Quantium Virtual Internship - Retail Strategy and Analytics - Task 2

Kharchenko R.

2023-07-11

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Load required libraries and datasets

```r
# Load required libraries
library(data.table)

library(ggplot2)

library(readr)

library(dplyr)

library(tidyr)

# Load working file
data <- fread("QVI_mergedData.csv")

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

Select control stores for each trial store. Assess of trials.

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

Create the metrics of interest and filter to stores that are present throughout the pre-trial period

```r
# Calculate these measures over time for each store

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

head(data)

##    LYLTY_CARD_NBR       DATE STORE_NBR TXN_ID PROD_NBR
## 1:           1000 2018-10-17         1      1        5
## 2:           1002 2018-09-16         1      2       58
## 3:           1003 2019-03-07         1      3       52
```

```
## 4:           1003 2019-03-08         1      4      106
## 5:           1004 2018-11-02         1      5       96
## 6:           1005 2018-12-28         1      6       86
##                                    PROD_NAME PROD_QTY TOT_SALES PACK_SIZE
## 1: Natural Chip        Compny SeaSalt175g        2       6.0       175
## 2:  Red Rock Deli Chikn&Garlic Aioli 150g        1       2.7       150
## 3:  Grain Waves Sour    Cream&Chives 210G        1       3.6       210
## 4: Natural ChipCo      Hony Soy Chckn175g        1       3.0       175
## 5:          WW Original Stacked Chips 160g        1       1.9       160
## 6:                     Cheetos Puffs 165g        1       2.8       165
##         BRAND              LIFESTAGE PREMIUM_CUSTOMER YEARMONTH
## 1:    NATURAL  YOUNG SINGLES/COUPLES          Premium    201810
## 2:        RRD  YOUNG SINGLES/COUPLES       Mainstream    201809
## 3:    GRNWVES         YOUNG FAMILIES           Budget    201903
## 4:    NATURAL         YOUNG FAMILIES           Budget    201903
## 5: WOOLWORTHS  OLDER SINGLES/COUPLES       Mainstream    201811
## 6:    CHEETOS MIDAGE SINGLES/COUPLES       Mainstream    201812

# Next, 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 transaction 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 = .(STORE_NBR, YEARMONTH)][order(STORE_NBR, YEARMONTH)]

# Now 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, ]
head(preTrialMeasures, 10)

##     STORE_NBR YEARMONTH totSales nCustomers nTxnPerCust nChipsPerTxn
##  1:         1    201807    188.9         47    1.042553     1.183673
##  2:         1    201808    168.4         41    1.000000     1.268293
##  3:         1    201809    268.1         57    1.035088     1.203390
##  4:         1    201810    175.4         39    1.025641     1.275000
##  5:         1    201811    184.8         44    1.022727     1.222222
##  6:         1    201812    160.6         37    1.081081     1.200000
##  7:         1    201901    149.7         35    1.000000     1.171429
##  8:         2    201807    140.5         36    1.055556     1.131579
##  9:         2    201808    180.9         35    1.114286     1.282051
## 10:         2    201809    133.9         32    1.031250     1.090909
##     avgPricePerUnit
##  1:        3.256897
##  2:        3.238462
##  3:        3.776056
##  4:        3.439216
##  5:        3.360000
##  6:        3.345833
##  7:        3.651220
##  8:        3.267442
```

```
##  9:        3.618000
## 10:        3.719444

# Get a number of months in the pre-trial period to use in the next calculations
numMonthsPreTrial <- preTrialMeasures[, uniqueN(YEARMONTH)]
```

Now we need to work out a way of ranking how similar each potential control store is to the trial store.

Create a function to calculate how correlated the performance of each store is to the trial store

```
# 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 (store in storeNumbers) {
    if (store != storeComparison) {
      calculatedMeasure <- data.table("Store1" = storeComparison,
                                      "Store2" = store,
                                      "corr_measure" = cor(inputTable[STORE_NBR ==
                                                                      storeComparison,
                                                                      eval(metricCol)],
                                                           inputTable[STORE_NBR == store,
                                                                      eval(metricCol)]))

      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.

Create a function to calculate a standardized magnitude distance

```
# 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 (store in storeNumbers) {
    if (store != storeComparison) {
      calculatedMeasure <- data.table(Store1 = storeComparison,
                                      Store2 = store,
                                      YEARMONTH = inputTable[STORE_NBR ==
                                                             storeComparison,
```

```
                                                                YEARMONTH],
                                          measure = abs(inputTable[STORE_NBR ==
                                                                  storeComparison,
                                                                  eval(metricCol)]
                                                         - inputTable[STORE_NBR == store,
                                                                    eval(metricCol)]))
      calcDistTable <- rbind(calcDistTable, calculatedMeasure)
    }
  }

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

  return(finalDistTable)
}
```

Now let's use the above 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.

Select control store for trial store 77

```
# Use the created functions 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_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers), trial_store)

# Now 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 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 table with the magnitude table.

corr_weight <- 0.5

# By using (1 - corr_weight) for the weight of the magnitude score, we ensure that the sum of the weights
# for both scores is equal to 1. Thus we allow for a balanced combination of both scores,
# and adjusting corr_weight allows us to easily control the balance based on our preference or specific requirements.
```

```r
score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1", "Store2"))
score_nSales[, scoreNSales := corr_weight * corr_measure + (1 - corr_weight) *
            mag_measure]

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

Now we have a score for each of the total number of sales and a number of customers.

Let's combine the two via a simple average.

```r
# Combine scores across the drivers by merging the sales scores and customer scores into
a single table
score_Control <- merge(score_nSales, score_nCustomers, by = c("Store1", "Store2"))

# Calculate the final control score using a simple average
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]
score_Control

##      Store1 Store2 corr_measure.x mag_measure.x  scoreNSales corr_measure.y
##   1:     77      1   -0.005382429     0.9536909  0.474154250    0.337865596
##   2:     77      2   -0.251182809     0.9372067  0.343011955   -0.596491730
##   3:     77      3    0.660446832     0.3454316  0.502939207    0.755248715
##   4:     77      4   -0.347846468     0.1810682 -0.083389122   -0.305411652
##   5:     77      5   -0.139047983     0.5651305  0.213041257    0.224768439
##  ---
## 254:     77    268    0.395460337     0.9636567  0.679558507    0.369735946
## 255:     77    269   -0.466370424     0.4552162 -0.005577134   -0.247580595
## 256:     77    270    0.274854303     0.4584257  0.366639990   -0.009181744
## 257:     77    271    0.195189898     0.5727032  0.383946560    0.023634941
## 258:     77    272   -0.179646952     0.8928227  0.356587891    0.068677178
##      mag_measure.y  scoreNCust finalControlScore
##   1:     0.9391149  0.63849027        0.55632226
##   2:     0.9087322  0.15612025        0.24956610
##   3:     0.3431594  0.54920406        0.52607164
##   4:     0.2022603 -0.05157567       -0.06748239
##   5:     0.5135798  0.36917410        0.29110768
##  ---
## 254:     0.9435154  0.65662569        0.66809210
## 255:     0.3624207  0.05742008        0.02592147
## 256:     0.3910055  0.19091187        0.27877593
## 257:     0.5245199  0.27407741        0.32901199
## 258:     0.9481501  0.50841362        0.43250076
```

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

