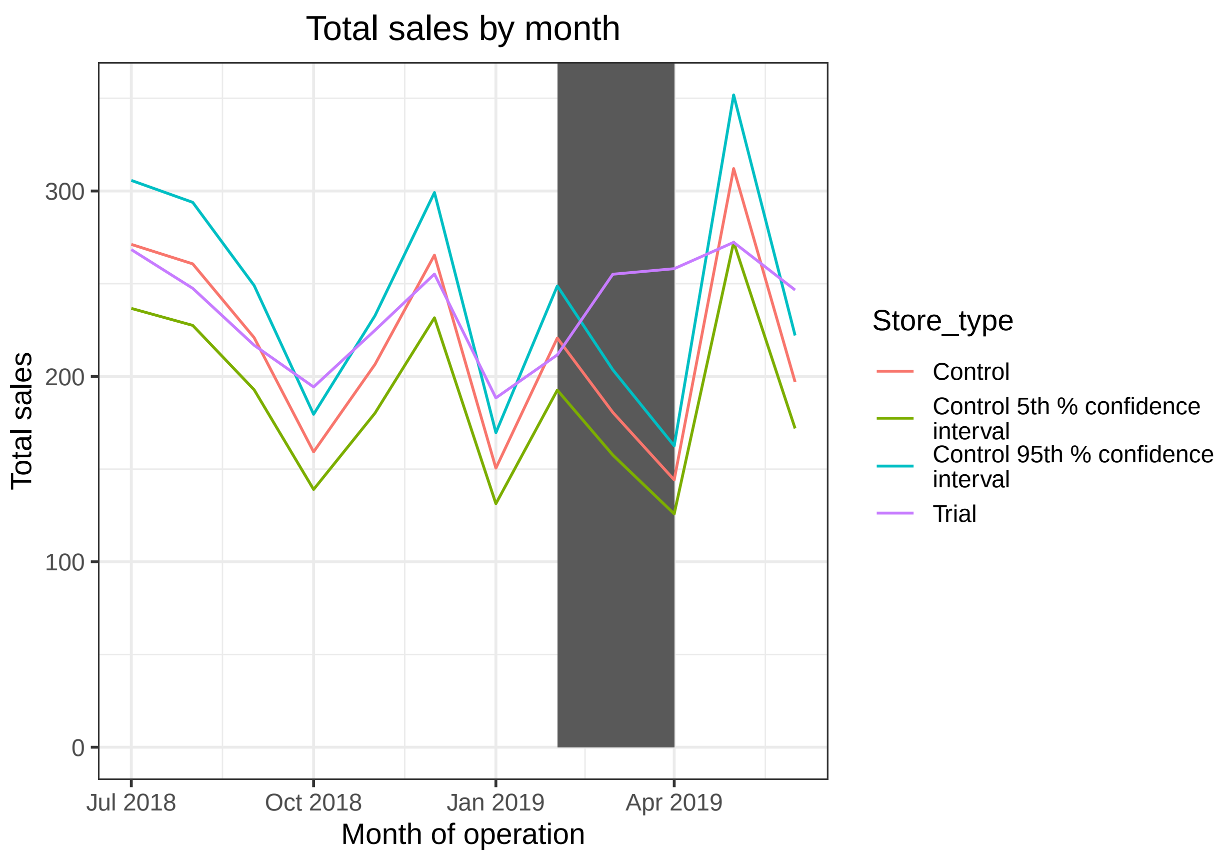
Now that we have comparable sales figures for the control store, we can calculate the percentage difference between the scaled control sales and the trial store's sales during the trial period.

```
## 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)/controlSales]

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

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(as.numeric(YEARMONTH) %/% 100,
 as.numeric(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
```

Let's see if the difference is significant!

```
measureOverTimeSales <- measureOverTime
#### Trial and control store total sales
## Creating new variables Store_type, totSales and TransactionMonth in the data table.
pastSales <- measureOverTimeSales[, totSales := mean(totSales), by =
c("YEARMONTH", "Store_type")
 ][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")
```



