Task 2: Trial Store Layout Analysis, Feb-Mar 2019

Ed Garcia

8/17/2021

There have been store layout changes to stores 77, 86, and 88 during Feb-Apr 2019.

Are there any significant improvements on total sales or number of customers for chips at these stores?

set options for R markdown knitting

```
knitr::opts_chunk$set(warning = FALSE, message = FALSE)
```

LOAD REQUIRED LIBRARIES

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

IMPORT SOURCE CSV FILE INTO R FROM TASK 1

```
data <- fread(paste0("C:/Users/garci/OneDrive/Desktop/Data Analysis Education/Forage Virtual Internship
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 2019-03-08         1      4      106
## 4:           1003 2019-03-07         1      3       52
## 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: Natural ChipCo      Hony Soy Chckn175g        1       3.0       175
## 4:  Grain Waves Sour    Cream&Chives 210G        1       3.6       210
## 5:         WW Original Stacked Chips 160g        1       1.9       160
## 6:                     Cheetos Puffs 165g        1       2.8       165
##                  BRANDS             LIFESTAGE PREMIUM_CUSTOMER
## 1: Natural Chip Company  YOUNG SINGLES/COUPLES          Premium
## 2:        Red Rock Deli  YOUNG SINGLES/COUPLES       Mainstream
## 3: Natural Chip Company         YOUNG FAMILIES           Budget
## 4:          Grain Waves         YOUNG FAMILIES           Budget
## 5:           Woolworths  OLDER SINGLES/COUPLES       Mainstream
## 6:              Cheetos MIDAGE SINGLES/COUPLES       Mainstream
##                      CUSTOMER_SEGMENT UNIT_PRICE
## 1:     Premium - YOUNG SINGLES/COUPLES        3.0
```

```
## 2:  Mainstream - YOUNG SINGLES/COUPLES        2.7
## 3:              Budget - YOUNG FAMILIES        3.0
## 4:              Budget - YOUNG FAMILIES        3.6
## 5:  Mainstream - OLDER SINGLES/COUPLES        1.9
## 6: Mainstream - MIDAGE SINGLES/COUPLES        2.8
```

SELECT THE RANGE OF POSSIBLE CONTROL STORES

The client has selected 3 trial stores (store numbers 77, 86, 88), and the next task is to match these trial stores to control stores with similar attributes prior to the trial period:

- Stores must have been operational for the entire pre-trial observation period (before Feb 2019)
- Similar monthly overall sales revenue
- Similar monthly number of customers
- Similar monthly number of transactions per customer

Calculate these measures over time for each store

Add a new month ID column to the data with the format yyyymm.

```
data[, YEARMONTH := year(DATE)*100 + month(DATE)]
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 2019-03-08         1      4      106
## 4:           1003 2019-03-07         1      3       52
## 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: Natural ChipCo      Hony Soy Chckn175g        1       3.0       175
## 4:  Grain Waves Sour    Cream&Chives 210G        1       3.6       210
## 5:         WW Original Stacked Chips 160g        1       1.9       160
## 6:                     Cheetos Puffs 165g        1       2.8       165
##                   BRANDS             LIFESTAGE PREMIUM_CUSTOMER
## 1: Natural Chip Company  YOUNG SINGLES/COUPLES          Premium
## 2:        Red Rock Deli  YOUNG SINGLES/COUPLES       Mainstream
## 3: Natural Chip Company         YOUNG FAMILIES           Budget
## 4:          Grain Waves         YOUNG FAMILIES           Budget
## 5:           Woolworths  OLDER SINGLES/COUPLES       Mainstream
## 6:              Cheetos MIDAGE SINGLES/COUPLES       Mainstream
##                      CUSTOMER_SEGMENT UNIT_PRICE YEARMONTH
## 1:     Premium - YOUNG SINGLES/COUPLES        3.0    201810
## 2:  Mainstream - YOUNG SINGLES/COUPLES        2.7    201809
## 3:              Budget - YOUNG FAMILIES        3.0    201903
## 4:              Budget - YOUNG FAMILIES        3.6    201903
## 5:  Mainstream - OLDER SINGLES/COUPLES        1.9    201811
## 6: Mainstream - MIDAGE SINGLES/COUPLES        2.8    201812
```

Define the measure calculations to be used during the analysis.

For each store and month, calculate the following in a single data frame:

- total sales
- number of customers
- transactions per customer
- chips per transaction
- average price per unit

```
measureOverTime <- data[, .(totSales = sum(TOT_SALES),
                            nCustomers = uniqueN(LYLTY_CARD_NBR),
                            nTxnPerCust = uniqueN(TXN_ID)/uniqueN(LYLTY_CARD_NBR),
                            nChipsPerTxn = sum(PROD_QTY)/uniqueN(TXN_ID),
                            avgPricePerUnit = sum(TOT_SALES)/sum(PROD_QTY)),
                        by = c("STORE_NBR", "YEARMONTH")][order(STORE_NBR, YEARMONTH)]
head(measureOverTime)
```

```
##    STORE_NBR YEARMONTH totSales nCustomers nTxnPerCust nChipsPerTxn
## 1:         1    201807    191.6         48    1.041667     1.180000
## 2:         1    201808    168.4         41    1.000000     1.268293
## 3:         1    201809    268.1         57    1.035088     1.203390
## 4:         1    201810    178.0         40    1.025000     1.268293
## 5:         1    201811    187.5         45    1.022222     1.217391
## 6:         1    201812    160.6         37    1.081081     1.200000
##    avgPricePerUnit
## 1:        3.247458
## 2:        3.238462
## 3:        3.776056
## 4:        3.423077
## 5:        3.348214
## 6:        3.345833
```

Find the stores that only contain full observations during the pre-trial period

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

Verify that the number of unique stores have been filtered.

```
uniqueN(measureOverTime$STORE_NBR)
```

```
## [1] 271
```

```
uniqueN(preTrialMeasures$STORE_NBR)
```

```
## [1] 259
```

The number of unique stores has decreased. Which ones have been filtered?

```
unique(preTrialMeasures$STORE_NBR)
```

```
##   [1]   1   2   3   4   5   6   7   8   9  10  12  13  14  15  16  17  18  19
##  [19]  20  21  22  23  24  25  26  27  28  29  30  32  33  34  35  36  37  38
##  [37]  39  40  41  42  43  45  46  47  48  49  50  51  52  53  54  55  56  57
##  [55]  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75
##  [73]  77  78  79  80  81  82  83  84  86  87  88  89  90  91  93  94  95  96
##  [91]  97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114
## [109] 115 116 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133
```

```
## [127] 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151
## [145] 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169
## [163] 170 171 172 173 174 175 176 178 179 180 181 182 183 184 185 186 187 188
## [181] 189 190 191 192 194 195 196 197 198 199 200 201 202 203 204 205 207 208
## [199] 209 210 212 213 214 215 216 217 219 220 221 222 223 224 225 226 227 228
## [217] 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246
## [235] 247 248 249 250 251 253 254 255 256 257 258 259 260 261 262 263 264 265
## [253] 266 267 268 269 270 271 272
```

Store 11, 31, and ten others have been filtered out for not containing full observations during the months in the pre-trial period.

Rank how similar each potential control store is to the trial store.

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.

Define:

- inputTable as a metric table with potential comparison stores
- metricCol as the store metric used to calculate correlation on
- storeComparison as the store number of the trial store

```
calculateCorrelation <- function(inputTable, metricCol, storeComparison)
{ calcCorrTable = data.table(Store1 = numeric(),
                             Store2 = numeric(),
                             corr_measure = numeric())
  storeNumbers <- unique(inputTable[, STORE_NBR])
    for (i in storeNumbers) {
      calculatedMeasure = data.table("Store1" = storeComparison, "Store2" = i,
                                     "corr_measure" =
                                       cor(inputTable[STORE_NBR ==
                                                        storeComparison,
                                                      eval(metricCol)],
                                           inputTable[STORE_NBR == i, eval(metricCol)]))
      calcCorrTable <- rbind(calcCorrTable, calculatedMeasure)
                             }
  return(calcCorrTable)
}
```

Calculate a standardised metric based on the absolute difference between the trial store's performance and each control store's performance.

Create a function to calculate a standardised magnitude distance for a measure, looping through each control store.

Use the same arguments for the previous function.

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

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

SELECT A CONTROL STORE TO MATCH WITH TRIAL STORE NUMBER 77

Use the two functions above to find control stores based on their similarity to trial store 77 in terms of their monthly total sales ($) and monthly number of customers.

Assign the value 77 to trial_store for use in the functions.

```
trial_store <- 77
```

Calculate correlation against store 77 using total sales.

```
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales), trial_store)
corr_nSales[order(-corr_measure)]
```

```
##      Store1 Store2 corr_measure
##   1:     77     77    1.0000000
##   2:     77    233    0.9653030
##   3:     77     50    0.9015274
##   4:     77     71    0.8576903
##   5:     77    119    0.8392625
##  ---
## 255:     77      9   -0.7997364
## 256:     77     75   -0.8249615
## 257:     77    242   -0.8277850
## 258:     77    158   -0.8339073
## 259:     77    186   -0.8600030
```

Calculate correlation against store 77 using number of customers.