```r
# Select the most appropriate control store for trial store 77 by finding the store with
the highest final score.
control_store <- score_Control[Store1 == trial_store, .(Control_Store = Store2,
                                                        Final_Score = finalControlScore)
                               ][order(-Final_Score)][1, Control_Store]
cat("Trial Store:", 77, " Control Store:", control_store, "\n")
```

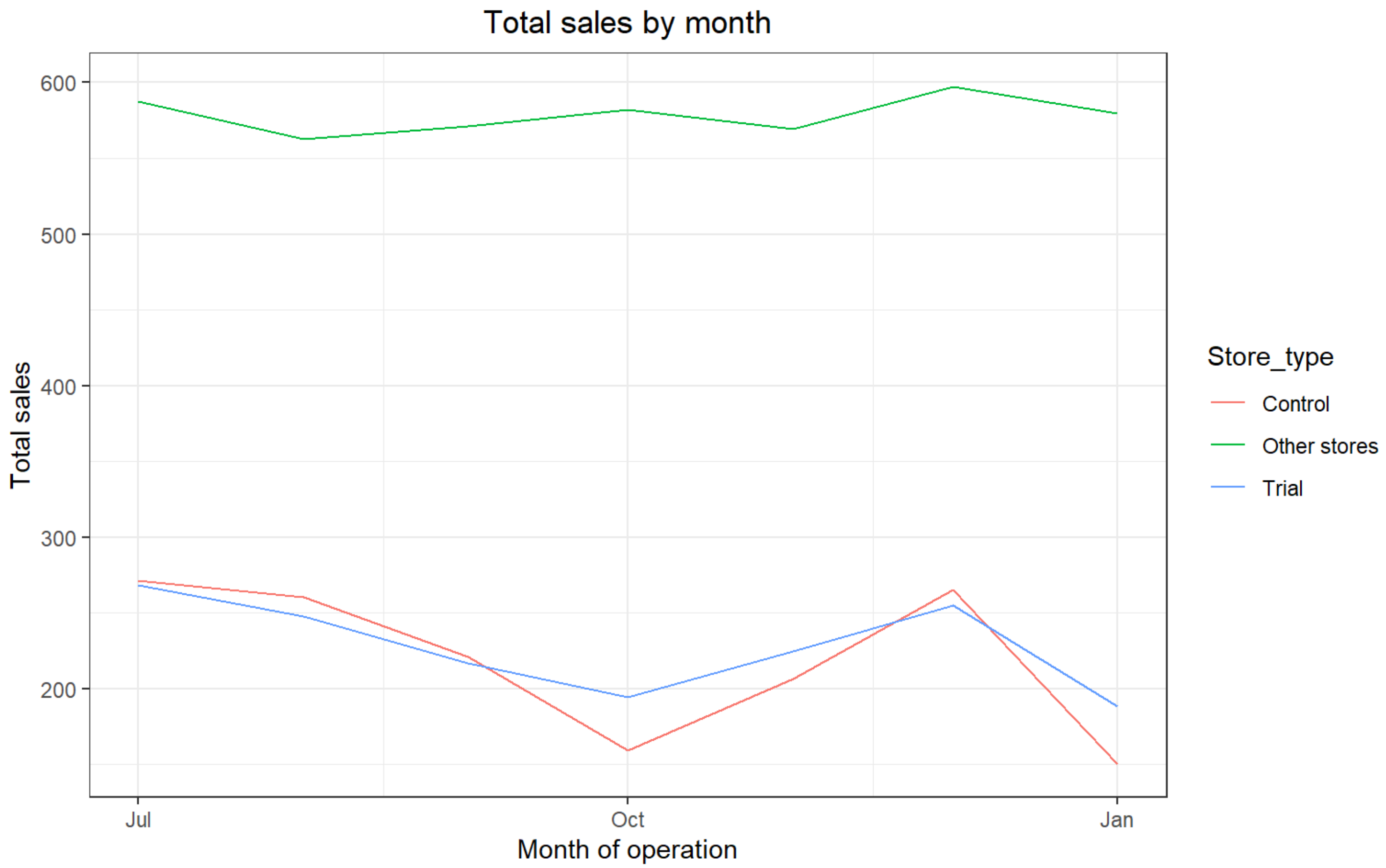
```
## Trial Store: 77  Control Store: 233
```

Now that we have found a control store, let's check visually if the drivers are indeed similar in the period before the trial.

Check the visual similarity of the test and control store drivers

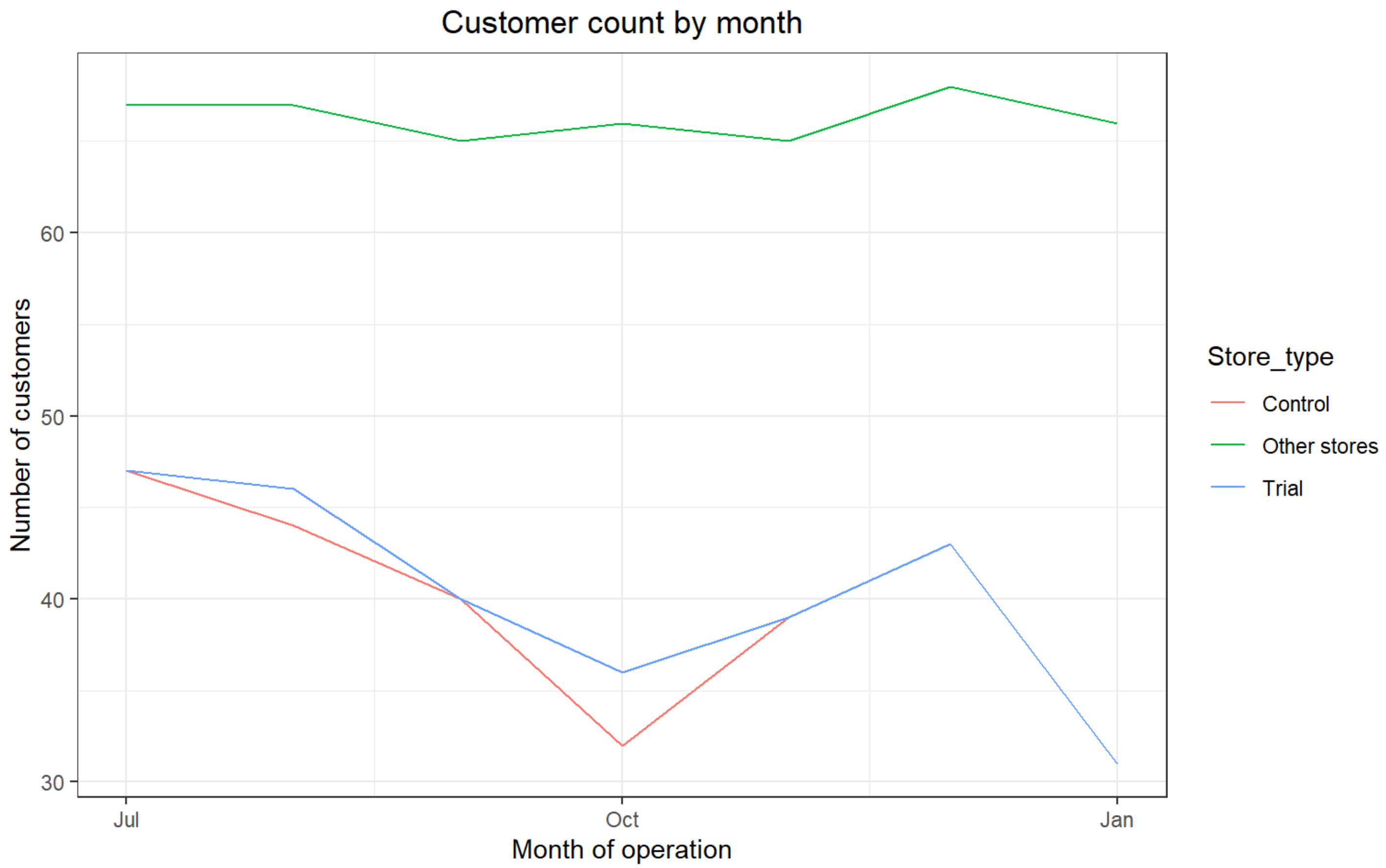
Let's look at total sales first.

```
# Visual checks on trends based on the drivers
measureOverTimeSales <- measureOverTime
measureOverTimeSales[, YEARMONTH := as.numeric(as.character(YEARMONTH))]  # Convert
YEARMONTH to numeric
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 < 201902 , ]

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



Next, the number of customers.

```
# Conduct visual checks on customer count trends by comparing the trial store to the
control store and other stores.
measureOverTimeCusts <- measureOverTime
measureOverTimeCusts[, YEARMONTH := as.integer(as.character(YEARMONTH))]  # Convert
YEARMONTH to integer
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store,
                                                              "Trial",
                                                          ifelse(STORE_NBR ==
                                                                 control_store, "Control",
                                                                 "Other stores"))
                                      ][, nCustomers := as.integer(mean(nCustomers)), by =
                                                          c("YEARMONTH", "Store_type")
                                      ][, TransactionMonth := as.Date(paste(YEARMONTH %/%
                                                       100, YEARMONTH %% 100, 1, sep = "-"),
                                                                     "%Y-%m-%d")
                                      ][YEARMONTH < 201902 , ]

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



Observations

- As can be seen from the above visuals, trial store 77 and control store 233 are indeed very close to each other in terms of performance during the pre-trial period. This is especially noticeable when compared to other stores.

Assessment of trial at store 77

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 controlling for any differences between the two stores outside of the trial period.

Assess the trial in terms of 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]
                                            scaledControlSales[, controlSales, totSales]
##      totSales controlSales
##   1:    271.2     281.9808
##   2:    260.7     271.0634
##   3:    220.9     229.6813
##   4:    159.3     165.6326
```

```
## 5:    206.5     214.7089
## 6:    265.4     275.9503
## 7:    150.5     156.4827
## 8:    220.7     229.4733
## 9:    180.6     187.7793
## 10:    144.2     149.9323
## 11:    312.1     324.5067
## 12:    197.0     204.8312
```

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
trialPeriodStart <- 201902
trialPeriodEnd <- 201904
trialMonths <- c(201902, 201903, 201904)

percentageDiff <- merge(scaledControlSales[, c("STORE_NBR","YEARMONTH","totSales",
                                               "Store_type","controlSales")],
                        measureOverTimeSales[STORE_NBR == trial_store,
                                    c("STORE_NBR","YEARMONTH","totSales","Store_type")],
                        by = c("YEARMONTH")
                        )[, percentageDiff := abs(controlSales - totSales.y) /
                                                  controlSales]

trialPercentageDiff <- percentageDiff[YEARMONTH %in% trialMonths]
trialPercentageDiff
##    YEARMONTH STORE_NBR.x totSales.x Store_type.x controlSales STORE_NBR.y
## 1:    201902         233      220.7      Control     229.4733          77
## 2:    201903         233      180.6      Control     187.7793          77
## 3:    201904         233      144.2      Control     149.9323          77
##    totSales.y Store_type.y percentageDiff
## 1:      211.6        Trial     0.07788855
## 2:      255.1        Trial     0.35850987
## 3:      258.1        Trial     0.72144372
```

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])

# Define degrees of freedom
degreesOfFreedom <- numMonthsPreTrial - 1

# We will test with a null hypothesis of there being 0 difference between trial and
control stores.
# Calculate the t-values for the trial months.
# After that, find the 95th percentile of the t-distribution with the appropriate degrees
of freedom
# to check whether the hypothesis is statistically significant.
```

```r
# Calculate the t-values for the trial months
trialPercentageDiff[, tValue := (as.numeric(trialPercentageDiff$percentageDiff) - 0) /
                                 stdDev]

# Find the 95th percentile of the t-distribution with the appropriate degrees of freedom
trialPercentageDiff[, tCritical := qt(0.95, df = degreesOfFreedom)]

# Check whether the t-values are statistically significant
trialPercentageDiff[, isSignificant := tValue > tCritical]
trialPercentageDiff
##    YEARMONTH STORE_NBR.x totSales.x Store_type.x controlSales STORE_NBR.y
## 1:    201902         233      220.7      Control     229.4733          77
## 2:    201903         233      180.6      Control     187.7793          77
## 3:    201904         233      144.2      Control     149.9323          77
##    totSales.y Store_type.y percentageDiff    tValue tCritical isSignificant
## 1:      211.6        Trial     0.07788855  1.223912   1.94318         FALSE
## 2:      255.1        Trial     0.35850987  5.633494   1.94318          TRUE
## 3:      258.1        Trial     0.72144372 11.336505   1.94318          TRUE
```