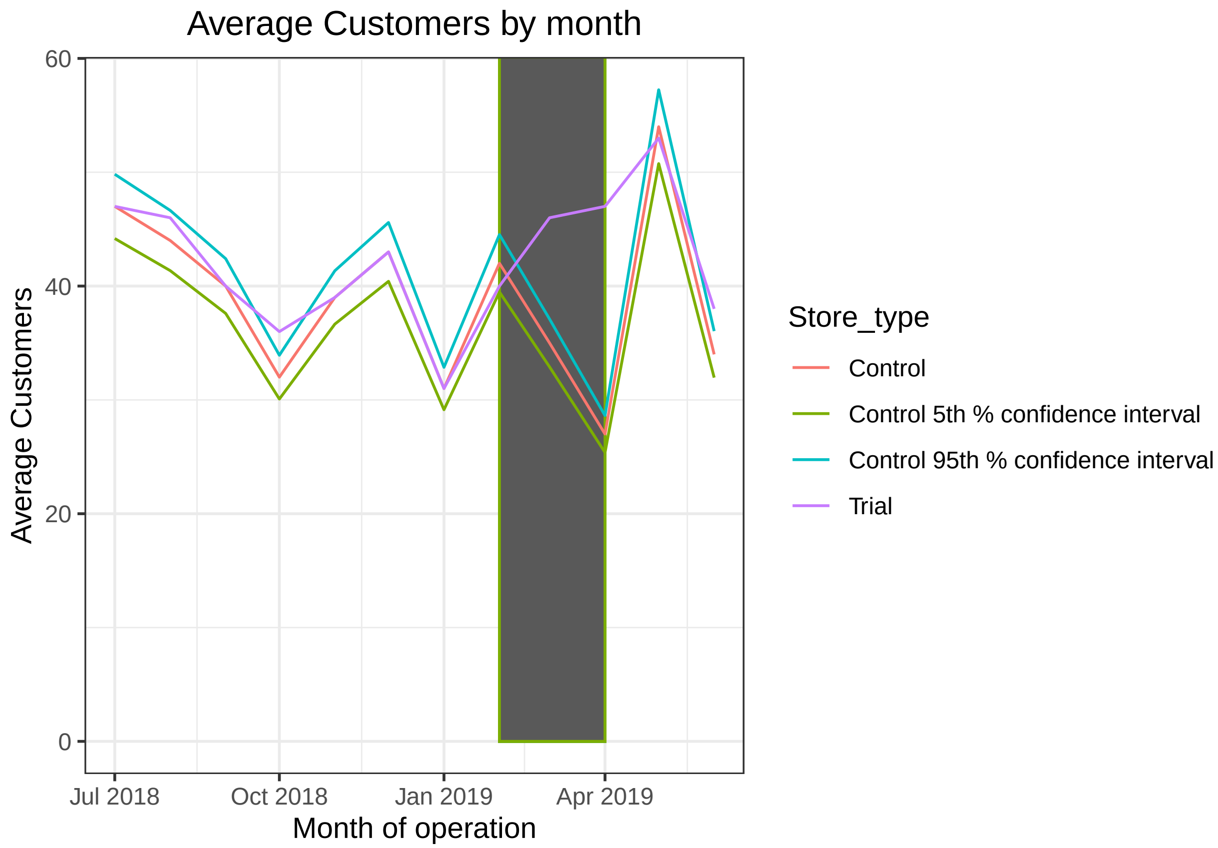
The results show that the trial in store 77 is significantly different to its control store in the trial period as the trial store performance lies outside the 5% to 95% confidence interval of the control store in two of the three trial months. Let's have a look at assessing this for number of customers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

```
####  Compute a scaling factor to align control store customer  counts to our trial store.
#### Then, apply the scaling factor to control store customer counts.

scalingFactorForControlcust <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(nCustomers)] / preTrialMeasures[STORE_NBR == control_store & YEARMONTH < 201902
#### Apply the scaling factor
measureOverTimecusts <- measureOverTime
scaledControlcustomers <- measureOverTimecusts[STORE_NBR == control_store, ][ ,
controlCustomers := nCustomers * scalingFactorForControlcust]

percentageDiff <- merge(scaledControlcustomers[, c("YEARMONTH",
"controlCustomers")],
 measureOverTime[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 th
```

```
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7
#### Trial and control store number of customers
pastCustomers <- measureOverTimeCusts[, nCusts := mean(nCustomers), by =
c("YEARMONTH", "Store_type")
 ][Store_type %in% c("Trial", "Control"), ]
#### Control store 95th percentile
pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",
 ][, nCusts := nCusts * (1 + stdDev * 2)
 ][, Store_type := "Control 95th % confidence interval"]

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

trialAssessment <- rbind(pastCustomers, pastCustomers_Controls95,
pastCustomers_Controls5)
#### Over to you! Plot everything into one nice graph.
#### Hint: geom_rect creates a rectangle in the plot. Use this to highlight the trial period in our gra
ggplot(trialAssessment, aes(TransactionMonth, nCusts, color = Store_type)) +
  geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
    aes(xmin = min(TransactionMonth),
      xmax = max(TransactionMonth),
      ymin = 0 ,
      ymax = Inf),
      show.legend = FALSE) +
  geom_line() +
  labs(x = "Month of operation", y = "Average Customers",
      title = "Average 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

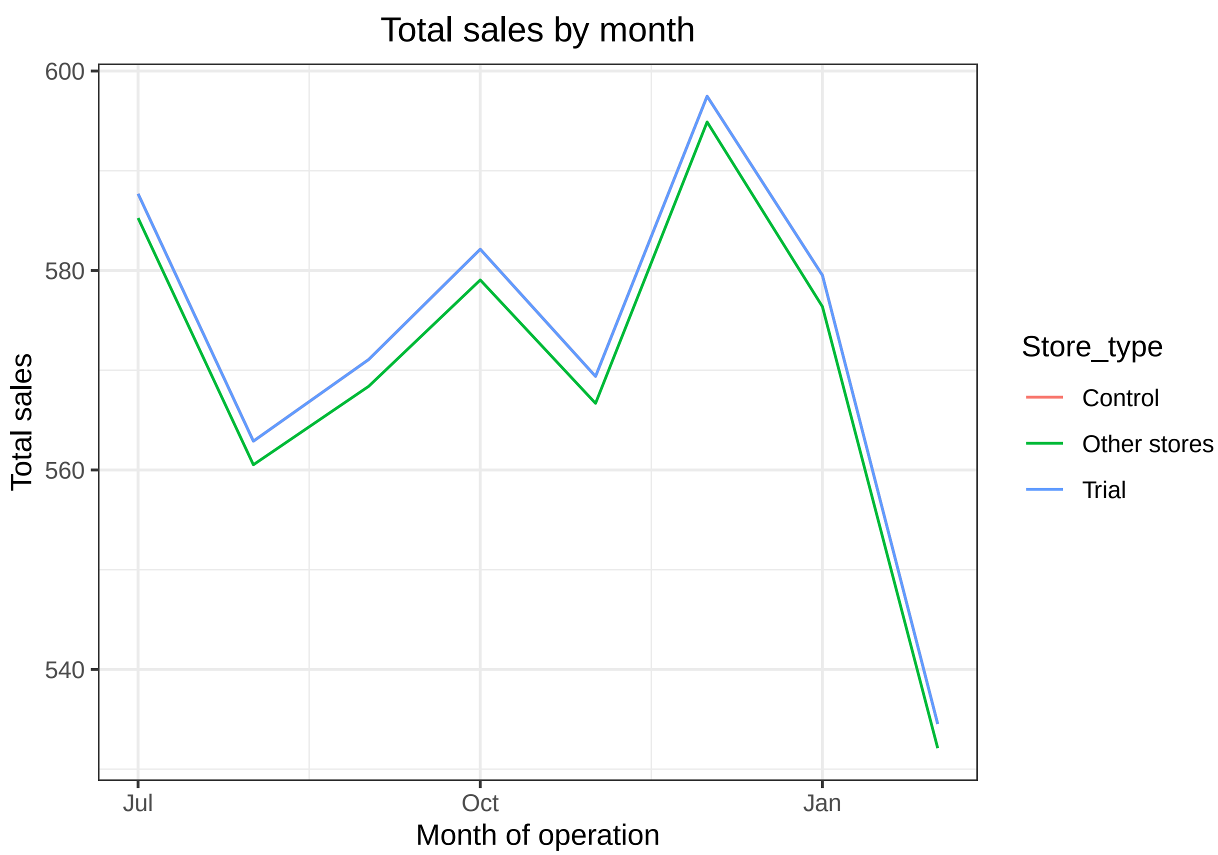
```
#Use the functions we created earlier to calculate correlations and magnitude for each potential contro
trial_store <- 86
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales),  trial_store)

corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(totSales), trial_store)

#### Then, use the functions for calculating magnitude.
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales),
trial_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures,
quote(nCustomers), trial_store)
#### Now, create a combined score composed of correlation and magnitude
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales , by = "Store2")[, scoreNSales := 0.5 * corr_measur
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = "Store2")[, scoreNCust := 0.5 * c

#### Finally, combine scores across the drivers using a simple average.
score_Control <- merge(score_nSales, score_nCustomers, by = "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)
```

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

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

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

```r
## 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("YEARMONTH",
"Store_type")
 ][, TransactionMonth := as.Date(paste(as.numeric(YEARMONTH) %/%
100, as.numeric(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")
```



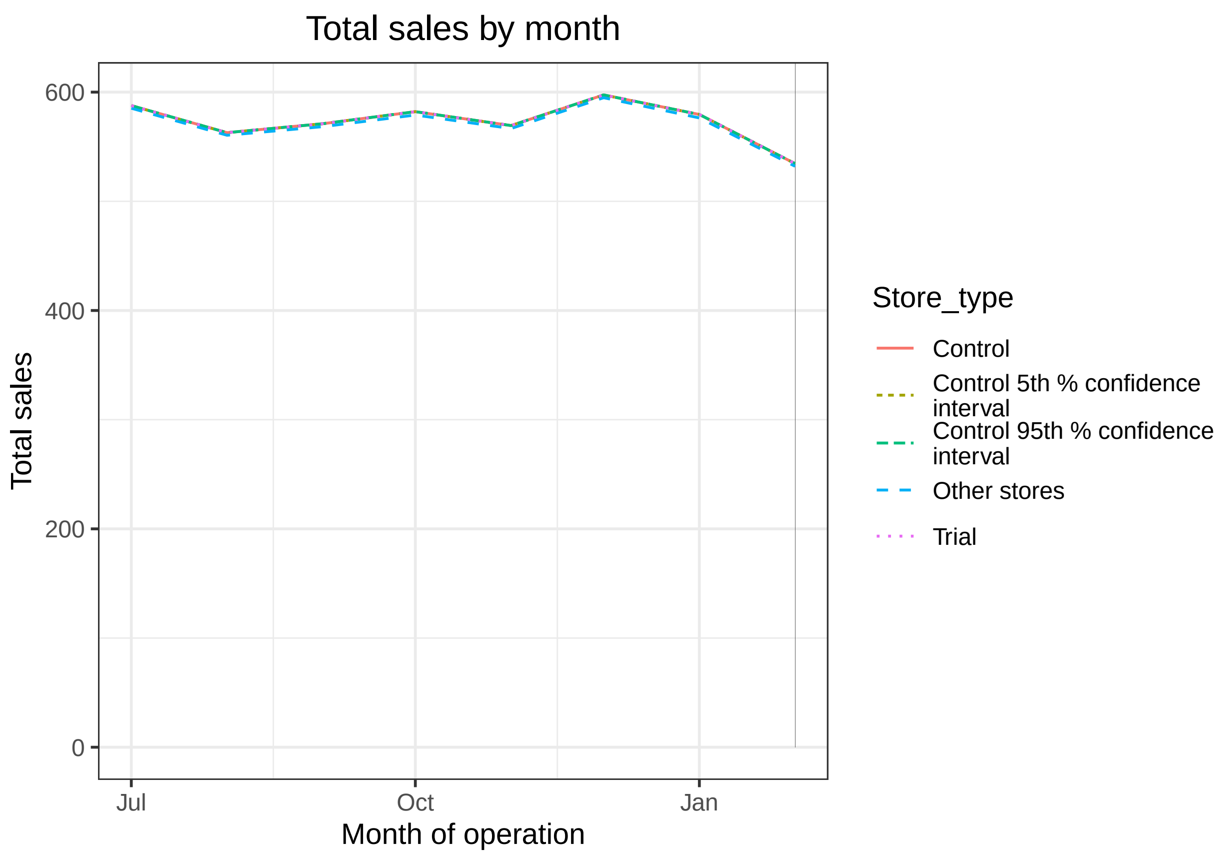
We can see that they follow a similar trend. Its safe to assume this is true across both drivers.