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

```
##      Store1 Store2 corr_measure
##   1:     77     77    1.0000000
##   2:     77    233    0.9509684
##   3:     77    119    0.9381303
##   4:     77    254    0.9242477
##   5:     77    113    0.8935020
##  ---
## 255:     77    242   -0.7691330
## 256:     77     54   -0.7793361
```

```
## 257:     77      9   -0.7840135
## 258:     77    147   -0.8237194
## 259:     77    208   -0.8255558
```

Calculate magnitude against store 77 using total sales.

```
magnitude_nSales <-
  calculateMagnitudeDistance(preTrialMeasures, quote(totSales), trial_store)
magnitude_nSales[order(-mag_measure)]
```

```
##      Store1 Store2 mag_measure
##   1:     77     77  1.00000000
##   2:     77    233  0.98505378
##   3:     77    188  0.97844141
##   4:     77    205  0.97621371
##   5:     77     50  0.97574751
##  ---
## 255:     77      4  0.18110399
## 256:     77    165  0.16216626
## 257:     77     88  0.15103405
## 258:     77    237  0.14175541
## 259:     77    226  0.06175832
```

Calculate magnitude against store 77 using number of customers.

```
magnitude_nCustomers <-
  calculateMagnitudeDistance(preTrialMeasures, quote(nCustomers), trial_store)
magnitude_nCustomers[order(-mag_measure)]
```

```
##      Store1 Store2 mag_measure
##   1:     77     77  1.00000000
##   2:     77    233  0.97703630
##   3:     77     41  0.97167608
##   4:     77    115  0.95976626
##   5:     77     17  0.95919316
##  ---
## 255:     77    165  0.18115352
## 256:     77     58  0.17415359
## 257:     77     88  0.14761690
## 258:     77    237  0.14039092
## 259:     77    226  0.05073274
```

Create a combined score composed of correlation and magnitude in order to determine the final control score measurement.

Assign a weighted average of 0.5 to use on the scores.

```
corr_weight <- 0.5
```

Merge the sales correlation table with the sales magnitude table.

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

```
##      Store1 Store2 corr_measure mag_measure scoreNSales
##   1:     77     77    1.0000000   1.0000000  1.00000000
```

```
##   2:     77    233    0.9653030   0.9850538  0.97517838
##   3:     77     50    0.9015274   0.9757475  0.93863747
##   4:     77     41    0.7701958   0.9590673  0.86463157
##   5:     77    167    0.6693229   0.9520936  0.81070827
##  ---
## 255:     77    172   -0.6801420   0.5264928 -0.07682461
## 256:     77    201   -0.4778564   0.2810825 -0.09838698
## 257:     77     88   -0.3838376   0.1510341 -0.11640176
## 258:     77    138   -0.7553321   0.5109711 -0.12218048
## 259:     77     75   -0.8249615   0.3166568 -0.25415238
```

Merge the customers correlation table with the customers magnitude table.

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

```
##      Store1 Store2 corr_measure mag_measure scoreNCust
##   1:     77     77    1.0000000   1.0000000  1.0000000
##   2:     77    233    0.9509684   0.9770363  0.9640024
##   3:     77    254    0.9242477   0.9242624  0.9242550
##   4:     77     27    0.8469826   0.9562141  0.9015983
##   5:     77     41    0.7965235   0.9716761  0.8840998
##  ---
## 255:     77     75   -0.5712283   0.3436918 -0.1137682
## 256:     77    208   -0.8255558   0.5633020 -0.1311269
## 257:     77    138   -0.7059030   0.4226633 -0.1416198
## 258:     77    147   -0.8237194   0.5186571 -0.1525311
## 259:     77    227   -0.7472179   0.4172868 -0.1649656
```

Merge the sales/customer scores and their related correlation/magnitude measures. Determine the final control score by:

1. multiplying the sales score by the weight average 0.5,
2. multiplying the customers score by the weight average 0.5, and
3. adding both products together

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

```
##      Store1 Store2 corr_measure.x mag_measure.x scoreNSales corr_measure.y
##   1:     77     77      1.0000000     1.0000000  1.00000000      1.0000000
##   2:     77    233      0.9653030     0.9850538  0.97517838      0.9509684
##   3:     77     41      0.7701958     0.9590673  0.86463157      0.7965235
##   4:     77     50      0.9015274     0.9757475  0.93863747      0.6840326
##   5:     77    254      0.6880164     0.9183432  0.80317977      0.9242477
##  ---
## 255:     77    172     -0.6801420     0.5264928 -0.07682461     -0.6024519
## 256:     77    227     -0.5550681     0.4977738 -0.02864713     -0.7472179
## 257:     77    147     -0.6718388     0.5691491 -0.05134488     -0.8237194
## 258:     77    138     -0.7553321     0.5109711 -0.12218048     -0.7059030
## 259:     77     75     -0.8249615     0.3166568 -0.25415238     -0.5712283
##      mag_measure.y  scoreNCust finalControlScore
```

```
##   1:     1.0000000  1.00000000        1.00000000
##   2:     0.9770363  0.96400236        0.96959037
##   3:     0.9716761  0.88409980        0.87436568
##   4:     0.9348799  0.80945626        0.87404686
##   5:     0.9242624  0.92425503        0.86371740
##  ---
## 255:     0.4851636 -0.05864413       -0.06773437
## 256:     0.4172868 -0.16496557       -0.09680635
## 257:     0.5186571 -0.15253115       -0.10193801
## 258:     0.4226633 -0.14161984       -0.13190016
## 259:     0.3436918 -0.11376822       -0.18396030
```

Select the most appropriate control store for trial store 77 based on the highest matching store (choose the 2nd highest store since the control store can't be the trial store itself)

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

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

The control store for trial store 77 is store 233.

Looking back at the results of previous scores and measurements, store 233 has placed 2nd highest in each result. This confirms the result of the control store test. Visualizations will provide further confirmation.

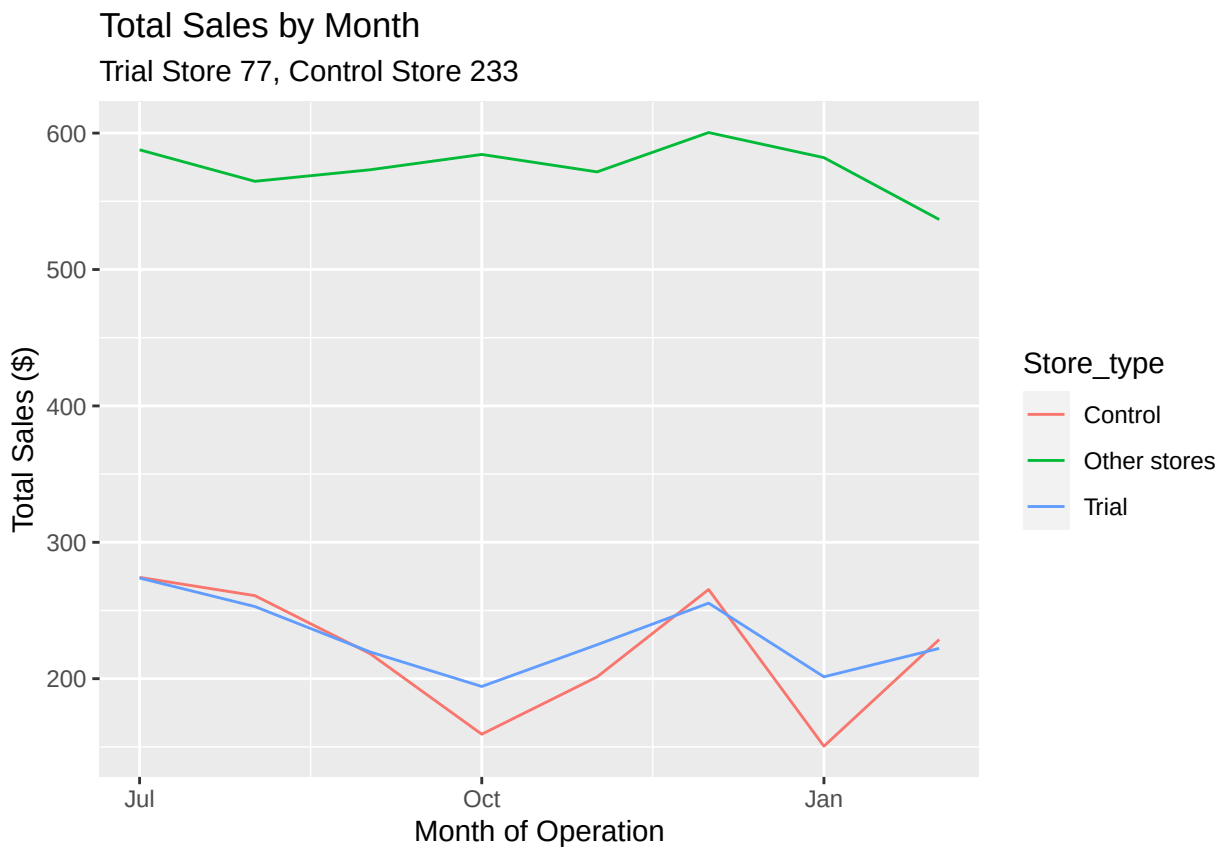
Visualize the movement of sales and number of customers against the trial store, control store, and all other stores.