Observations

1. In February 2019 the percentage difference is 7.79%, and the t-value is 1.22. It is not statistically significant.

2. In March 2019 the percentage difference is 35.85%, and the t-value is 5.63. It is statistically significant.

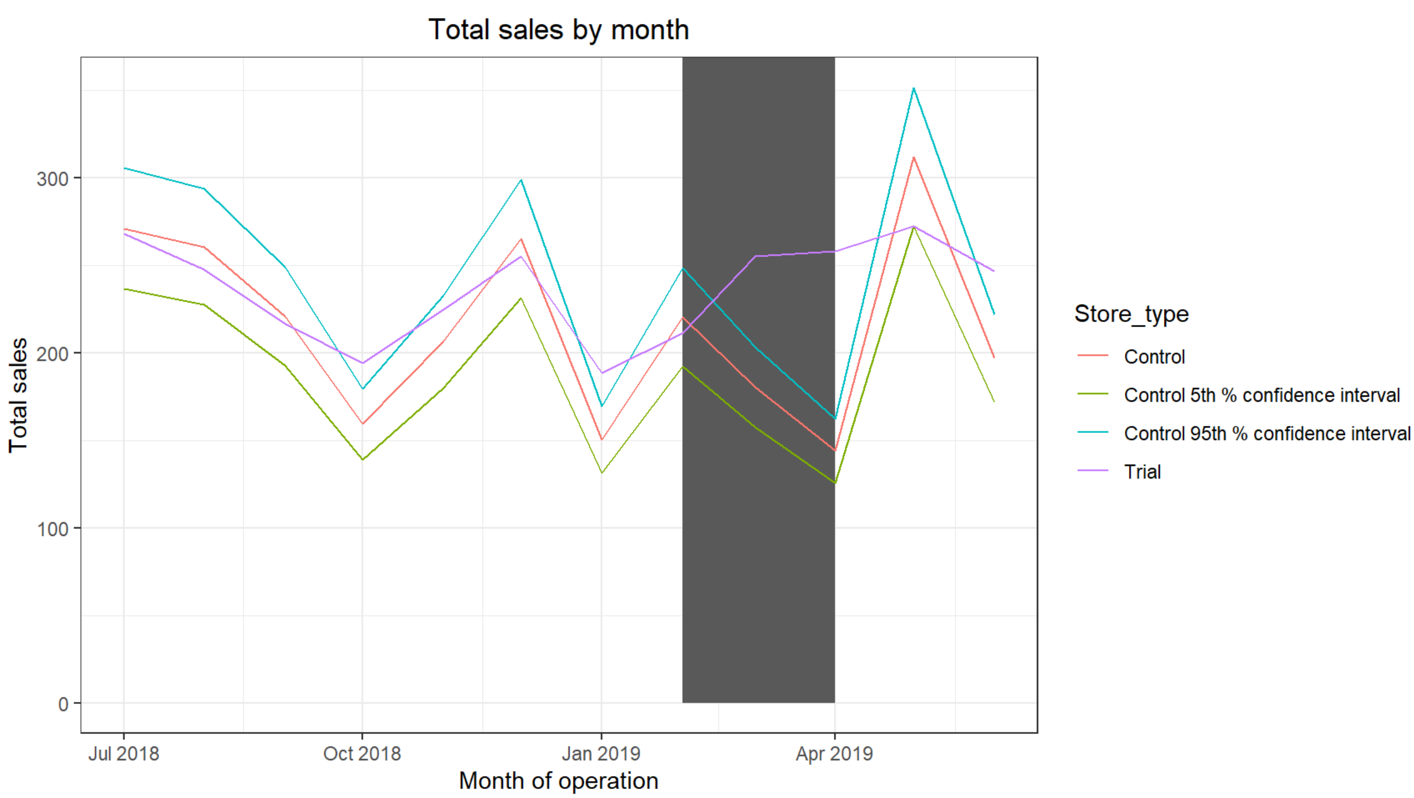
3. In April 2019 the percentage difference is 72.14%, and the t-value is 11.34. It is statistically significant.

4. 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 store, and the 5th and 95th percentile value of sales of the control store.

```r
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"), ]
```

```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)

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



Observations

- The results show that the trial in store 77 is significantly different from its control store 233 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.

Assess the trial in terms of number of customers

```r
# Scale pre-trial control customer counts to match pre-trial trial store customer counts
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]
scaledControlCustomers[, controlCustomers, nCustomers]
##     nCustomers controlCustomers
##  1:         47         48.02174
##  2:         44         44.95652
##  3:         40         40.86957
##  4:         32         32.69565
##  5:         39         39.84783
##  6:         43         43.93478
##  7:         31         31.67391
##  8:         42         42.91304
##  9:         35         35.76087
## 10:         27         27.58696
## 11:         54         55.17391
## 12:         34         34.73913
```

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

```r
# Calculate the percentage difference between scaled control and trial customer counts
percentageDiff <- merge(scaledControlCustomers[,  c("STORE_NBR","YEARMONTH","nCustomers",
                                                   "Store_type","controlCustomers")],
                        measureOverTimeCusts[STORE_NBR == trial_store, c("STORE_NBR",
                                                 "YEARMONTH","nCustomers","Store_type")],
                        by = c("YEARMONTH")
                        )[, percentageDiff := abs(controlCustomers - nCustomers.y) /
                                                  controlCustomers]

trialPercentageDiff <- percentageDiff[YEARMONTH %in% trialMonths]
trialPercentageDiff
##    YEARMONTH STORE_NBR.x nCustomers.x Store_type.x controlCustomers STORE_NBR.y
## 1:    201902         233           42      Control         42.91304          77
## 2:    201903         233           35      Control         35.76087          77
## 3:    201904         233           27      Control         27.58696          77
##    nCustomers.y Store_type.y percentageDiff
## 1:           40        Trial     0.06788247
## 2:           46        Trial     0.28632219
## 3:           47        Trial     0.70370370
```

Let's see if the difference is significant!

```r
# 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 <- numMonthsPreTrial - 1
```

```r
# We will test with a null hypothesis of there being 0 difference between trial and
control stores.
# Calculate the t-values for the trial months.
# After that, find the 95th percentile of the t-distribution with the appropriate degrees
of freedom
# to check whether the hypothesis is statistically significant.

# Calculate the t-values for the trial months
trialPercentageDiff[, tValue := (as.numeric(trialPercentageDiff$percentageDiff) - 0) /
                                 stdDev]

# Find the 95th percentile of the t-distribution with the appropriate degrees of freedom
trialPercentageDiff[, tCritical := qt(0.95, df = degreesOfFreedom)]

# Check whether the t-values are statistically significant
trialPercentageDiff[, isSignificant := tValue > tCritical]
trialPercentageDiff
##    YEARMONTH STORE_NBR.x nCustomers.x Store_type.x controlCustomers STORE_NBR.y
## 1:    201902         233           42      Control         42.91304          77
## 2:    201903         233           35      Control         35.76087          77
## 3:    201904         233           27      Control         27.58696          77
##    nCustomers.y Store_type.y percentageDiff   tValue tCritical isSignificant
## 1:           40        Trial     0.06788247  2.25947   1.94318          TRUE
## 2:           46        Trial     0.28632219  9.53024   1.94318          TRUE
## 3:           47        Trial     0.70370370 23.42279   1.94318          TRUE
```

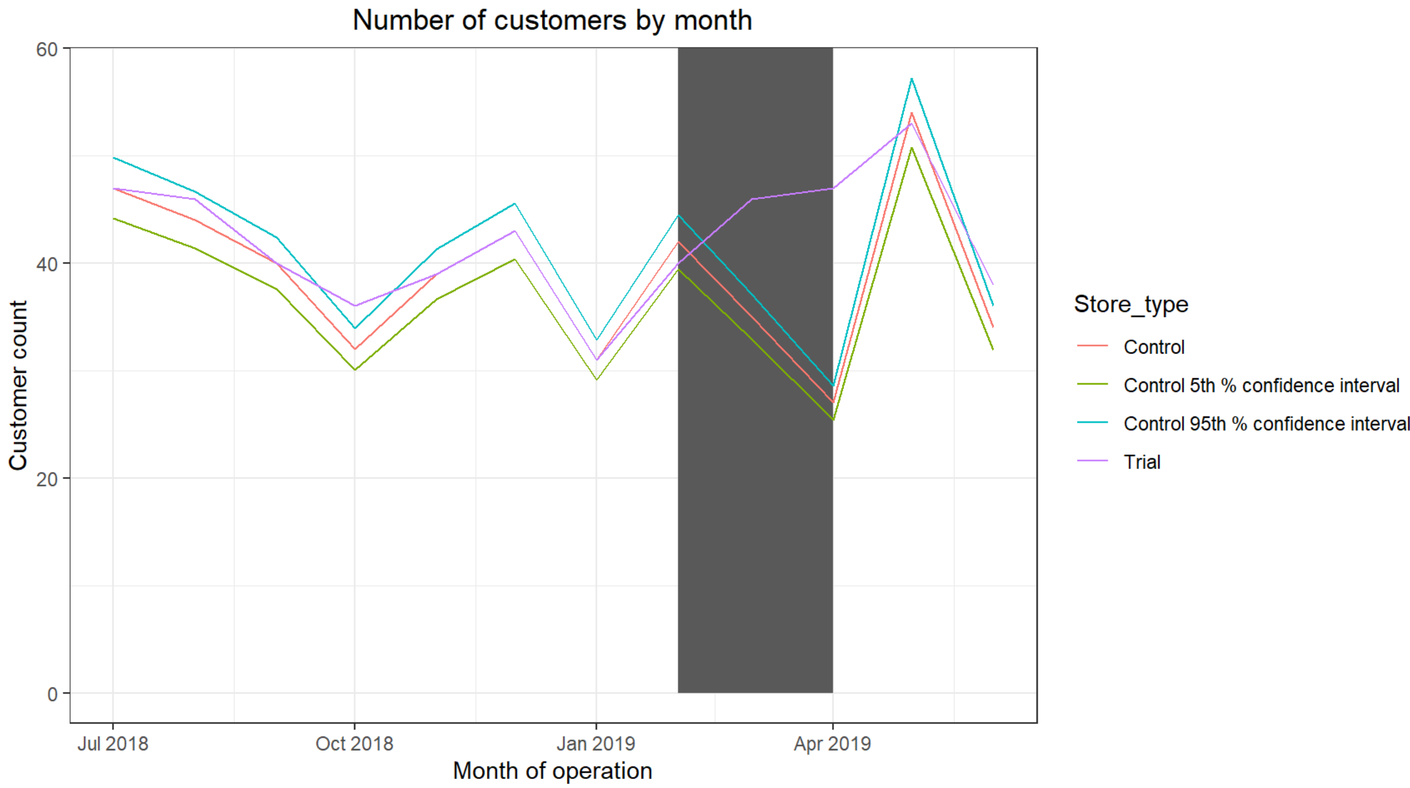
Observations

1. In February 2019 the percentage difference is 6.79%, and the t-value is 2.26. It is statistically significant.
2. In March 2019 the percentage difference is 28.63%, and the t-value is 9.53. It is statistically significant.
3. In April 2019 the percentage difference is 70.37%, and the t-value is 23.42. It is statistically significant.
4. We can observe that the t-value is much larger than the 95th percentile value of the t-distribution for all three months, i.e., the increase in customer counts in the trial store during the trial period is statistically greater than in the control store.

Let's create a more visual version of this by plotting the customer counts of the control store, the customer counts of the trial store, and the 5th and 95th percentile values of the control store.

```r
measureOverTimeCusts <- measureOverTime

# Trial and control store customer counts
# Create new variables `Store_type`, `nCusts` and `TransactionMonth` in the data table.
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store,
                                                             "Trial",
                                                        ifelse(STORE_NBR==control_store,
                                                               "Control",
                                                               "Other stores"))
                                 ][, nCusts := mean(nCustomers), 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
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 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 = "Customer count",
       title = "Number of customers by month")
```



Observations

- The results show that the trial in store 77 is significantly different from its control store 233 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.

Select control store for trial store 86

```
# Define the measure calculations to use during the analysis.
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 = .(STORE_NBR, YEARMONTH)][order(STORE_NBR, YEARMONTH)]

# Now 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, ]

# Get a number of months in the pre-trial period to use in the next calculations
numMonthsPreTrial <- preTrialMeasures[, uniqueN(YEARMONTH)]
# Use the created functions to calculate correlations against store 86 using total sales
and number of customers

trial_store <- 86
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales), trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers), trial_store)
```

```
# Now 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 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.

```
# Create a combined score composed of correlation and magnitude,
# by first merging the correlations table with the magnitude table.

corr_weight <- 0.5

# By using (1 - corr_weight) for the weight of the magnitude score, we ensure that the
sum of the weights
# for both scores is equal to 1. Thus we allow for a balanced combination of both scores,
# and adjusting corr_weight allows us to easily control the balance based on our
preference or specific requirements.

score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1", "Store2"))
score_nSales[, scoreNSales := corr_weight * corr_measure + (1 - corr_weight) *
                              mag_measure]

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

Now we have a score for each of the total number of sales and a number of customers.

Let's combine the two via a simple average.