```r
#### 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]



#percentage 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-trialperiod, let's take the

stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7
#### Trial and control store total sales

measureOverTimeSales <- measureOverTime

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(as.numeric(YEARMONTH) %/%
100, as.numeric(YEARMONTH) %% 100, 1, sep = "-"), "%Y-%m-%d")
 ][YEARMONTH < 201903 , ]
```

```
##       YEARMONTH STORE_NBR totSales nCustomers nTxn nTxnPerCust nChipsPerTxn
##    1:    201807         1 585.2638         47   49    1.042553     1.183673
##    2:    201807         2 585.2638         36   38    1.055556     1.131579
##    3:    201807         3 585.2638        108  134    1.240741     1.962687
##    4:    201807         4 585.2638        121  152    1.256198     1.986842
##    5:    201807         5 585.2638         86  111    1.290698     2.000000
##   ---
## 2106:    201902       268 532.1028         35   36    1.028571     1.250000
## 2107:    201902       269 532.1028         97  123    1.268041     2.000000
## 2108:    201902       270 532.1028         88  116    1.318182     2.000000
## 2109:    201902       271 532.1028         81   93    1.148148     2.000000
## 2110:    201902       272 532.1028         44   47    1.068182     1.893617
##       avgPricePerUnit   Store_type TransactionMonth mean_cust   nCusts
##    1:        3.328571 Other stores       2018-07-01  67.20229 67.20229
##    2:        3.223684 Other stores       2018-07-01  67.20229 67.20229
##    3:        4.432090 Other stores       2018-07-01  67.20229 67.20229
```