Prepare a sales data frame for plotting purposes. Demarcate the trial store, control store, and all other stores:

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

Plot the total sales by month for the pre-trial period.

```
ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line() +
  labs(x = "Month of Operation", y = "Total Sales ($)",
       title = "Total Sales by Month", subtitle = "Trial Store 77, Control Store 233")
```



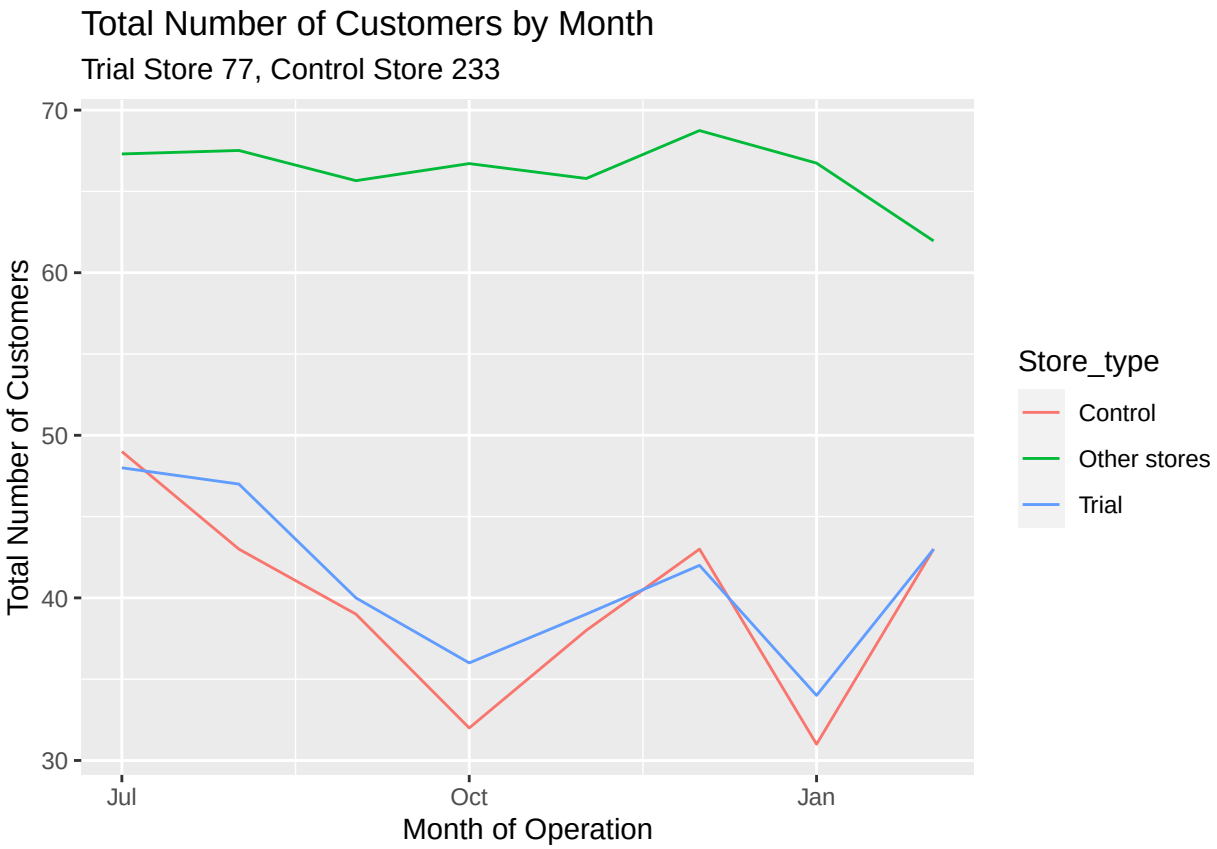
The plot shows that total sales are indeed similar between the trial and control stores.

Prepare a customer data frame for plotting purposes. Demarcate the trial store, control store, and all other stores:

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

Plot the total number of customers by month for the pre-trial period.

```
ggplot(pastCustomers, aes(TransactionMonth, numberCustomers, color = Store_type)) +
  geom_line() +
  labs(x = "Month of Operation", y = "Total Number of Customers",
       title = "Total Number of Customers by Month",
       subtitle = "Trial Store 77, Control Store 233")
```



ASSESSMENT OF TRIAL STORE 77: TOTAL SALES AND NUMBER OF CUSTOMERS

Assessment 1: Has there been an uplift in overall chip sales?

Compute scaling factor for sales by dividing the sum of pre-trial trial store sales by the sum of pre-trial control store sales.

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

```
## [1] 1.060327
```

Apply the scaling factor to the control store in a new data frame by multiplying the total control store sales by the scaling factor.

```
scaledControlSales <-
  measureOverTime[STORE_NBR == control_store, ][, controlSales :=
                                                  totSales * scalingFactorForControlSales]
head(scaledControlSales)
```

```
##    STORE_NBR YEARMONTH totSales nCustomers nTxnPerCust nChipsPerTxn
## 1:       233    201807    274.3         49    1.020408     1.600000
## 2:       233    201808    260.9         43    1.046512     1.600000
## 3:       233    201809    218.3         39    1.076923     1.595238
## 4:       233    201810    159.3         32    1.000000     1.500000
```

```
## 5:       233    201811    201.3         38    1.026316     1.512821
## 6:       233    201812    265.4         43    1.046512     1.555556
##    avgPricePerUnit Store_type TransactionMonth numberCustomers controlSales
## 1:        3.428750    Control       2018-07-01              49     290.8476
## 2:        3.623611    Control       2018-08-01              43     276.6393
## 3:        3.258209    Control       2018-09-01              39     231.4693
## 4:        3.318750    Control       2018-10-01              32     168.9101
## 5:        3.411864    Control       2018-11-01              38     213.4438
## 6:        3.791429    Control       2018-12-01              43     281.4107
```

Calculate the percentage difference between scaled control store sales and trial store sales

```
percentageDiff <-
  merge(scaledControlSales[, c("YEARMONTH", "controlSales")],
        measureOverTime[STORE_NBR == trial_store, c("totSales", "YEARMONTH")],
        by = "YEARMONTH")[, percentageDiff := abs(controlSales - totSales) / controlSales]
percentageDiff <- rename(percentageDiff, trialSales = totSales)
head(percentageDiff)
```

```
##    YEARMONTH controlSales trialSales percentageDiff
## 1:    201807     290.8476      273.8     0.05861365
## 2:    201808     276.6393      252.9     0.08581306
## 3:    201809     231.4693      219.6     0.05127824
## 4:    201810     168.9101      194.3     0.15031634
## 5:    201811     213.4438      224.9     0.05367322
## 6:    201812     281.4107      255.4     0.09242978
```

Is the percentage difference significant?

The null hypothesis is that the trial period is the same as the pre-trial period.

Determine the standard deviation based on the scaled percentage difference in the pre-trial period

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

```
## [1] 0.07623433
```

Determine the degrees of freedom.

Since there are 8 months in the pre-trial period, then 8 - 1 = 7 degrees of freedom

```
degreesOfFreedom <- 7
```

Test the null hypothesis of there being 0 difference between trial and control stores.

Calculate the t-values for the trial months.

```
percentageDiff[, tValue := (percentageDiff - 0) / stdDev
               ][, TransactionMonth := as.Date(paste(
                 YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
                 ][YEARMONTH < 201905 & YEARMONTH > 201901, .(TransactionMonth, tValue)]
```

```
##    TransactionMonth   tValue
## 1:       2019-02-01 1.097918
## 2:       2019-03-01 4.612199
## 3:       2019-04-01 9.488618
```

Find the 95th percentile of the t distribution with the appropriate degrees of freedom to check whether the hypothesis is statistically significant.

```
qt(0.95, df = degreesOfFreedom)
```

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

The t-value for March (4.61) and April (9.49) is much larger than the 95th percentile value of the t-distribution (1.89).

The increase in sales in the trial store in March and April is statistically greater than in the control store.

Plot 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.

Trial and control store total sales

```
pastSales <- measureOverTime[Store_type %in% c("Trial", "Control"), ]
```

Control store 5th and 95th percentile

```
pastSales_Controls5 <- pastSales[Store_type == "Control",
                                 ][, totSales := totSales * (1 - stdDev * 2)
                                 ][, Store_type := "Control 5th % confidence interval"]
pastSales_Controls95 <- pastSales[Store_type == "Control",
                                  ][, totSales := totSales * (1 + stdDev * 2)
                                  ][, Store_type := "Control 95th % confidence interval"]
```