```
# Combine scores across the drivers by merging the sales scores and customer scores into
a single table
score_Control <- merge(score_nSales, score_nCustomers, by = c("Store1", "Store2"))

# Calculate the final control score using a simple average
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]
score_Control

##      Store1 Store2 corr_measure.x mag_measure.x scoreNSales corr_measure.y
##   1:     86      1     0.36473363     0.2161616  0.29044762    0.384378894
##   2:     86      2    -0.52649154     0.1742579 -0.17611683   -0.064384086
##   3:     86      3     0.13978875     0.7554657  0.44762721    0.063780081
##   4:     86      4     0.03561817     0.5108346  0.27322637   -0.006241881
##   5:     86      5     0.44682291     0.9121176  0.67947028    0.099455888
##  ---
## 254:     86    268    -0.40807020     0.2436171 -0.08222656   -0.024864582
## 255:     86    269     0.74743234     0.9162900  0.83186119    0.311707212
## 256:     86    270    -0.73061378     0.8417507  0.05556847   -0.699534793
## 257:     86    271     0.55789426     0.9035345  0.73071437    0.286874462
## 258:     86    272     0.34156742     0.4324564  0.38701193   -0.429957137
##      mag_measure.y   scoreNCust finalControlScore
##   1:     0.4386618  0.411520331        0.35098397
##   2:     0.3609889  0.148302401       -0.01390721
```

```
##   3:     0.9157076  0.489743843        0.46868553
##   4:     0.7784629  0.386110487        0.32966843
##   5:     0.9059452  0.502700562        0.59108542
##  ---
## 254:     0.4127411  0.193938243        0.05585584
## 255:     0.9252952  0.618501206        0.72518120
## 256:     0.8692618  0.084863517        0.07021599
## 257:     0.8977030  0.592288731        0.66150155
## 258:     0.4167748 -0.006591159        0.19021038
```

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

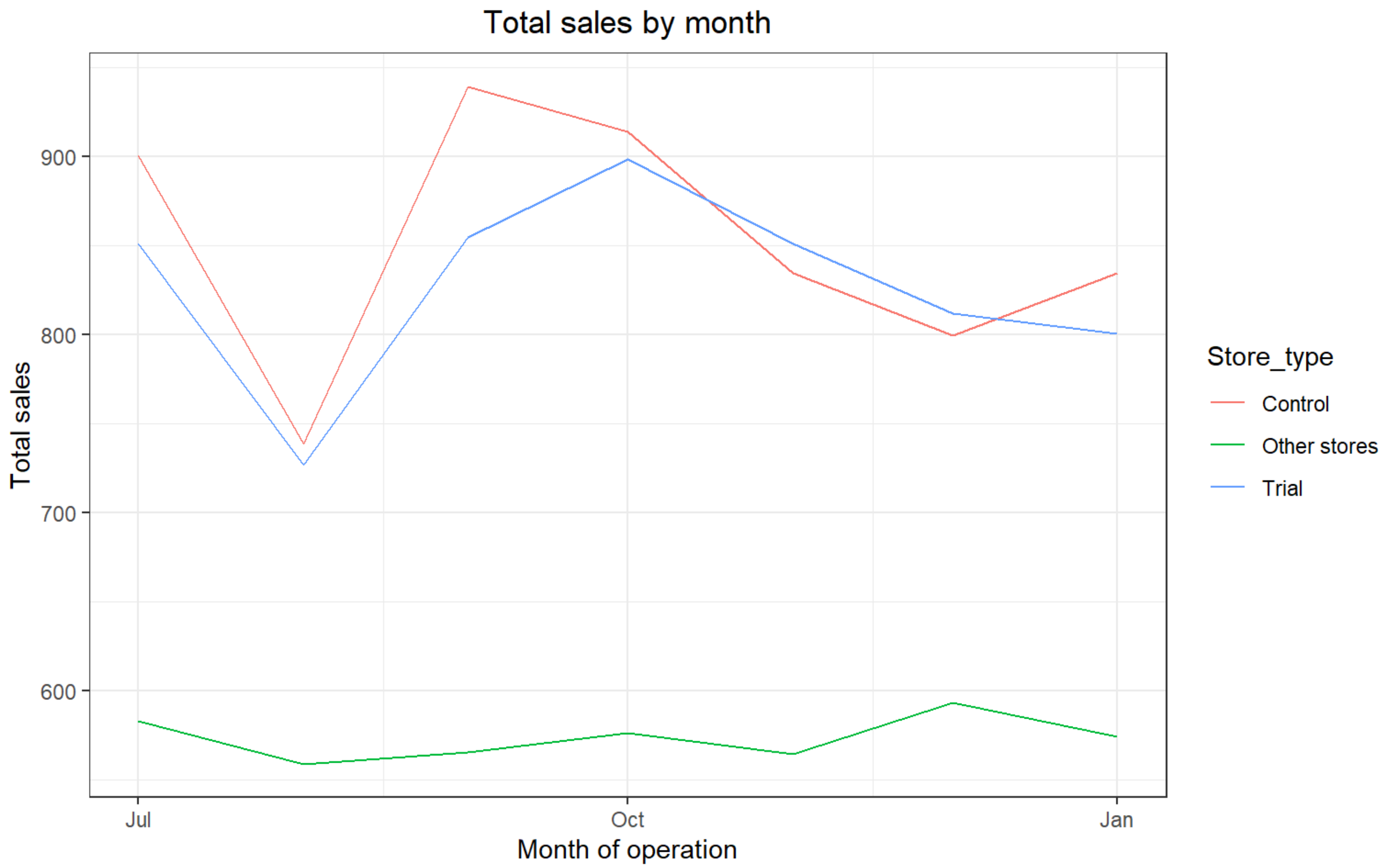
```
# Select the most appropriate control store for trial store 77 by finding the store with
the highest final score.
control_store <- score_Control[Store1 == trial_store, .(Control_Store = Store2,
                                                   Final_Score = finalControlScore)
                              ][order(-Final_Score)][1, Control_Store]
cat("Trial Store:", trial_store, " Control Store:", control_store, "\n")
## Trial Store: 86  Control Store: 155
```

Now that we have found a control store, let's check visually if the drivers are indeed similar in the period before the trial.

Check the visual similarity of the test and control store drivers

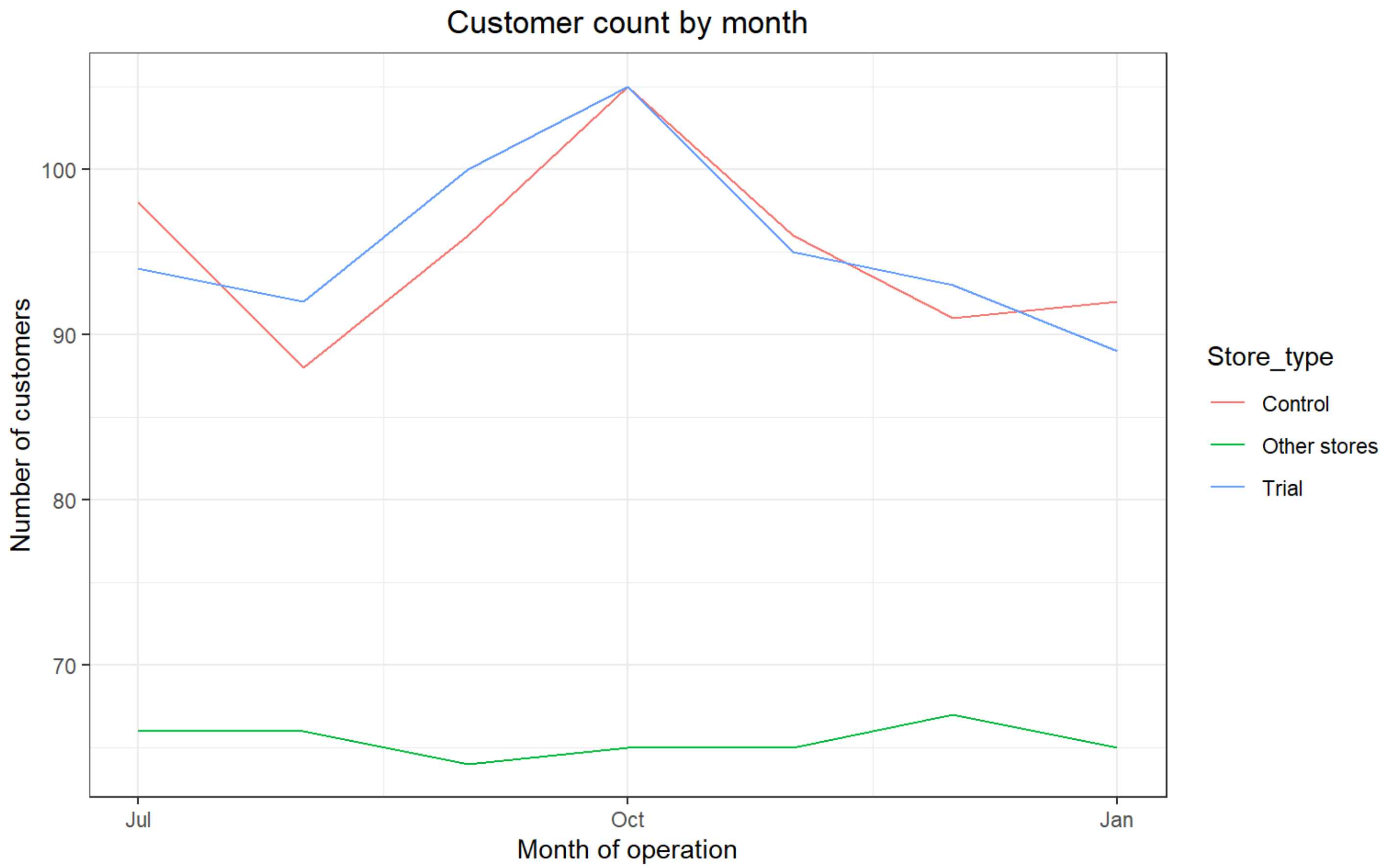
Let's look at total sales first.