```
##    4:        4.369079 Other stores       2018-07-01  67.20229 67.20229
##    5:        3.440541 Other stores       2018-07-01  67.20229 67.20229
##   ---
## 2106:        3.558333 Other stores       2019-02-01  61.67557 61.67557
## 2107:        3.665854 Other stores       2019-02-01  61.67557 61.67557
## 2108:        3.474138 Other stores       2019-02-01  61.67557 61.67557
## 2109:        3.622581 Other stores       2019-02-01  61.67557 61.67557
## 2110:        4.342553 Other stores       2019-02-01  61.67557 61.67557
```

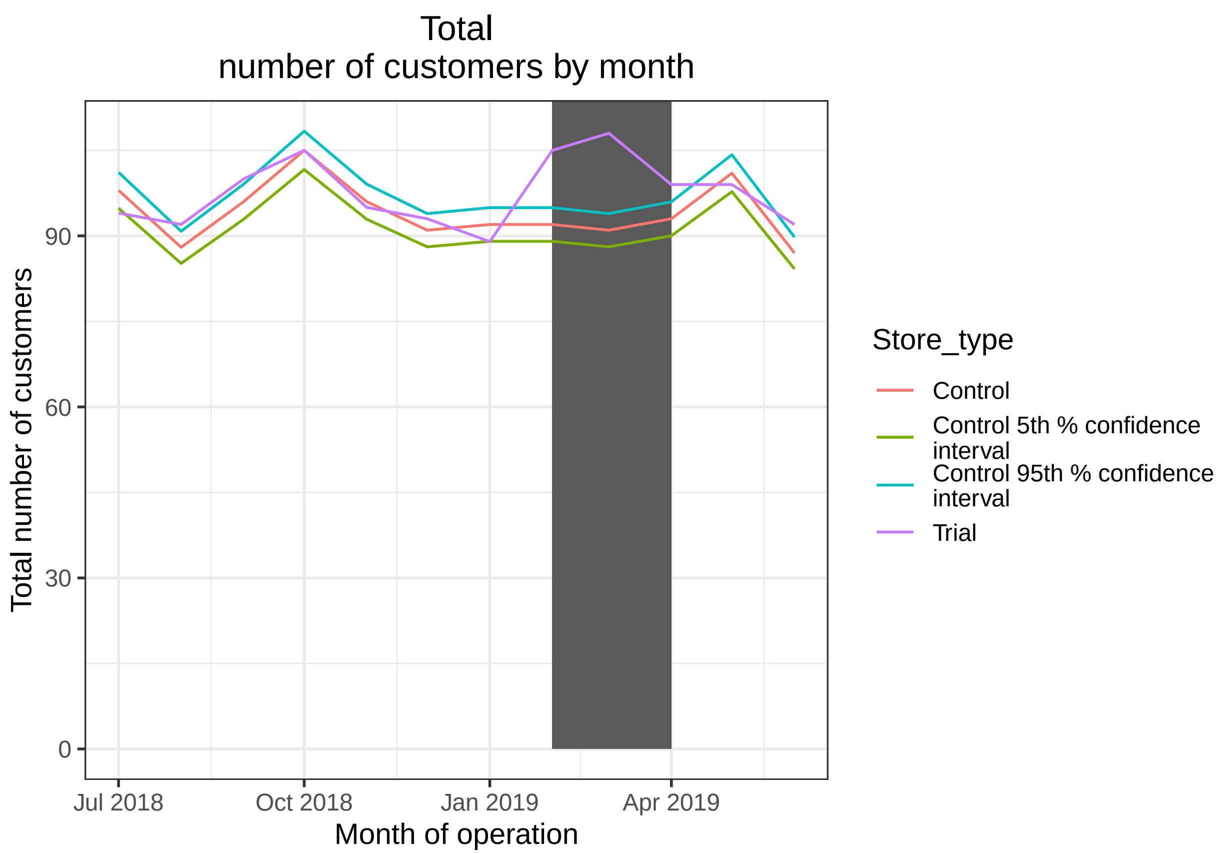
```r
#### Control store 95th percentile
pastSales_Controls95 <- pastSales[Store_type == "Control",
 ][, totSales := totSales * (1 + stdDev * 2)
 ][, Store_type := "Control 95th % confidence
interval"]
#### Control store 5th percentile
pastSales_Controls5 <- pastSales[Store_type == "Control",
 ][, totSales := 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 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.
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7
#### Trial and control store number of customers
pastCustomers <- measureOverTimeCusts[, nCusts := mean(nCustomers), by =
c("YEARMONTH", "Store_type")
][Store_type %in% c("Trial", "Control"), ]
#### Control store 95th percentile
pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",
][, nCusts := nCusts * (1 + stdDev * 2)
][, Store_type := "Control 95th % confidence
interval"]
#### Control store 5th percentile
pastCustomers_Controls5 <- pastCustomers[Store_type == "Control",
][, nCusts := nCusts * (1 - stdDev * 2)
][, Store_type := "Control 5th % confidence
interval"]
trialAssessment <- rbind(pastCustomers, pastCustomers_Controls95,
pastCustomers_Controls5)
#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, nCusts, color = Store_type)) +
geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0 , ymax =
Inf, color = NULL), show.legend = FALSE) +
geom_line() +
labs(x = "Month of operation", y = "Total number of customers", title = "Total
number of customers by month")
```



It looks like the number of customers is significantly higher in all of the three months. This seems to suggest that the trial had a significant impact on increasing the number of customers in trial store 86 but as we saw, sales were not significantly higher. We should check with the Category Manager if there were special deals in the trial store that were may have resulted in lower prices, impacting the results.

Trial store 88

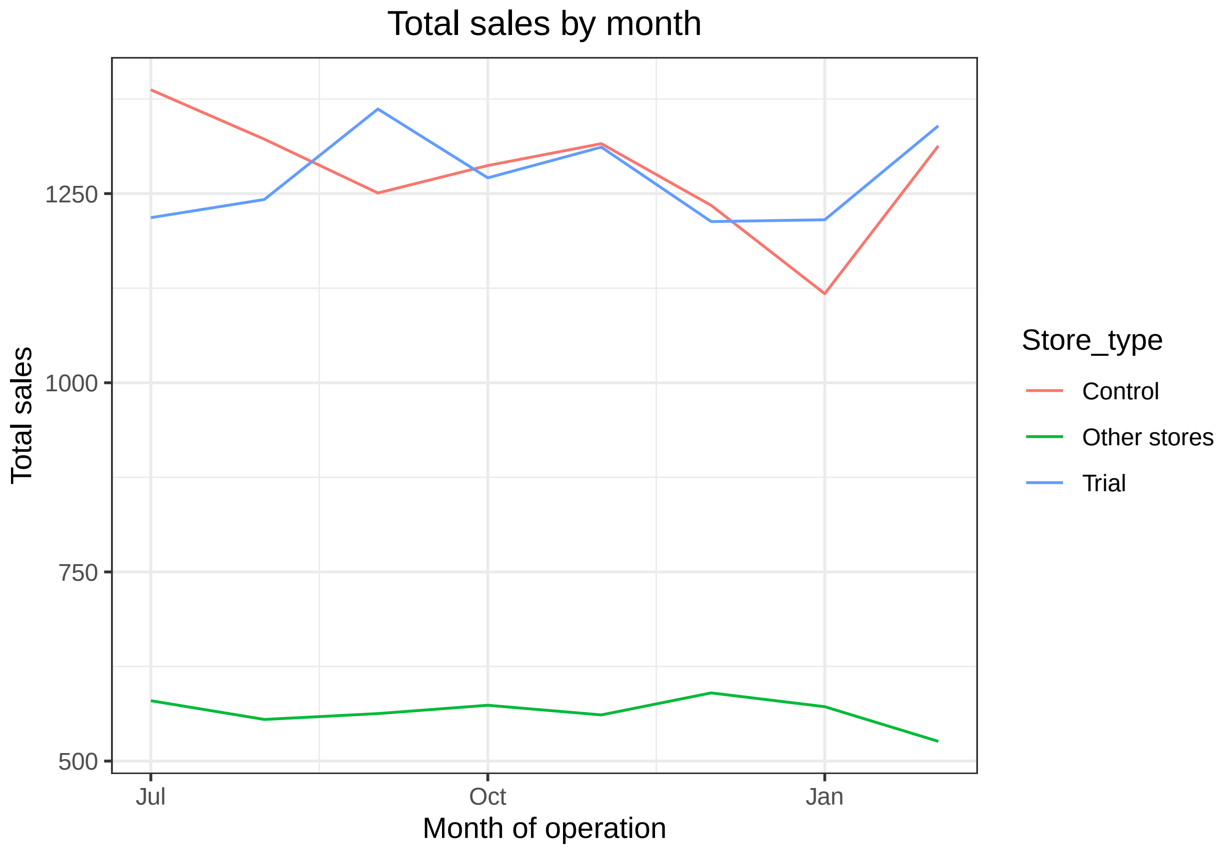
```
#### Conduct the analysis on trial store 88.
measureOverTime <- data[, list(totSales = sum(TOT_SALES),
                              nCustomers = uniqueN(LYLTY_CARD_NBR),
                              nTxn = .N,
                              nTxnPerCust=.N/uniqueN(LYLTY_CARD_NBR),
                              nChipsPerTxn = sum(PROD_QTY)/.N,
                              avgPricePerUnit = mean(price_per_qty)),
                              by = .(YEARMONTH, STORE_NBR)][order(YEARMONTH)]