Bind the measurements for the plot

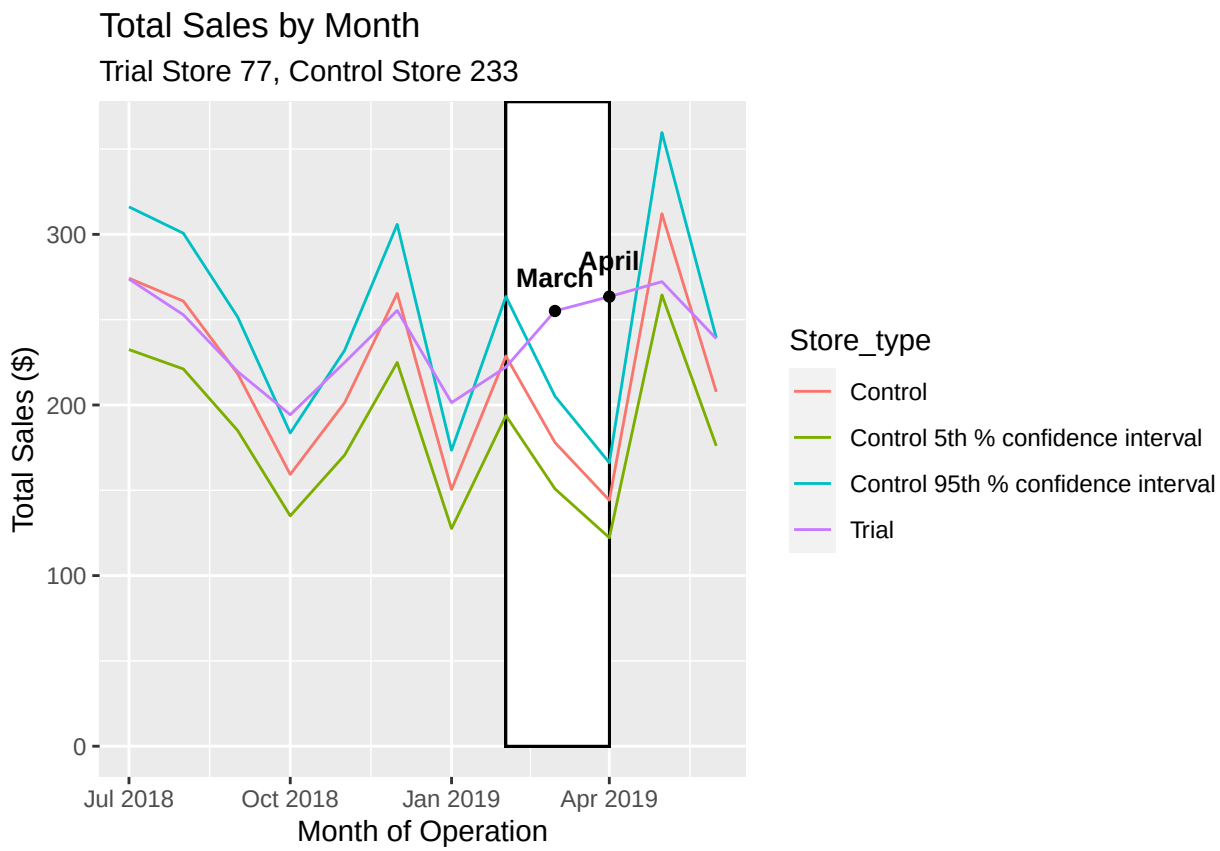
```
trialAssessmentSales <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)
```

Subset total sales from months with sales increases (for plot points)

```
pointSales <- subset(trialAssessmentSales,
                     STORE_NBR == 77 & YEARMONTH == 201903 |
                     STORE_NBR == 77 & YEARMONTH == 201904)
```

Make plot with a rectangular trial period

```
ggplot(trialAssessmentSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_rect(data = trialAssessmentSales[YEARMONTH < 201905 & YEARMONTH > 201901, ],
            aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
                ymin = 0, ymax = Inf, color = NULL),
            color = "black", fill = "white", show.legend = FALSE) +
  geom_line() +
  labs(x = "Month of Operation", y = "Total Sales ($)",
       title = "Total Sales by Month", subtitle = "Trial Store 77, Control Store 233") +
  geom_point(data = pointSales, color = "black") +
  annotate("text", label = "March",
           x = as.Date("2019-03-01"), y = 275,
           color = "black", size = 3.5, fontface = "bold") +
  annotate("text", label = "April",
           x = as.Date("2019-04-01"), y = 285,
           color = "black", size = 3.5, fontface = "bold")
```



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% and 95% confidence interval of the control store in 2 out of 3 trial months

Assessment 2: : Has there been an uplift in overall number of chips customers?

Repeat the steps before for total sales

Compute scaling factor for number of customers by dividing the sum of pre-trial trial store number of customers by the sum of pre-trial control store number of customers.

```
scalingFactorForControlCust <-
  preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum(nCustomers)] /
  preTrialMeasures[STORE_NBR == control_store & YEARMONTH < 201902, sum(nCustomers)]
scalingFactorForControlCust
```

```
## [1] 1.04
```

Apply the scaling factor to the control store in a new data frame by multiplying the total control store sales by the scaling factor.

```
scaledControlCustomers <-
  measureOverTime[STORE_NBR == control_store,
                  ][, controlCustomers := nCustomers * scalingFactorForControlCust
                    ][, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                             ifelse(STORE_NBR == control_store,
                                                    "Control", "Other stores"))
                      ]
head(scaledControlCustomers)
```

```
##    STORE_NBR YEARMONTH totSales nCustomers nTxnPerCust nChipsPerTxn
## 1:       233    201807    274.3         49    1.020408     1.600000
## 2:       233    201808    260.9         43    1.046512     1.600000
## 3:       233    201809    218.3         39    1.076923     1.595238
## 4:       233    201810    159.3         32    1.000000     1.500000
## 5:       233    201811    201.3         38    1.026316     1.512821
## 6:       233    201812    265.4         43    1.046512     1.555556
##    avgPricePerUnit Store_type TransactionMonth numberCustomers controlCustomers
## 1:        3.428750    Control       2018-07-01              49            50.96
## 2:        3.623611    Control       2018-08-01              43            44.72
## 3:        3.258209    Control       2018-09-01              39            40.56
## 4:        3.318750    Control       2018-10-01              32            33.28
## 5:        3.411864    Control       2018-11-01              38            39.52
## 6:        3.791429    Control       2018-12-01              43            44.72
```

Calculate the percentage difference between scaled control store number of customers and trial store number of customers.

```
percentageDiff <-
  merge(scaledControlCustomers[, c("YEARMONTH", "controlCustomers")],
        measureOverTime[STORE_NBR == trial_store, c("nCustomers", "YEARMONTH")],
        by = "YEARMONTH")[, percentageDiff :=
                            abs(controlCustomers - nCustomers) / controlCustomers]
percentageDiff <- rename(percentageDiff, trialCustomers = nCustomers)
head(percentageDiff)
```

```
##    YEARMONTH controlCustomers trialCustomers percentageDiff
## 1:    201807            50.96             48     0.05808477
## 2:    201808            44.72             47     0.05098390
## 3:    201809            40.56             40     0.01380671
## 4:    201810            33.28             36     0.08173077
## 5:    201811            39.52             39     0.01315789
## 6:    201812            44.72             42     0.06082290
```

Is the percentage difference significant?

The null hypothesis is that the trial period is the same as the pre-trial period.

Determine the standard deviation based on the scaled percentage difference in the pre-trial period

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

The degreesOfFreedom is still 7.

Test the null hypothesis of there being 0 difference between trial and control stores.

Calculate the t-values for the trial months.

```
percentageDiff[, tValue := (percentageDiff - 0) / stdDev
               ][, TransactionMonth := as.Date(paste(
                 YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
                 ][YEARMONTH < 201905 & YEARMONTH > 201901, .(TransactionMonth, tValue)]
```

```
##    TransactionMonth    tValue
## 1:       2019-02-01  1.520673
## 2:       2019-03-01 11.897026
## 3:       2019-04-01 26.639930
```

Find the 95th percentile of the t distribution with the appropriate degrees of freedom to check whether the hypothesis is statistically significant.

```
qt(0.95, df = degreesOfFreedom)
```

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

The t-value for March (11.90) and April (26.64) is much larger than the 95th percentile value of the t-distribution (1.89). ### The increase in number of customers in the trial store in March and April is statistically greater than in the control store.

Plot the number of customers in the control store, the the trial store, and the 5th and 95th percentile value of customer numbers in the control store.

Trial and control store number of customers

```
pastCustomers <-
  measureOverTime[, nCusts := mean(nCustomers), by = c("YEARMONTH", "Store_type")
                                  ][Store_type %in% c("Trial", "Control"), ]
```

Control store 5th and 95th percentile

```
pastCustomers_Controls5 <-
  pastCustomers[Store_type == "Control", ][, nCusts := nCusts * (1 - stdDev * 2)
                                           ][, Store_type :=
                                               "Control 5th % confidence interval"]
pastCustomers_Controls95 <-
  pastCustomers[Store_type == "Control",
                ][, nCusts := nCusts * (1 + stdDev * 2)
                  ][, Store_type := "Control 95th % confidence interval"]
```