```
# Visual checks on trends based on the drivers
measureOverTimeSales <- measureOverTime
measureOverTimeSales[, YEARMONTH := as.numeric(as.character(YEARMONTH))]  # Convert
YEARMONTH to numeric
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 < 201902 , ]

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



Next, the number of customers.

```
# Conduct visual checks on customer count trends by comparing the trial store to the
control store and other stores.
measureOverTimeCusts <- measureOverTime
measureOverTimeCusts[, YEARMONTH := as.integer(as.character(YEARMONTH))]  # Convert
YEARMONTH to integer
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store,
                                                   "Trial",
                                                 ifelse(STORE_NBR==control_store,
                                                   "Control", "Other stores"))
                                    ][, nCustomers := as.integer(mean(nCustomers)), by =
                                                   c("YEARMONTH", "Store_type")
                                    ][, TransactionMonth := as.Date(paste(YEARMONTH %/%
                                              100, YEARMONTH %% 100, 1, sep = "-"),
                                                       "%Y-%m-%d")
                                    ][YEARMONTH < 201902 , ]

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



Observations

- As can be seen from the above visuals, trial store 86 and control store 155 are indeed very close to each other in terms of performance during the pre-trial period. This is especially noticeable when compared to other stores.

Assessment of trial at store 86

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 controlling for any differences between the two stores outside of the trial period.

Assess the trial in terms of 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]
scaledControlSales[, controlSales, totSales]
##     totSales controlSales
##  1:   900.60     875.4277
##  2:   738.70     718.0529
##  3:   939.60     913.3376
##  4:   914.00     888.4531
```

```
##  5:   835.00     811.6612
##  6:   799.80     777.4451
##  7:   834.60     811.2724
##  8:   850.80     827.0196
##  9:   767.00     745.5619
## 10:   800.40     778.0283
## 11:   863.25     839.1216
## 12:   760.80     739.5352
```

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("STORE_NBR","YEARMONTH","totSales",
                                               "Store_type","controlSales")],
                        measureOverTimeSales[STORE_NBR == trial_store, c("STORE_NBR",
                                             "YEARMONTH","totSales","Store_type")],
                        by = c("YEARMONTH")
                        )[, percentageDiff := abs(controlSales - totSales.y) /
                                                 controlSales]

trialPercentageDiff <- percentageDiff[YEARMONTH %in% trialMonths]
trialPercentageDiff
##    YEARMONTH STORE_NBR.x totSales.x Store_type.x controlSales STORE_NBR.y
## 1:    201902         155      850.8      Control     827.0196          86
## 2:    201903         155      767.0      Control     745.5619          86
## 3:    201904         155      800.4      Control     778.0283          86
##    totSales.y Store_type.y percentageDiff
## 1:      872.8        Trial     0.05535587
## 2:      945.4        Trial     0.26803695
## 3:      804.0        Trial     0.03338141
```

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])
degreesOfFreedom <- numMonthsPreTrial - 1

# We will test with a null hypothesis of there being 0 difference between trial and
control stores.
# Calculate the t-values for the trial months.
# After that, find the 95th percentile of the t-distribution with the appropriate degrees
of freedom
# to check whether the hypothesis is statistically significant.

# Calculate the t-values for the trial months
trialPercentageDiff[, tValue := (as.numeric(trialPercentageDiff$percentageDiff) - 0) /
                                             stdDev]

# Find the 95th percentile of the t-distribution with the appropriate degrees of freedom
trialPercentageDiff[, tCritical := qt(0.95, df = degreesOfFreedom)]

# Check whether the t-values are statistically significant
```

```r
trialPercentageDiff[, isSignificant := tValue > tCritical]
trialPercentageDiff
##    YEARMONTH STORE_NBR.x totSales.x Store_type.x controlSales STORE_NBR.y
## 1:    201902         155      850.8      Control     827.0196          86
## 2:    201903         155      767.0      Control     745.5619          86
## 3:    201904         155      800.4      Control     778.0283          86
##    totSales.y Store_type.y percentageDiff    tValue tCritical isSignificant
## 1:      872.8        Trial     0.05535587  2.642804   1.94318          TRUE
## 2:      945.4        Trial     0.26803695 12.796638   1.94318          TRUE
## 3:      804.0        Trial     0.03338141  1.593697   1.94318         FALSE
```

Observations

1. In February 2019 the percentage difference is 5.54%, and the t-value is 2.64. It is statistically significant.

2. In March 2019 the percentage difference is 26.80%, and the t-value is 12.8. It is statistically significant.

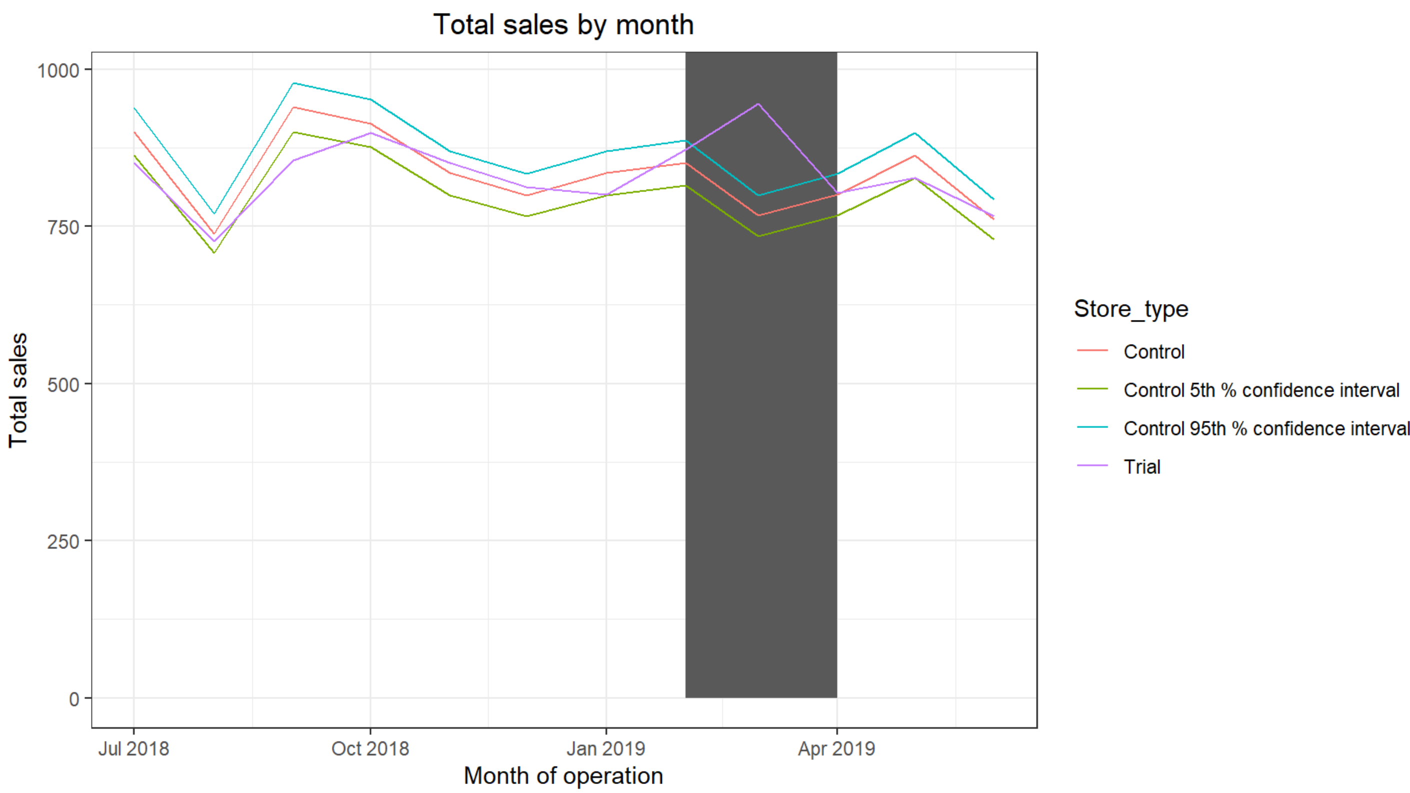
3. In April 2019 the percentage difference is 3.34%, and the t-value is 1.59. It is not statistically significant.

4. We can see that the t-value is larger than the 95th percentile value of the t-distribution for February and March, i.e., the increase in sales in the trial store is observed in the first two months of the trial period.

Let's create a more visual version of this by plotting the sales of the control store, the sales of the trial store, and the 5th and 95th percentile value of sales of the control store.

```r
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"]
```

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

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



Observations

- The results show that the trial in store 86 is not significantly different from its control store 155 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.

Assess the trial in terms of number of customers

```r
# Scale pre-trial control customer counts to match pre-trial trial store customer counts
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]
scaledControlCustomers[, controlCustomers, nCustomers]
##     nCustomers controlCustomers
##  1:         98         98.29429
```

```
##  2:         88         88.26426
##  3:         96         96.28829
##  4:         96         96.28829
##  5:        105        105.31532
##  6:         91         91.27327
##  7:         91         91.27327
##  8:         92         92.27628
##  9:         92         92.27628
## 10:         93         93.27928
## 11:        101        101.30330
## 12:         87         87.26126
```

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

```
# Calculate the percentage difference between scaled control and trial customer counts
percentageDiff <- merge(scaledControlCustomers[, c("STORE_NBR","YEARMONTH","nCustomers",
                                                   "Store_type","controlCustomers")],
                        measureOverTimeCusts[STORE_NBR == trial_store, c("STORE_NBR",
                                                   "YEARMONTH","nCustomers","Store_type")],
                        by = c("YEARMONTH")
                        )[, percentageDiff := abs(controlCustomers - nCustomers.y) /
                                                  controlCustomers]

trialPercentageDiff <- percentageDiff[YEARMONTH %in% trialMonths]
trialPercentageDiff
##    YEARMONTH STORE_NBR.x nCustomers.x Store_type.x controlCustomers STORE_NBR.y
## 1:    201902         155           92      Control         92.27628          86
## 2:    201903         155           91      Control         91.27327          86
## 3:    201904         155           93      Control         93.27928          86
##    nCustomers.y Store_type.y percentageDiff
## 1:          105        Trial     0.13788727
## 2:          108        Trial     0.18325985
## 3:           99        Trial     0.06132895
```

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])
degreesOfFreedom <- numMonthsPreTrial - 1

# We will test with a null hypothesis of there being 0 difference between trial and
control stores.
# Calculate the t-values for the trial months.
# After that, find the 95th percentile of the t-distribution with the appropriate degrees
of freedom
# to check whether the hypothesis is statistically significant.