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

#### Use the functions from earlier to calculate the correlation of the sales and number of customers o
trial_store <- 88

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

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

#### Then, 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, by first merging the correlations tabl


corr_weight <- 0.5

score_nSales <- merge(corr_nSales, magnitude_nSales , by ="Store2")[, scoreNSales := 0.5 * corr_measure

score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = "Store2")[, scoreNCust := 0.5 * c

#### Combine scores across the drivers by first merging our sales scores and customer scores into a sin

score_Control <- merge(score_nSales, score_nCustomers, by = "Store2")

score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]

control_store <- score_Control[order(-finalControlScore)][2, Store2]
control_store
```

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

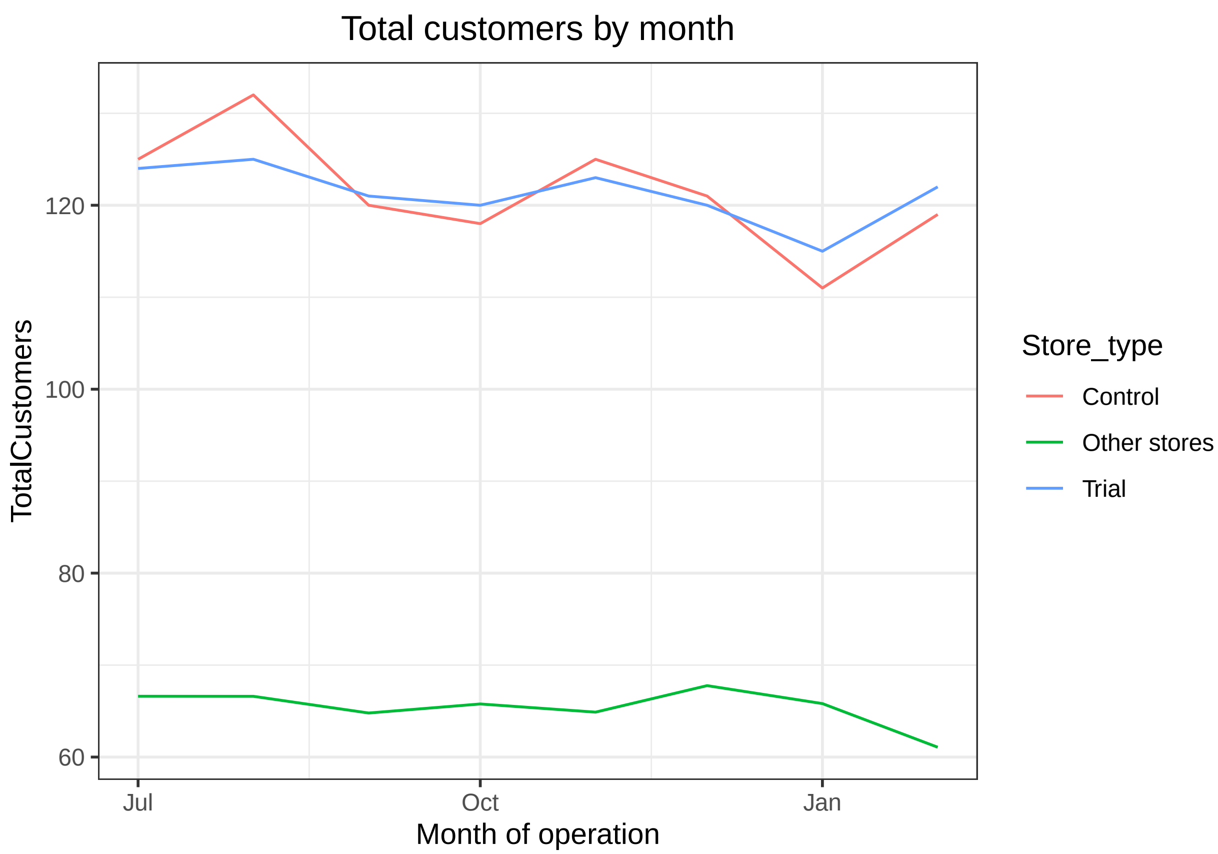
We've now found store 91 to be a suitable control store for trial store 88. Again, let's check visually if the drivers are indeed similar in the period before the trial. We'll look at total sales first.