Bind the measurements for the plot

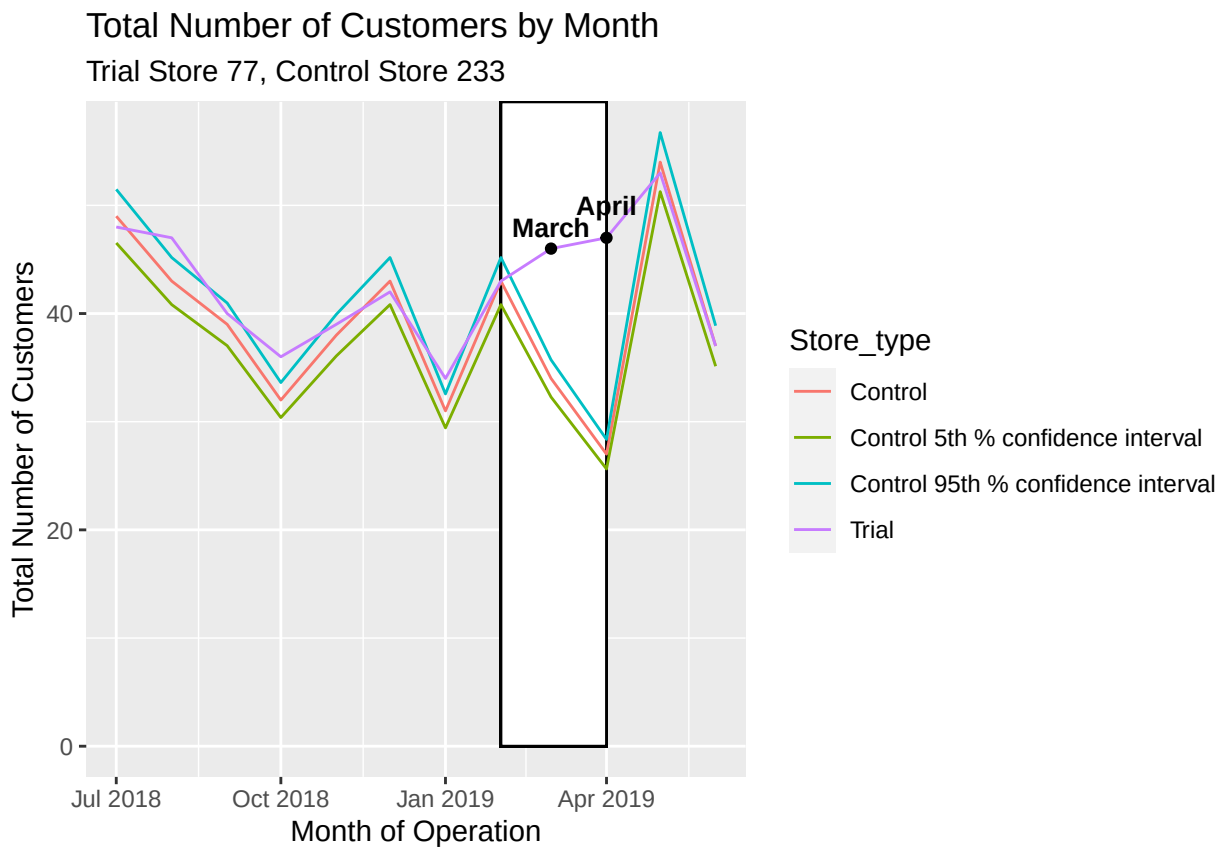
```
trialAssessmentCust <-
  rbind(pastCustomers, pastCustomers_Controls95, pastCustomers_Controls5)
```

Subset total customers from months with customer increases (for plot points)

```
pointCust <- subset(trialAssessmentCust,
                    STORE_NBR == 77 & YEARMONTH == 201903 |
                     STORE_NBR == 77 & YEARMONTH == 201904)
# the YEARMONTH filter prevents duplicates from pre-trial months
```

Plot for total customers

```
ggplot(trialAssessmentCust, aes(TransactionMonth, nCusts, color = Store_type)) +
 geom_rect(data = trialAssessmentCust[YEARMONTH < 201905 & YEARMONTH > 201901, ],
           aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
               ymin = 0, ymax = Inf, color = NULL),
           color = "black", fill = "white", show.legend = FALSE) +
 geom_line() +
 labs(x = "Month of Operation", y = "Total Number of Customers",
      title = "Total Number of Customers by Month",
      subtitle = "Trial Store 77, Control Store 233") +
  geom_point(data = pointCust, color = "black") +
  annotate("text", label = "March",
           x = as.Date("2019-03-01"), y = 48,
           color = "black", size = 3.5, fontface = "bold") +
  annotate("text", label = "April",
           x = as.Date("2019-04-01"), y = 50,
           color = "black", size = 3.5, fontface = "bold")
```



For trial store 77, The store layout changes during the trial period have resulted in significantly increased sales and number of customers, especially in the months of March and April.

Complete the same steps above for the other two trial stores (determine their respective control stores, craft assessments, and visualize the results).

SELECT A CONTROL STORE TO MATCH WITH TRIAL STORE 86

Assign the value 86 to trial_store for use in the functions.

```
trial_store <- 86
```

Calculate correlation and magnitude against store 86 using total sales and number of customers.

```
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales), trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers), trial_store)
magnitude_nSales <-
  calculateMagnitudeDistance(preTrialMeasures, quote(totSales), trial_store)
magnitude_nCustomers <-
  calculateMagnitudeDistance(preTrialMeasures, quote(nCustomers), trial_store)
```

Create a combined score composed of correlation and magnitude in order to determine the final control score measurement.

The corr_weight is still 0.5

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

```
                                    mag_measure * (1 - corr_weight)]
score_nCustomers <-
  merge(corr_nCustomers, magnitude_nCustomers,
        by = c("Store1", "Store2"))[, scoreNCust := corr_measure * corr_weight +
                                    mag_measure * (1 - corr_weight)]
score_Control <- merge(score_nSales, score_nCustomers, by = c("Store1", "Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]
score_Control[order(-finalControlScore)]
```

```
##      Store1 Store2 corr_measure.x mag_measure.x scoreNSales corr_measure.y
##   1:     86     86      1.0000000    1.00000000   1.0000000      1.0000000
##   2:     86    155      0.8251619    0.95439748   0.8897797      0.8094892
##   3:     86    114      0.7814783    0.93275912   0.8571187      0.8692800
##   4:     86     56      0.7801984    0.80934694   0.7947727      0.7852674
##   5:     86    138      0.7427750    0.93910688   0.8409409      0.4967472
##  ---
## 255:     86    120     -0.8915775    0.16869577  -0.3614409     -0.5631961
## 256:     86    192     -0.4057016    0.03107236  -0.1873146     -0.6845492
## 257:     86     52     -0.5374276    0.03410050  -0.2516636     -0.5688313
## 258:     86    146     -0.8207912    0.01821195  -0.4012896     -0.4801425
## 259:     86     42     -0.7629821    0.01794225  -0.3725199     -0.5724834
##      mag_measure.y  scoreNCust finalControlScore
##   1:    1.00000000  1.00000000         1.0000000
##   2:    0.96888493  0.88918708         0.8894834
##   3:    0.94218331  0.90573166         0.8814252
##   4:    0.83852171  0.81189458         0.8033336
##   5:    0.92286890  0.70980806         0.7753745
##  ---
## 255:    0.36413226 -0.09953193        -0.2304864
## 256:    0.05729043 -0.31362940        -0.2504720
## 257:    0.04156949 -0.26363090        -0.2576472
## 258:    0.02638174 -0.22688039        -0.3140850
## 259:    0.03735491 -0.26756425        -0.3200421
```

Select the most appropriate control store for trial store 86.

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

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

The control store for trial store 86 is store 155.

Visualizations will provide further confirmation.

Visualize the movement of sales and number of customers against the trial store, control store, and all other stores.

Recall the data frame measureOverTime

```
measureOverTime <- data[, .(totSales = sum(TOT_SALES),
                            nCustomers = uniqueN(LYLTY_CARD_NBR),
                            nTxnPerCust = uniqueN(TXN_ID)/uniqueN(LYLTY_CARD_NBR),
                            nChipsPerTxn = sum(PROD_QTY)/uniqueN(TXN_ID),
                            avgPricePerUnit = sum(TOT_SALES)/sum(PROD_QTY))
```

```
                        , by = c("STORE_NBR", "YEARMONTH")][order(STORE_NBR, YEARMONTH) ]
head(measureOverTime)
```