# Calculate the t-values for the trial months
trialPercentageDiff[, tValue := (as.numeric(trialPercentageDiff$percentageDiff) - 0) /
                                            stdDev]

# Find the 95th percentile of the t-distribution with the appropriate degrees of freedom
trialPercentageDiff[, tCritical := qt(0.95, df = degreesOfFreedom)]
```

```r
# Check whether the t-values are statistically significant
trialPercentageDiff[, isSignificant := tValue > tCritical]
trialPercentageDiff
##    YEARMONTH STORE_NBR.x nCustomers.x Store_type.x controlCustomers STORE_NBR.y
## 1:    201902         155           92      Control         92.27628          86
## 2:    201903         155           91      Control         91.27327          86
## 3:    201904         155           93      Control         93.27928          86
##    nCustomers.y Store_type.y percentageDiff    tValue tCritical isSignificant
## 1:          105        Trial     0.13788727  8.605720   1.94318          TRUE
## 2:          108        Trial     0.18325985 11.437481   1.94318          TRUE
## 3:           99        Trial     0.06132895  3.827618   1.94318          TRUE
```

Observations

1. In February 2019 the percentage difference is 13.79%, and the t-value is 8.61. It is statistically significant.

2. In March 2019 the percentage difference is 18.33%, and the t-value is 11.44. It is statistically significant.

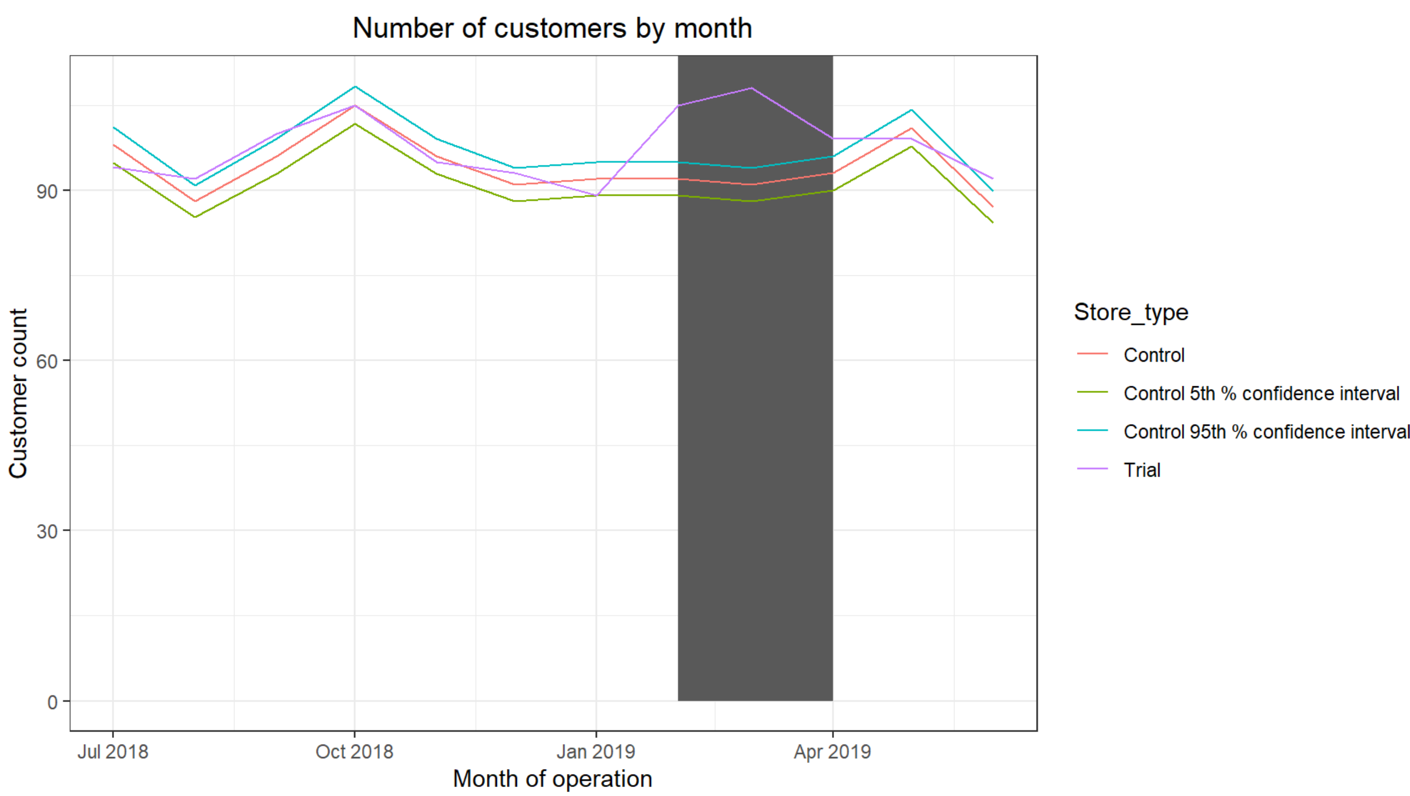
3. In April 2019 the percentage difference is 6.13%, and the t-value is 3.83. It is statistically significant.

4. We can observe that the t-value is much larger than the 95th percentile value of the t-distribution for all three months, i.e., the increase in customer counts in the trial store during the trial period is statistically greater than in the control store.

Let's create a more visual version of this by plotting the customer counts of the control store, the customer counts of the trial store, and the 5th and 95th percentile values of the control store.

```r
measureOverTimeCusts <- measureOverTime

# Trial and control store customer counts
# Create new variables `Store_type`, `nCusts` and `TransactionMonth` in the data table.
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store,
                                                              "Trial",
                                                         ifelse(STORE_NBR==control_store,
                                                                "Control",
                                                                "Other stores"))
                                 ][, nCusts := mean(nCustomers), 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
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 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 = "Customer count",
       title = "Number of customers by month")
```



Observations

- 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 may have resulted in lower prices, impacting the results.

Select control store for trial store 88

```
# Next, define the measure calculations to use during the analysis.
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),
```

```r
                       avgPricePerUnit = sum(TOT_SALES) / sum(PROD_QTY)),
                  by = .(STORE_NBR, YEARMONTH)][order(STORE_NBR, YEARMONTH)]

# Now 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, ]

# Get a number of months in the pre-trial period to use in the next calculations
numMonthsPreTrial <- preTrialMeasures[, uniqueN(YEARMONTH)]
# Use the created functions to calculate correlations against store 88 using total sales
and number of customers
trial_store <- 88
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales), trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers), trial_store)

# Now 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 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.

```r
# Create a combined score composed of correlation and magnitude,
# by first merging the correlations table with the magnitude table.

corr_weight <- 0.5

# By using (1 - corr_weight) for the weight of the magnitude score, we ensure that the
sum of the weights
# for both scores is equal to 1. Thus we allow for a balanced combination of both scores,
# and adjusting corr_weight allows us to easily control the balance based on our
preference or specific requirements.

score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1", "Store2"))
score_nSales[, scoreNSales := corr_weight * corr_measure + (1 - corr_weight) *
             mag_measure]

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

Now we have a score for each of the total number of sales and a number of customers.

Let's combine the two via a simple average.

```r
# Combine scores across the drivers by merging the sales scores and customer scores into
a single table
score_Control <- merge(score_nSales, score_nCustomers, by = c("Store1", "Store2"))

# Calculate the final control score using a simple average
```

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

##      Store1 Store2 corr_measure.x mag_measure.x  scoreNSales corr_measure.y
##   1:     88      1      0.8422323     0.1415119  0.491872135     0.42997723
##   2:     88      2     -0.2324942     0.1138731 -0.059310524    -0.54739963
##   3:     88      3     -0.4673303     0.8198073  0.176238530     0.43408024
##   4:     88      4     -0.5061296     0.9114412  0.202655807    -0.21677788
##   5:     88      5      0.3385254     0.6032565  0.470890978    -0.02653491
##  ---
## 254:     88    268     -0.2015731     0.1589548 -0.021309112     0.53863277
## 255:     88    269     -0.1013492     0.7131668  0.305908815    -0.06571521
## 256:     88    270     -0.6959380     0.7094149  0.006738432    -0.07469496
## 257:     88    271     -0.1609274     0.5990261  0.219049352    -0.11123054
## 258:     88    272     -0.6457516     0.2847024 -0.180524610    -0.13301096
##      mag_measure.y  scoreNCust finalControlScore
##   1:     0.3452338  0.38760550        0.43973882
##   2:     0.2839079 -0.13174588       -0.09552820
##   3:     0.8474894  0.64078483        0.40851168
##   4:     0.9349609  0.35909153        0.28087367
##   5:     0.7121839  0.34282450        0.40685774
##  ---
## 254:     0.3236297  0.43113124        0.20491106
## 255:     0.8338969  0.38409084        0.34499983
## 256:     0.8087794  0.36704223        0.18689033
## 257:     0.7062867  0.29752809        0.25828872
## 258:     0.3267499  0.09686946       -0.04182757
```

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 final score.
control_store <- score_Control[Store1 == trial_store, .(Control_Store = Store2,
                                                        Final_Score = finalControlScore)

                              ][order(-Final_Score)][1, Control_Store]
cat("Trial Store:", trial_store, " Control Store:", control_store, "\n")