```
## 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(as.numeric(YEARMONTH) %/%
                                      100, as.numeric(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")
```



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 mon
measureOverTimeCusts <- measureOverTime


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

ggplot(pastCustomers, aes(TransactionMonth, mean_cust, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "TotalCustomers", title = "Total 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.

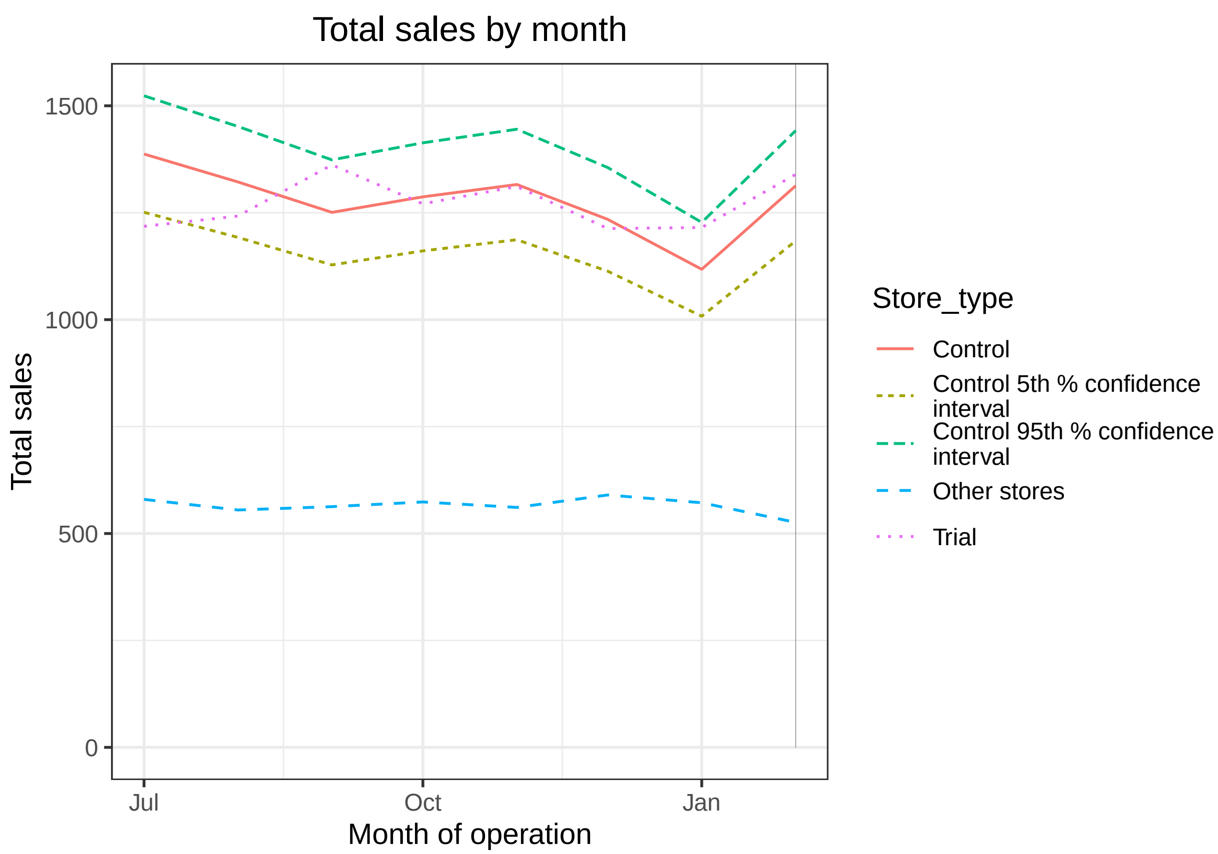
```
#### Scale pre-trial control store sales to match pre-trial trial store sales
#### Apply the scaling factor
measureOverTimeSales <- measureOverTime

scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(totSales)]/preTrialMeasures[STORE_NBR == control_store &
YEARMONTH < 201902, sum(totSales)]

scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][ ,
controlSales := totSales * scalingFactorForControlSales]

## Calculate the percentage difference between scaled control sales and trial sales
trialSales <- measureOverTimeSales[STORE_NBR == trial_store, ][ ,
trialSales := totSales]

#percentage difference
percentageDiff <- merge(trialSales,
 scaledControlSales,
by = "YEARMONTH" )[, percentageDiff := abs(trialSales - controlSales)/controlSales ]
## As our null hypothesis is that the trial period is the same as the pre-trial period, let's take the

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

```r
#### Trial and control store total sales

measureOverTimeSales <- measureOverTime

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