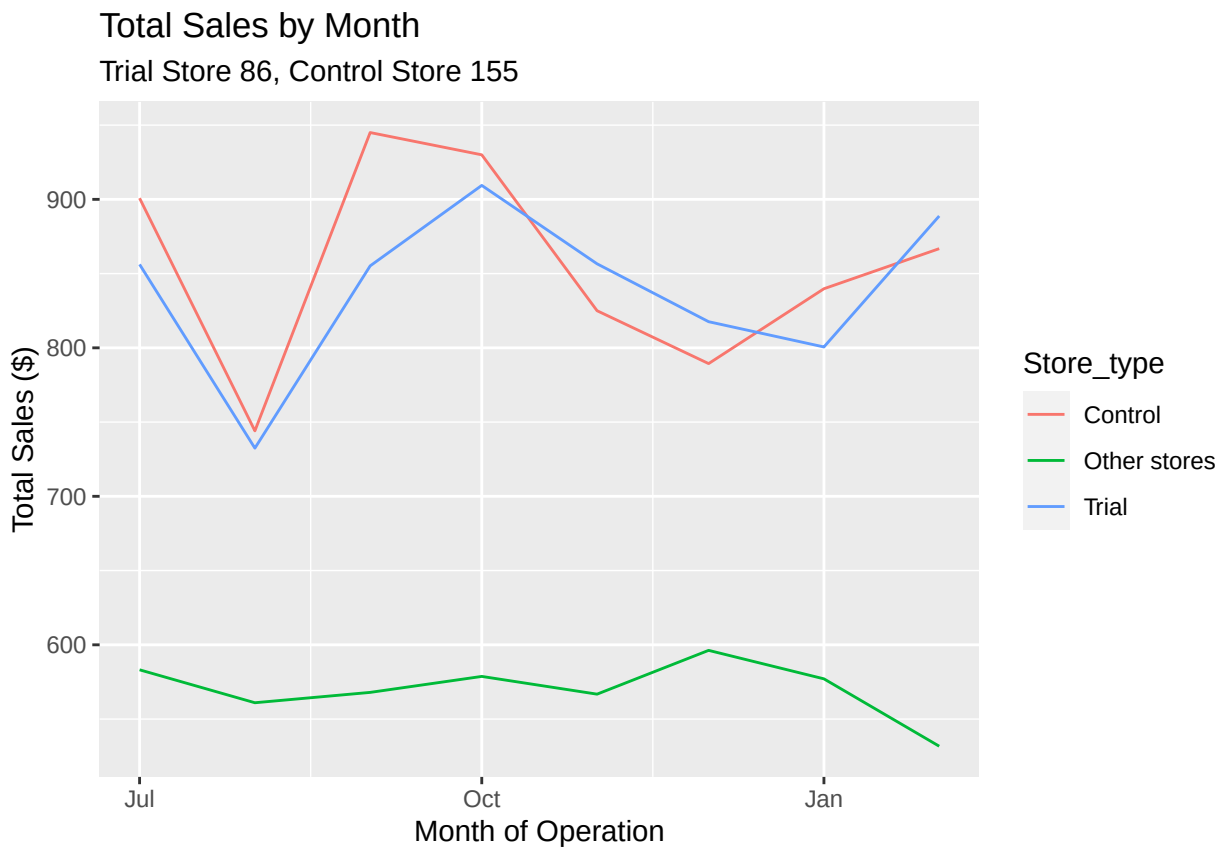
```
##    STORE_NBR YEARMONTH totSales nCustomers nTxnPerCust nChipsPerTxn
## 1:         1    201807    191.6         48    1.041667     1.180000
## 2:         1    201808    168.4         41    1.000000     1.268293
## 3:         1    201809    268.1         57    1.035088     1.203390
## 4:         1    201810    178.0         40    1.025000     1.268293
## 5:         1    201811    187.5         45    1.022222     1.217391
## 6:         1    201812    160.6         37    1.081081     1.200000
##    avgPricePerUnit
## 1:        3.247458
## 2:        3.238462
## 3:        3.776056
## 4:        3.423077
## 5:        3.348214
## 6:        3.345833
```

Prepare a sales data frame for plotting purposes. Demarcate the trial store, control store, and all other stores:

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

Plot the total sales by month for the pre-trial period.

```
ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line() +
  labs(x = "Month of Operation", y = "Total Sales ($)",
       title = "Total Sales by Month", subtitle = "Trial Store 86, Control Store 155")
```



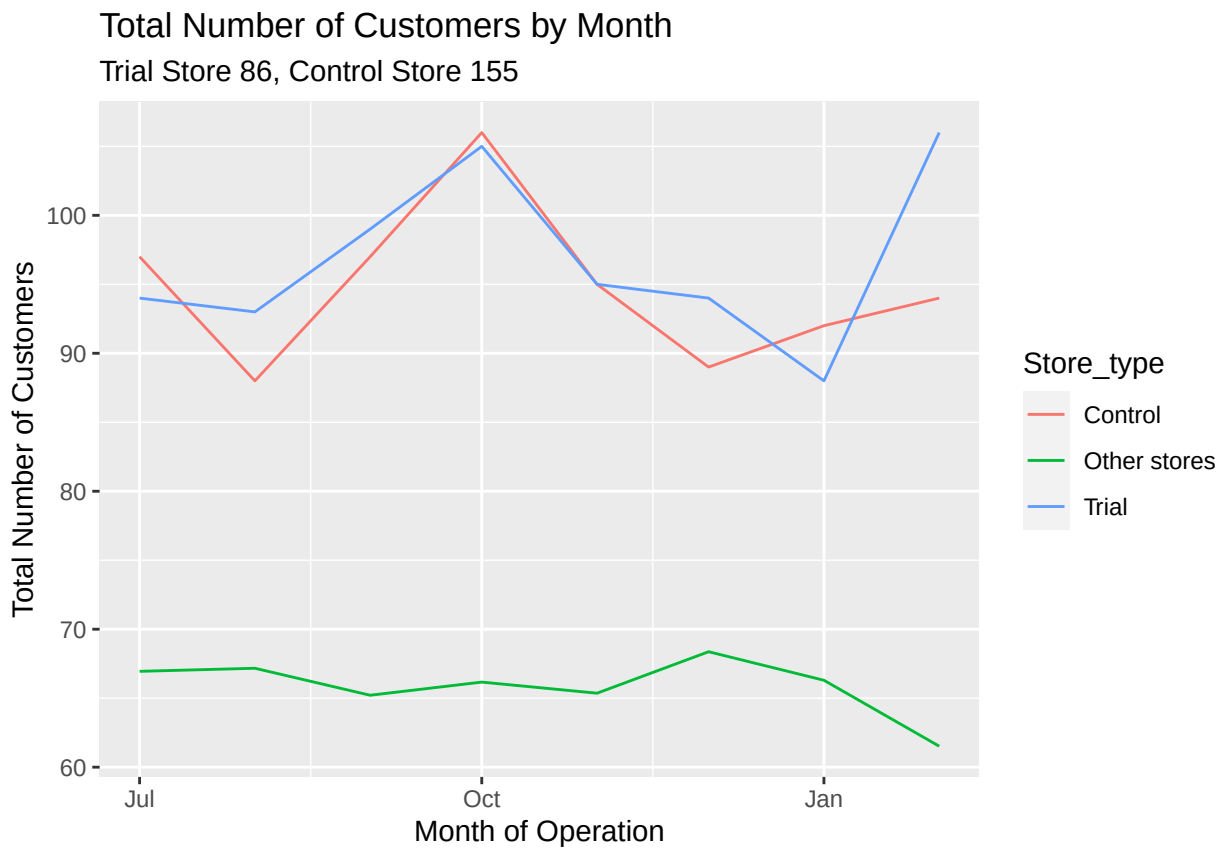
The plot shows that total sales are indeed similar between the trial and control stores.

Prepare a customer data frame for plotting purposes. Demarcate the trial store, control store, and all other stores:

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

Plot the total number of customers by month for the pre-trial period.

```
ggplot(pastCustomers, aes(TransactionMonth, numberCustomers, color = Store_type)) +
  geom_line() +
  labs(x = "Month of Operation", y = "Total Number of Customers",
       title = "Total Number of Customers by Month",
       subtitle = "Trial Store 86, Control Store 155")
```



ASSESSMENT OF TRIAL STORE 86: TOTAL SALES AND NUMBER OF CUSTOMERS

Assessment 1: Has there been an uplift in overall chip sales?

Compute scaling factor for sales by dividing the sum of pre-trial trial store sales by the sum of pre-trial control store sales.

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

```
## [1] 0.9755528
```

Apply the scaling factor to the control store by multiplying the total control store sales by the scaling factor.

```
scaledControlSales <-
  measureOverTime[STORE_NBR ==
                    control_store, ][ , controlSales :=
                                        totSales * scalingFactorForControlSales]
head(scaledControlSales)
```

```
##    STORE_NBR YEARMONTH totSales nCustomers nTxnPerCust nChipsPerTxn
## 1:       155    201807    900.8         97    1.216495     2.033898
## 2:       155    201808    744.1         88    1.295455     1.912281
## 3:       155    201809    945.0         97    1.350515     2.015267
## 4:       155    201810    930.0        106    1.235849     2.000000
```

```
## 5:       155    201811    825.0         95    1.263158     2.033333
## 6:       155    201812    789.4         89    1.235955     2.018182
##    avgPricePerUnit Store_type TransactionMonth numberCustomers controlSales
## 1:        3.753333    Control       2018-07-01              97     878.7780
## 2:        3.413303    Control       2018-08-01              88     725.9088
## 3:        3.579545    Control       2018-09-01              97     921.8974
## 4:        3.549618    Control       2018-10-01             106     907.2641
## 5:        3.381148    Control       2018-11-01              95     804.8311
## 6:        3.555856    Control       2018-12-01              89     770.1014
```

Calculate the percentage difference between scaled control store sales and trial store sales

```
percentageDiff <-
  merge(scaledControlSales[, c("YEARMONTH", "controlSales")],
        measureOverTime[STORE_NBR == trial_store, c("totSales", "YEARMONTH")],
        by = "YEARMONTH")[, percentageDiff :=
                            abs(controlSales - totSales) / controlSales]
percentageDiff <- rename(percentageDiff, trialSales = totSales)
head(percentageDiff)
```

```
##    YEARMONTH controlSales trialSales percentageDiff
## 1:    201807     878.7780     856.20    0.025692457
## 2:    201808     725.9088     732.45    0.009010993
## 3:    201809     921.8974     855.20    0.072347963
## 4:    201810     907.2641     909.40    0.002354213
## 5:    201811     804.8311     856.60    0.064322738
## 6:    201812     770.1014     817.60    0.061678395
```

Is the percentage difference significant?

The null hypothesis is that the trial period is the same as the pre-trial period.

Determine the standard deviation based on the scaled percentage difference in the pre-trial period

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

```
## [1] 0.0286313
```

The degreesOfFreedom is still 7.

Test the null hypothesis of there being 0 difference between trial and control stores.