## Trial Store: 88  Control Store: 237
```

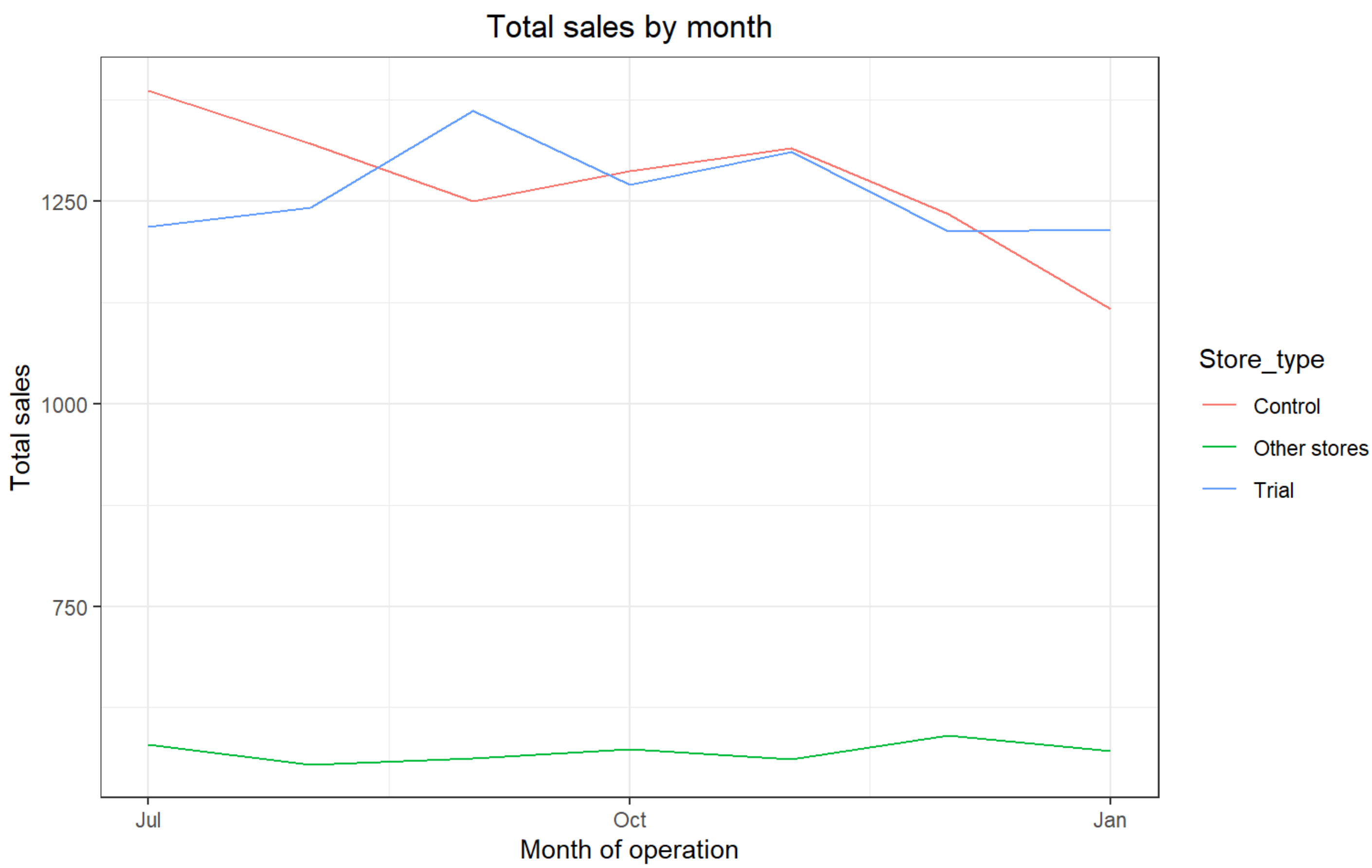
Now that we have found a control store, let's check visually if the drivers are indeed similar in the period before the trial.

Check the visual similarity of the test and control store drivers

Let's look at total sales first.

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

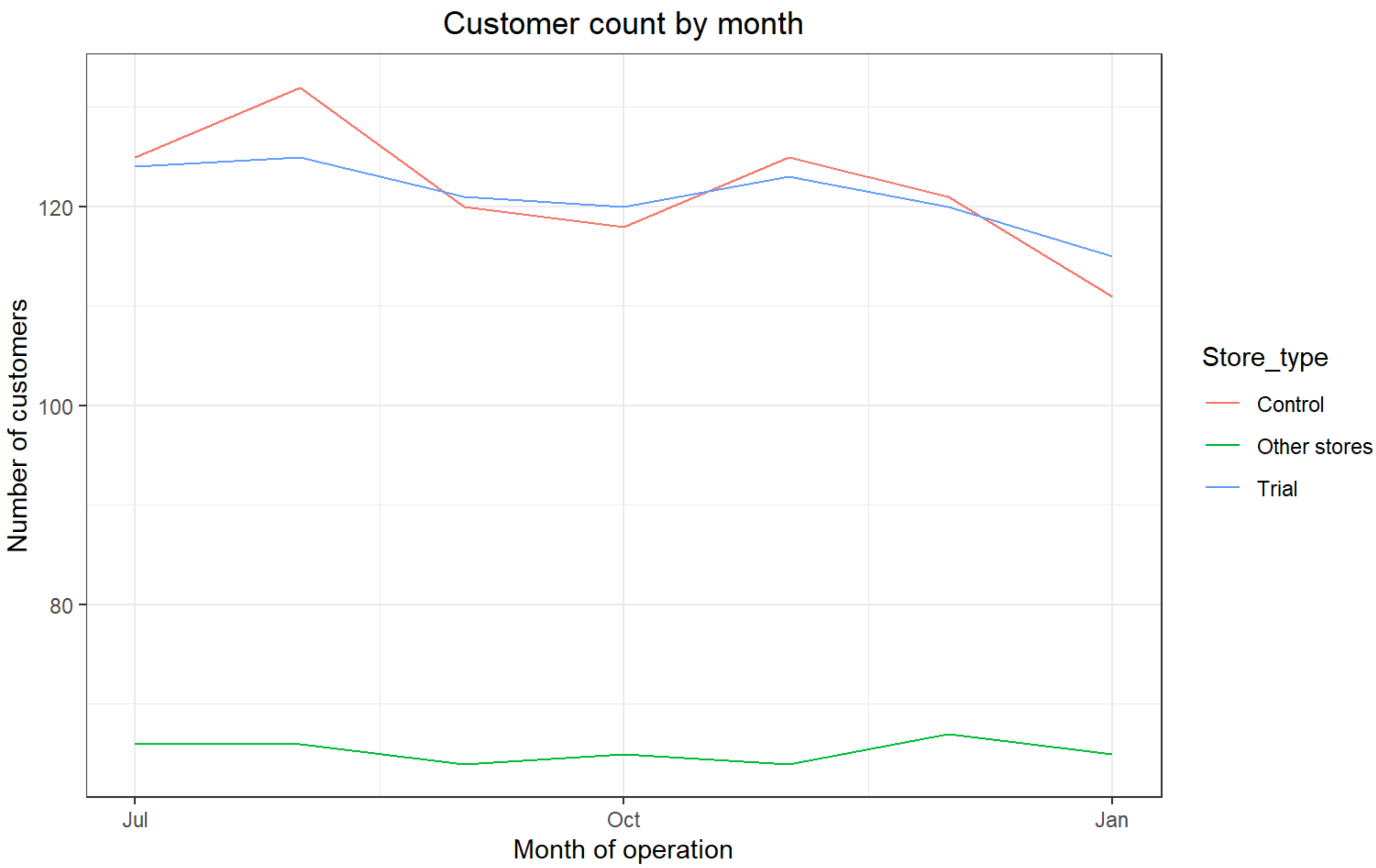
```r
                                                                        "Store_type")
                                    ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
                                                          YEARMONTH %% 100, 1, sep = "-"),
                                                                  "%Y-%m-%d")
                                    ][YEARMONTH < 201902 , ]

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



Next, the number of customers.

```r
# Conduct visual checks on customer count trends by comparing the trial store to the
control store and other stores.
measureOverTimeCusts <- measureOverTime
measureOverTimeCusts[, YEARMONTH := as.integer(as.character(YEARMONTH))]  # Convert
YEARMONTH to integer
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store,
                                                              "Trial",
                                                          ifelse(STORE_NBR==control_store,
                                                              "Control", "Other stores"))
                                      ][, nCustomers := as.integer(mean(nCustomers)), by =
                                        c("YEARMONTH", "Store_type")
                                      ][, TransactionMonth := as.Date(paste(YEARMONTH %/%
                                        100, YEARMONTH %% 100, 1, sep = "-"),
                                                                  "%Y-%m-%d")
                                      ][YEARMONTH < 201902 , ]
```

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



Observations

- As can be seen from the above visuals, trial store 88 and control store 237 are indeed very close to each other in terms of performance during the pre-trial period. This is especially noticeable when compared to other stores.

Assessment of trial at store 88

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 controlling for any differences between the two stores outside of the trial period.

Assess the trial in terms of sales

```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]
scaledControlSales[, controlSales, totSales]
##     totSales controlSales
##  1:   1387.2     1374.394
##  2:   1321.9     1309.697
##  3:   1250.8     1239.253
##  4:   1287.1     1275.218
##  5:   1316.0     1303.851
##  6:   1234.4     1223.005
##  7:   1117.7     1107.382
##  8:   1313.0     1300.879
##  9:   1177.6     1166.729
## 10:   1153.6     1142.951
## 11:   1127.9     1117.488
## 12:   1143.4     1132.845
```

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("STORE_NBR","YEARMONTH","totSales",
                                           "Store_type","controlSales")],
                        measureOverTimeSales[STORE_NBR == trial_store, c("STORE_NBR",
                                             "YEARMONTH","totSales","Store_type")],
                        by = c("YEARMONTH")
                        )[, percentageDiff := abs(controlSales - totSales.y) /
                                                  controlSales]

trialPercentageDiff <- percentageDiff[YEARMONTH %in% trialMonths]
trialPercentageDiff
##    YEARMONTH STORE_NBR.x totSales.x Store_type.x controlSales STORE_NBR.y
## 1:    201902         237     1313.0      Control     1300.879          88
## 2:    201903         237     1177.6      Control     1166.729          88
## 3:    201904         237     1153.6      Control     1142.951          88
##    totSales.y Store_type.y percentageDiff
## 1:     1339.6        Trial     0.02976526
## 2:     1467.0        Trial     0.25736144
## 3:     1317.0        Trial     0.15228086
```

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])
degreesOfFreedom <- numMonthsPreTrial - 1

# We will test with a null hypothesis of there being 0 difference between trial and
control stores.
# Calculate the t-values for the trial months.
# After that, find the 95th percentile of the t-distribution with the appropriate degrees
of freedom
# to check whether the hypothesis is statistically significant.
```

```r
# Calculate the t-values for the trial months
trialPercentageDiff[, tValue := (as.numeric(trialPercentageDiff$percentageDiff) - 0) /
                                    stdDev]

# Find the 95th percentile of the t-distribution with the appropriate degrees of freedom
trialPercentageDiff[, tCritical := qt(0.95, df = degreesOfFreedom)]

# Check whether the t-values are statistically significant
trialPercentageDiff[, isSignificant := tValue > tCritical]
trialPercentageDiff
##    YEARMONTH STORE_NBR.x totSales.x Store_type.x controlSales STORE_NBR.y
## 1:    201902         237     1313.0      Control     1300.879          88
## 2:    201903         237     1177.6      Control     1166.729          88
## 3:    201904         237     1153.6      Control     1142.951          88
##    totSales.y Store_type.y percentageDiff    tValue tCritical isSignificant
## 1:     1339.6        Trial     0.02976526 0.6064868   1.94318         FALSE
## 2:     1467.0        Trial     0.25736144 5.2439100   1.94318          TRUE
## 3:     1317.0        Trial     0.15228086 3.1028236   1.94318          TRUE
```