```
##       YEARMONTH STORE_NBR totSales nCustomers nTxn nTxnPerCust nChipsPerTxn
##    1:    201807         1 151909.1         47   49    1.042553     1.183673
##    2:    201807         2 151909.1         36   38    1.055556     1.131579
##    3:    201807         3 151909.1        108  134    1.240741     1.962687
##    4:    201807         4 151909.1        121  152    1.256198     1.986842
##    5:    201807         5 151909.1         86  111    1.290698     2.000000
##   ---
## 2106:    201902       268 137827.4         35   36    1.028571     1.250000
## 2107:    201902       269 137827.4         97  123    1.268041     2.000000
## 2108:    201902       270 137827.4         88  116    1.318182     2.000000
## 2109:    201902       271 137827.4         81   93    1.148148     2.000000
## 2110:    201902       272 137827.4         44   47    1.068182     1.893617
##       avgPricePerUnit   Store_type TransactionMonth mean_cust
##    1:        3.328571 Other stores       2018-07-01  66.61069
##    2:        3.223684 Other stores       2018-07-01  66.61069
##    3:        4.432090 Other stores       2018-07-01  66.61069
##    4:        4.369079 Other stores       2018-07-01  66.61069
##    5:        3.440541 Other stores       2018-07-01  66.61069
##   ---
## 2106:        3.558333 Other stores       2019-02-01  61.06870
## 2107:        3.665854 Other stores       2019-02-01  61.06870
## 2108:        3.474138 Other stores       2019-02-01  61.06870
## 2109:        3.622581 Other stores       2019-02-01  61.06870
## 2110:        4.342553 Other stores       2019-02-01  61.06870
```

```r
#### 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 =
```

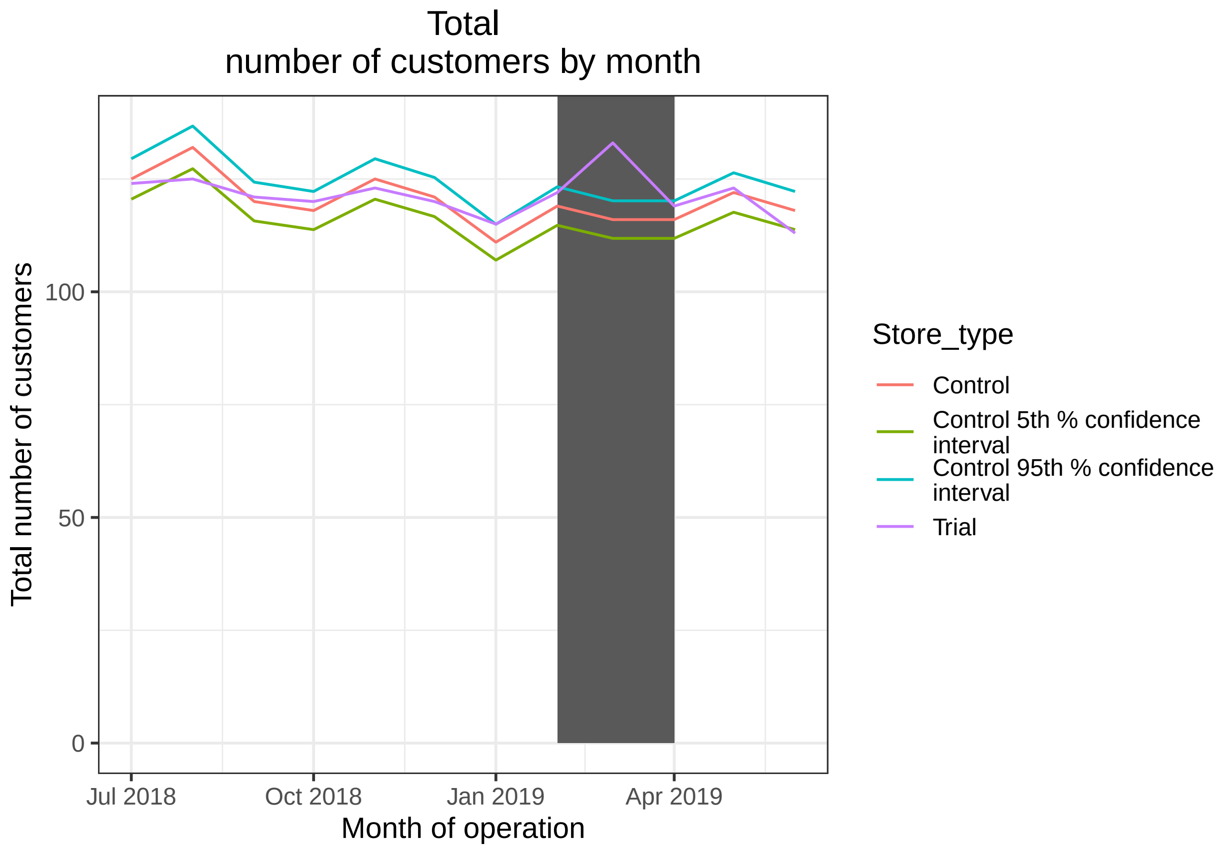
```
Inf, color = NULL), show.legend = FALSE) +
 geom_line(aes(linetype = Store_type)) +
 labs(x = "Month of operation", y = "Total sales", title = "Total sales by month")
```



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 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 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.
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")
```



Total number of customers in the trial period for the trial store is significantly higher than the control store for two out of three months, which indicates a positive trial effect.

Conclusion We've found control stores 233, 155, 237 for trial stores 77, 86 and 88 respectively. The results for trial stores 77 and 88 during the trial period show a significant difference in at least two of the three trial months but this is not the case for trial store 86. We can check with the client if the implementation of the trial was different in trial store 86 but overall, the trial shows a significant increase in sales. Now that we have finished our analysis, we can prepare our presentation to the Category Manager.