Calculate the t-values for the trial months.

```
percentageDiff[, tValue := (percentageDiff - 0) / stdDev
               ][, TransactionMonth := as.Date(paste(
                 YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
                 ][YEARMONTH < 201905 & YEARMONTH > 201901, .(TransactionMonth, tValue)]
```

```
##    TransactionMonth   tValue
## 1:       2019-02-01 1.783943
## 2:       2019-03-01 9.034548
## 3:       2019-04-01 1.035250
```

Recall the 95th percentile of the t distribution with the appropriate degrees of freedom to check whether the hypothesis is statistically significant.

```
qt(0.95, df = degreesOfFreedom)
```

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

The t-value for March (9.03) is much larger than the 95th percentile value of the tdistribution (1.89).

The increase in sales in the trial store in March is statistically greater than in the control store.

Plot 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.

Trial and control store total sales

```
pastSales <- measureOverTime[Store_type %in% c("Trial", "Control"), ]
```

Control store 5th and 95th percentile

```
pastSales_Controls5 <- pastSales[Store_type == "Control",
                                 ][, totSales := totSales * (1 - stdDev * 2)
                                 ][, Store_type := "Control 5th % confidence interval"]
pastSales_Controls95 <- pastSales[Store_type == "Control",
                                 ][, totSales := totSales * (1 + stdDev * 2)
                                 ][, Store_type := "Control 95th % confidence interval"]
```

Bind the measurements for the plot

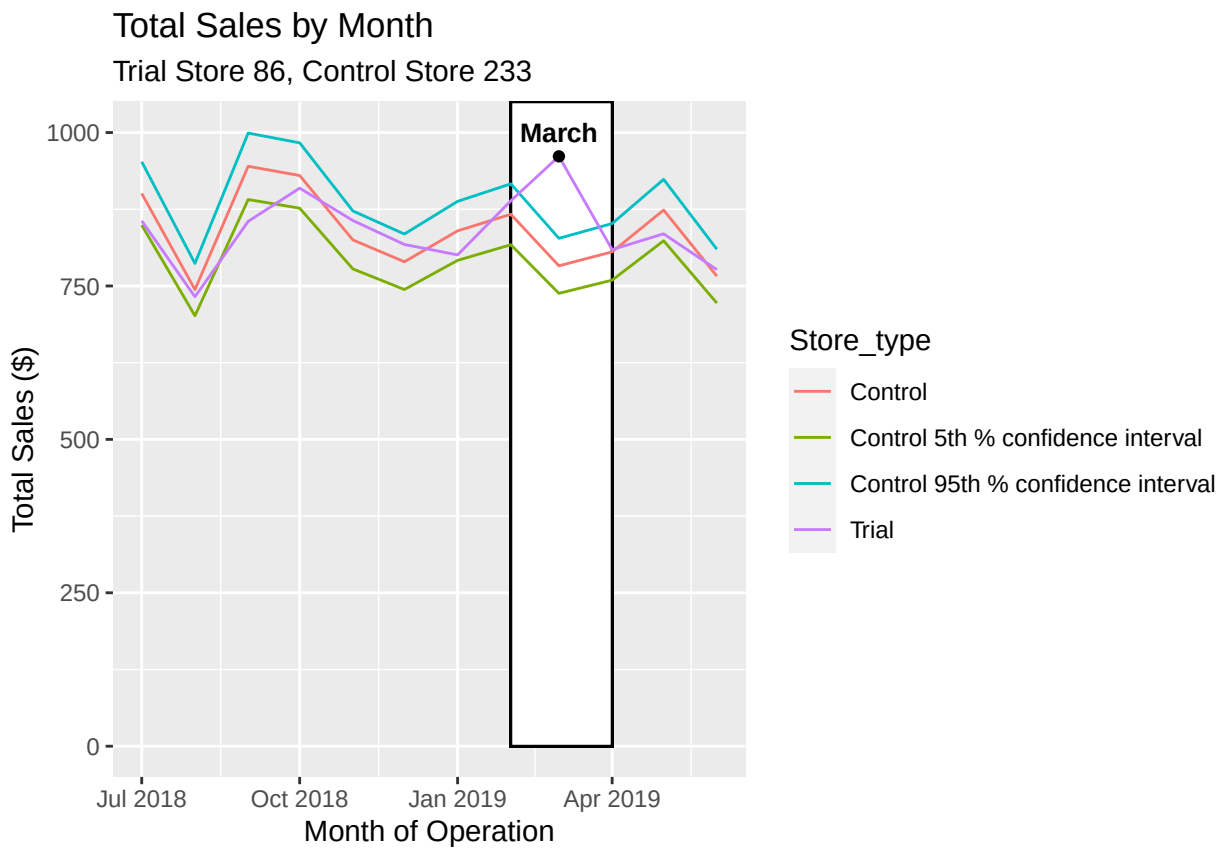
```
trialAssessmentSales <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)
```

Subset total sales from months with sales increases (for plot points)

```
pointSales <- subset(trialAssessmentSales, STORE_NBR == 86 & YEARMONTH == 201903)
```

Make plot with a rectangular trial period

```
ggplot(trialAssessmentSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_rect(data = trialAssessmentSales[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
            aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
            ymin = 0 , ymax = Inf, color = NULL),
            color = "black", fill = "white", show.legend = FALSE) +
  geom_line() +
  labs(x = "Month of Operation", y = "Total Sales ($)",
  title = "Total Sales by Month", subtitle = "Trial Store 86, Control Store 233") +
  geom_point(data = pointSales, color = "black") +
  annotate("text", label = "March",
           x = as.Date("2019-03-01"), y = 1000,
           color = "black", size = 3.5, fontface = "bold")
```



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.

Assessment 2: : Has there been an uplift in overall number of chips customers?

Repeat the steps before for total sales

Compute scaling factor for number of customers by dividing the sum of pre-trial trial store number of customers by the sum of pre-trial control store number of customers.

```
scalingFactorForControlCust <-
  preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum(nCustomers)] /
  preTrialMeasures[STORE_NBR == control_store & YEARMONTH < 201902, sum(nCustomers)]
scalingFactorForControlCust
```

```
## [1] 1.006024
```

Apply the scaling factor to the control store in a new data frame by multiplying the total control store sales by the scaling factor.

```
scaledControlCustomers <-
  measureOverTime[STORE_NBR == control_store,
                  ][, controlCustomers := nCustomers * scalingFactorForControlCust
                    ][, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                             ifelse(STORE_NBR == control_store,
                                                    "Control", "Other stores"))]
head(scaledControlCustomers)
```

```
##    STORE_NBR YEARMONTH totSales nCustomers nTxnPerCust nChipsPerTxn
## 1:       155    201807    900.8         97    1.216495     2.033898
## 2:       155    201808    744.1         88    1.295455     1.912281
## 3:       155    201809    945.0         97    1.350515     2.015267
## 4:       155    201810    930.0        106    1.235849     2.000000
## 5:       155    201811    825.0         95    1.263158     2.033333
## 6:       155    201812    789.4         89    1.235955     2.018182
##    avgPricePerUnit Store_type TransactionMonth numberCustomers controlCustomers
## 1:        3.753333    Control       2018-07-01              97         97.58434
## 2:        3.413303    Control       2018-08-01              88         88.53012
## 3:        3.579545    Control       2018-09-01              97         97.58434
## 4:        3.549618    Control       2018-10-01             106        106.63855
## 5:        3.381148    Control       2018-11-01              95         95.57229
## 6:        3.555856    Control       2018-12-01              89         89.53614
```

Calculate the percentage difference between scaled control store number of customers and trial store number of customers.

```
percentageDiff <-
  merge(scaledControlCustomers[, c("YEARMONTH", "controlCustomers")],
        measureOverTime[STORE_NBR == trial_store, c("nCustomers", "YEARMONTH")],
        by = "YEARMONTH")[, percentageDiff :=
                            abs(controlCustomers - nCustomers) / controlCustomers]
percentageDiff <- rename(percentageDiff, trialCustomers = nCustomers)
head(percentageDiff)
```

```
##    YEARMONTH controlCustomers trialCustomers percentageDiff
## 1:    201807         97.58434             94    0.036730662
## 2:    201808         88.53012             93    0.050489929
## 3:    201809         97.58434             99    0.014507068
## 4:    201810        106.63855            105    0.015365495
## 5:    201811         95.57229             95    0.005988024
## 6:    201812         89.53614             94    0.049855345
```

Is the percentage difference significant?

The null hypothesis is that the trial period is the same as the pre-trial period.

Determine the standard deviation based on the scaled percentage difference in the pre-trial period

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

```
## [1] 0.01931301
```

The degreesOfFreedom is still 7.

Test the null hypothesis of there being 0 difference between trial and control stores.

Calculate the t-values for the trial months.

```
percentageDiff[, tValue := (percentageDiff - 0) / stdDev
               ][, TransactionMonth := as.Date(paste(
                 YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
               ][YEARMONTH < 201905 & YEARMONTH > 201901, .(TransactionMonth, tValue)]
```

```
##    TransactionMonth   tValue
## 1:       2019-02-01 6.260398
## 2:       2019-03-01 9.759877
## 3:       2019-04-01 2.398818
```

Recall the 95th percentile of the t distribution with the appropriate degrees of freedom to check whether the hypothesis is statistically significant.

```
qt(0.95, df = degreesOfFreedom)
```

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

The t-value for all 3 months is much larger than the 95th percentile value of the tdistribution.

The increase in number of customers in the trial store for all 3 months is statistically greater than in the control store.