Observations

1. In February 2019 the percentage difference is 2.98%, and the t-value is 0.61. It is not statistically significant.

2. In March 2019 the percentage difference is 25.74%, and the t-value is 5.24. It is statistically significant.

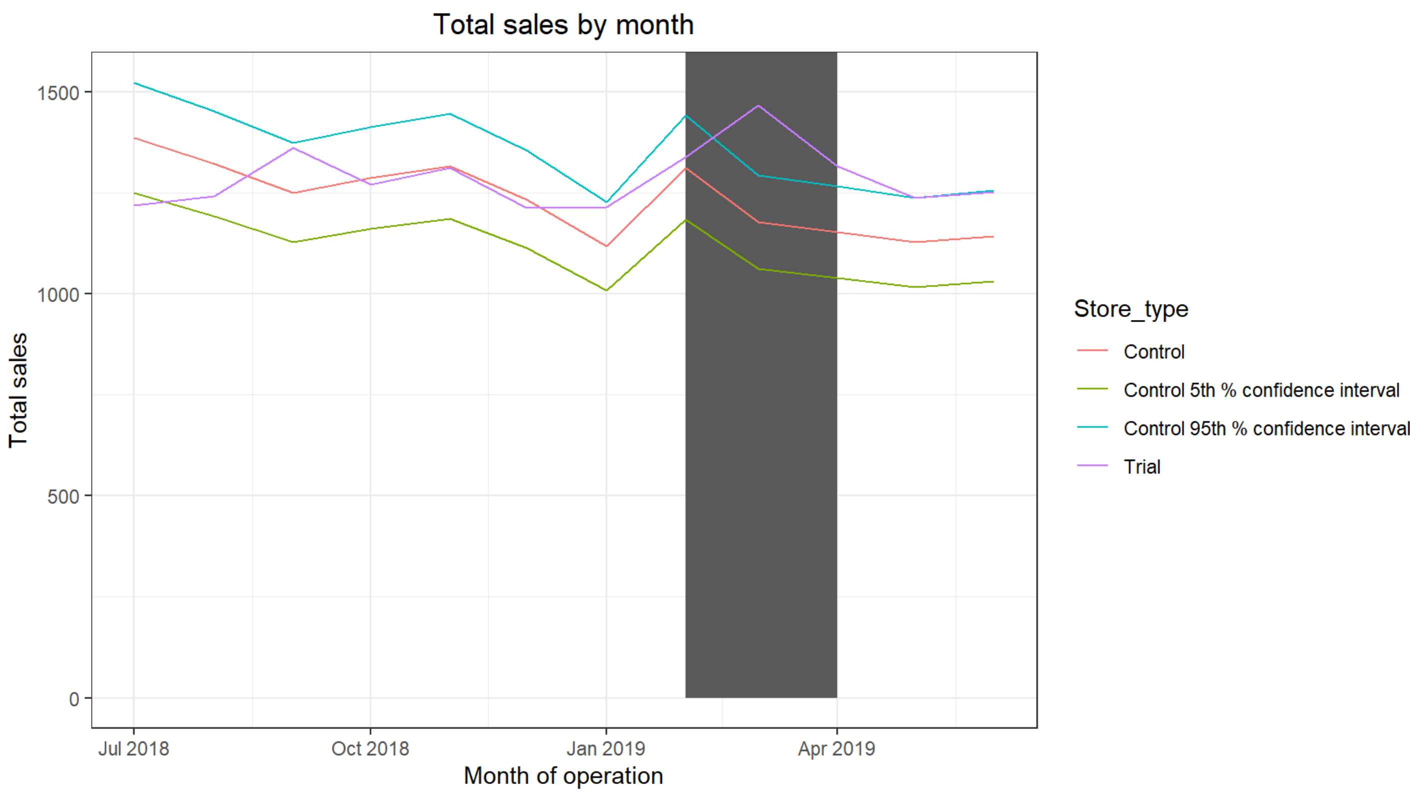
3. In April 2019 the percentage difference is 15.23%, and the t-value is 3.10. It is statistically significant.

4. We can observe that the t-value is 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 store, and the 5th and 95th percentile value of sales of the control store.

```r
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)

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



Observations

- The results show that the trial in store 88 is significantly different from 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.

Assess the trial in terms of number of customers

```
# Scale pre-trial control customer counts to match pre-trial trial store customer counts
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]
scaledControlCustomers[, controlCustomers, nCustomers]
##     nCustomers controlCustomers
##  1:        125         124.4131
##  2:        125         124.4131
##  3:        132         131.3803
##  4:        120         119.4366
##  5:        118         117.4460
##  6:        118         117.4460
##  7:        121         120.4319
##  8:        111         110.4789
##  9:        119         118.4413
## 10:        116         115.4554
## 11:        116         115.4554
## 12:        122         121.4272
```

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

```
# Calculate the percentage difference between scaled control and trial customer counts
percentageDiff <- merge(scaledControlCustomers[, c("STORE_NBR","YEARMONTH","nCustomers",
                                                   "Store_type","controlCustomers")],
                        measureOverTimeCusts[STORE_NBR == trial_store, c("STORE_NBR",
                                                 "YEARMONTH","nCustomers","Store_type")],
                        by = c("YEARMONTH")
                        )[, percentageDiff := abs(controlCustomers - nCustomers.y) /
                                                    controlCustomers]

trialPercentageDiff <- percentageDiff[YEARMONTH %in% trialMonths]
trialPercentageDiff
##    YEARMONTH STORE_NBR.x nCustomers.x Store_type.x controlCustomers STORE_NBR.y
## 1:    201902         237          119      Control         118.4413          88
## 2:    201903         237          116      Control         115.4554          88
## 3:    201904         237          116      Control         115.4554          88
##    nCustomers.y Store_type.y percentageDiff
## 1:          122        Trial     0.03004598
## 2:          133        Trial     0.15195999
## 3:          119        Trial     0.03070104
```

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])
degreesOfFreedom <- numMonthsPreTrial - 1

# We will test with a null hypothesis of there being 0 difference between trial and
control stores.
# Calculate the t-values for the trial months.
# After that, find the 95th percentile of the t-distribution with the appropriate degrees
```

```
of freedom
# to check whether the hypothesis is statistically significant.

# Calculate the t-values for the trial months
trialPercentageDiff[, tValue := (as.numeric(trialPercentageDiff$percentageDiff) - 0) /
                                   stdDev]

# Find the 95th percentile of the t-distribution with the appropriate degrees of freedom
trialPercentageDiff[, tCritical := qt(0.95, df = degreesOfFreedom)]

# Check whether the t-values are statistically significant
trialPercentageDiff[, isSignificant := tValue > tCritical]
trialPercentageDiff
##    YEARMONTH STORE_NBR.x nCustomers.x Store_type.x controlCustomers STORE_NBR.y
## 1:    201902         237          119      Control         118.4413          88
## 2:    201903         237          116      Control         115.4554          88
## 3:    201904         237          116      Control         115.4554          88
##    nCustomers.y Store_type.y percentageDiff   tValue tCritical isSignificant
## 1:          122        Trial     0.03004598 1.677105   1.94318         FALSE
## 2:          133        Trial     0.15195999 8.482095   1.94318          TRUE
## 3:          119        Trial     0.03070104 1.713669   1.94318         FALSE
```

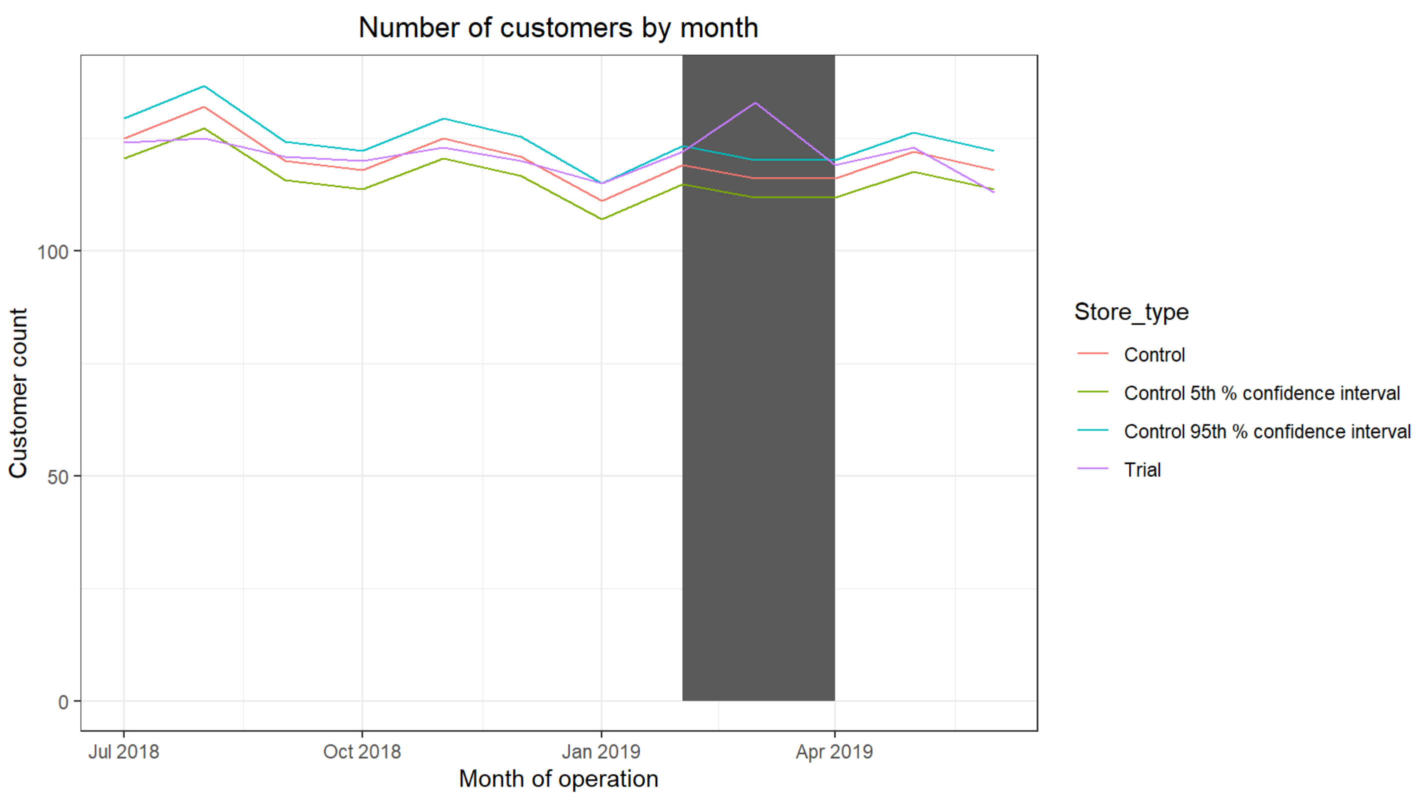
Observations

1. In February 2019 the percentage difference is 3%, and the t-value is 1.68. It is not statistically significant.
2. In March 2019 the percentage difference is 15.196%, and the t-value is 8.48. It is statistically significant.
3. In April 2019 the percentage difference is 3.07%, and the t-value is 1.71. It is not statistically significant.
4. We can see that the t-value is larger than the 95th percentile value of the t-distribution only for March, i.e., the increase in customer counts in the trial store is observed in only one month of the trial period.

Let's create a more visual version of this by plotting the customer counts of the control store, the customer counts of the trial store, and the 5th and 95th percentile values of the control store.

```
measureOverTimeCusts <- measureOverTime

# Trial and control store customer counts
# Create new variables `Store_type`, `nCusts` and `TransactionMonth` in the data table.
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store,
                                                           "Trial",
                                                      ifelse(STORE_NBR==control_store,
                                                             "Control",
                                                             "Other stores"))
                                       ][, nCusts := mean(nCustomers), 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
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 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 = "Customer count",
       title = "Number of customers by month")
```



Observations

- The results show that the trial in store 88 is not significantly different from its control store 237 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.

Conclusions

- We’ve found control stores 233, 155, and 237 for trial stores 77, 86, and 88 respectively.
- The results for trial store 77 during the trial period show a consistent pattern of increase in chip sales and customer counts compared to the control store and a significant difference in at least two of the three trial months.
- Trial store 86 shows a significant increase in customer counts compared to the control store throughout the trial period. However, sales growth is only visible in March, that is, in the middle of the trial. So, we need to clarify with the client if the implementation of the trial was different in trial store 86.
- Trial store 88 shows a significant increase in chip sales and customer counts compared to the control store by the middle of the trial period. However, by the end of the trial, both sales and customer counts declined significantly and this can hardly indicate the success of the trial.
- Further analysis and investigation are recommended, especially for trial stores 86 and 88, to understand the factors contributing to the varying levels of success and to evaluate the impact of any special deals or pricing strategies during the trial period.
- Now that we have finished our analysis, we can prepare our presentation to the Category Manager.