Plot the number of customers in the control store, the the trial store, and the 5th and 95th percentile value of customer numbers in the control store.

Trial and control store number of customers

```
pastCustomers <- measureOverTime[, nCusts := mean(nCustomers),
                                 by = c("YEARMONTH", "Store_type")
                                 ][Store_type %in% c("Trial", "Control"), ]
```

Control store 5th and 95th percentile

```
pastCustomers_Controls5 <-
  pastCustomers[Store_type == "Control",
                ][, nCusts := nCusts * (1 - stdDev * 2)
                  ][, Store_type := "Control 5th % confidence interval"]
pastCustomers_Controls95 <-
  pastCustomers[Store_type == "Control",
                ][, nCusts := nCusts * (1 + stdDev * 2)
                  ][, Store_type := "Control 95th % confidence interval"]
```

Bind the measurements for the plot

```
trialAssessmentCust <-
  rbind(pastCustomers, pastCustomers_Controls95, pastCustomers_Controls5)
```

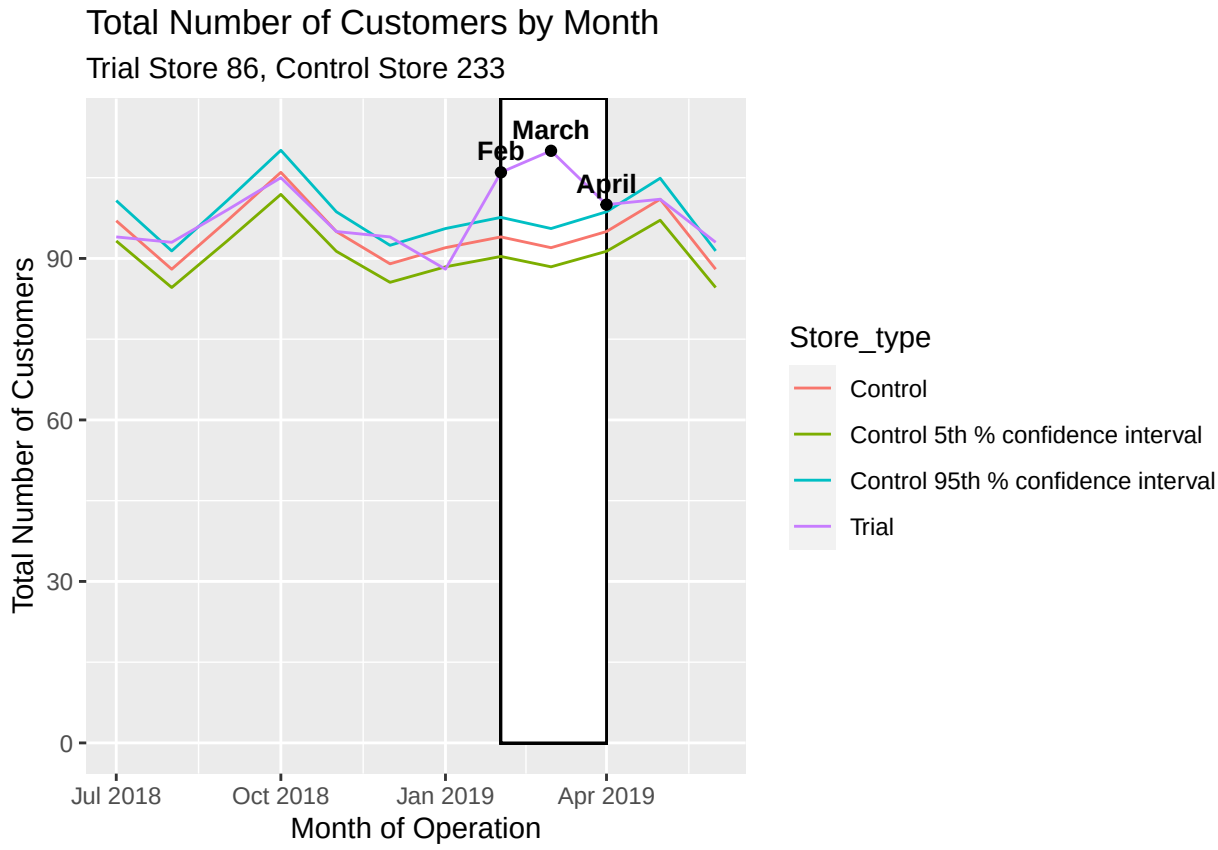
Subset total customers from months with customer increases (for plot points)

```
pointCust <-
  subset(trialAssessmentCust,
         STORE_NBR == 86 & YEARMONTH == 201902 |
         STORE_NBR == 86 & YEARMONTH == 201903 |
         STORE_NBR == 86 & YEARMONTH == 201904)
```

Plot for total customers

```
ggplot(trialAssessmentCust, aes(TransactionMonth, nCusts, color = Store_type)) +
  geom_rect(data = trialAssessmentCust[YEARMONTH < 201905 & YEARMONTH > 201901, ],
            aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
                ymin = 0, ymax = Inf, color = NULL),
            color = "black", fill = "white", show.legend = FALSE) +
  geom_line() +
  labs(x = "Month of Operation", y = "Total Number of Customers",
       title = "Total Number of Customers by Month",
       subtitle = "Trial Store 86, Control Store 233") +
  geom_point(data = pointCust, color = "black") +
  annotate("text", label = "Feb",
           x = as.Date("2019-02-01"), y = 110,
           color = "black", size = 3.5, fontface = "bold") +
```

```
  annotate("text", label = "March",
           x = as.Date("2019-03-01"), y = 114,
           color = "black", size = 3.5, fontface = "bold") +
  annotate("text", label = "April",
           x = as.Date("2019-04-01"), y = 104,
           color = "black", size = 3.5, fontface = "bold")
```



The number of customers is significantly higher in all of the three months.

The trial had a significant impact on increasing the number of customers in trial store 86 but sales were not significantly higher.

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 A CONTROL STORE TO MATCH WITH TRIAL STORE 88

Assign the value 88 to trial_store for use in the functions.

```
trial_store <- 88
```

Calculate correlation and magnitude against store 88 using total sales and number of customers.

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

```
magnitude_nCustomers <-
  calculateMagnitudeDistance(preTrialMeasures, quote(nCustomers), trial_store)
```

Create a combined score composed of correlation and magnitude in order to determine the final control score measurement.

The corr_weight is still 0.5

```
score_nSales <-
  merge(corr_nSales, magnitude_nSales,
        by = c("Store1", "Store2"))[, scoreNSales :=
                                     corr_measure * corr_weight +
                                     mag_measure * (1 - corr_weight)]
score_nCustomers <-
  merge(corr_nCustomers, magnitude_nCustomers,
        by = c("Store1", "Store2"))[, scoreNCust := corr_measure * corr_weight +
                                     mag_measure * (1 - corr_weight)]
score_Control <- merge(score_nSales, score_nCustomers, by = c("Store1", "Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]
score_Control[order(-finalControlScore)]
```

```
##      Store1 Store2 corr_measure.x mag_measure.x scoreNSales corr_measure.y
##   1:     88     88      1.0000000    1.00000000   1.0000000    1.000000000
##   2:     88    237      0.1098783    0.94267292   0.5262756    0.958895800
##   3:     88    123      0.4023386    0.85138678   0.6268627    0.660125169
##   4:     88    178      0.2877427    0.69360134   0.4906720    0.903192241
##   5:     88      7      0.6988456    0.78230952   0.7405775    0.345950663
##  ---
## 255:     88    264     -0.6630836    0.15517994  -0.2539518   -0.328623344
## 256:     88    235     -0.8348418    0.35411106  -0.2403654   -0.430325742
## 257:     88    239     -0.3661553    0.24990132  -0.0581270   -0.703172301
## 258:     88    135     -0.5928549    0.01635704  -0.2882489    0.009535493
## 259:     88    141     -0.7184766    0.20401783  -0.2572294   -0.643610755
##      mag_measure.y   scoreNCust finalControlScore
##   1:    1.00000000  1.000000000         1.0000000
##   2:    0.97846805  0.968681923         0.7474788
##   3:    0.88327867  0.771701921         0.6992823
##   4:    0.81398677  0.858589507         0.6746308
##   5:    0.83522857  0.590589615         0.6655836
##  ---
## 255:    0.33303393  0.002205292        -0.1258733
## 256:    0.39307827 -0.018623734        -0.1294946
## 257:    0.29402967 -0.204571313        -0.1313492
## 258:    0.03347902  0.021507258        -0.1333708
## 259:    0.29768871 -0.172961024        -0.2150952
```

Select the most appropriate control store for trial store 88.

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

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

The control store for trial store 88 is store 237.

Visualizations will provide further confirmation.

Visualize the movement of sales and number of customers against the trial store, control store, and all other stores.

Recall the data frame measureOverTime

```
measureOverTime <- data[, .(totSales = sum(TOT_SALES),
                            nCustomers = uniqueN(LYLTY_CARD_NBR),
                            nTxnPerCust = uniqueN(TXN_ID)/uniqueN(LYLTY_CARD_NBR),
                            nChipsPerTxn = sum(PROD_QTY)/uniqueN(TXN_ID),
                            avgPricePerUnit = sum(TOT_SALES)/sum(PROD_QTY))
                        , by = c("STORE_NBR", "YEARMONTH")][order(STORE_NBR, YEARMONTH)]
head(measureOverTime)
```

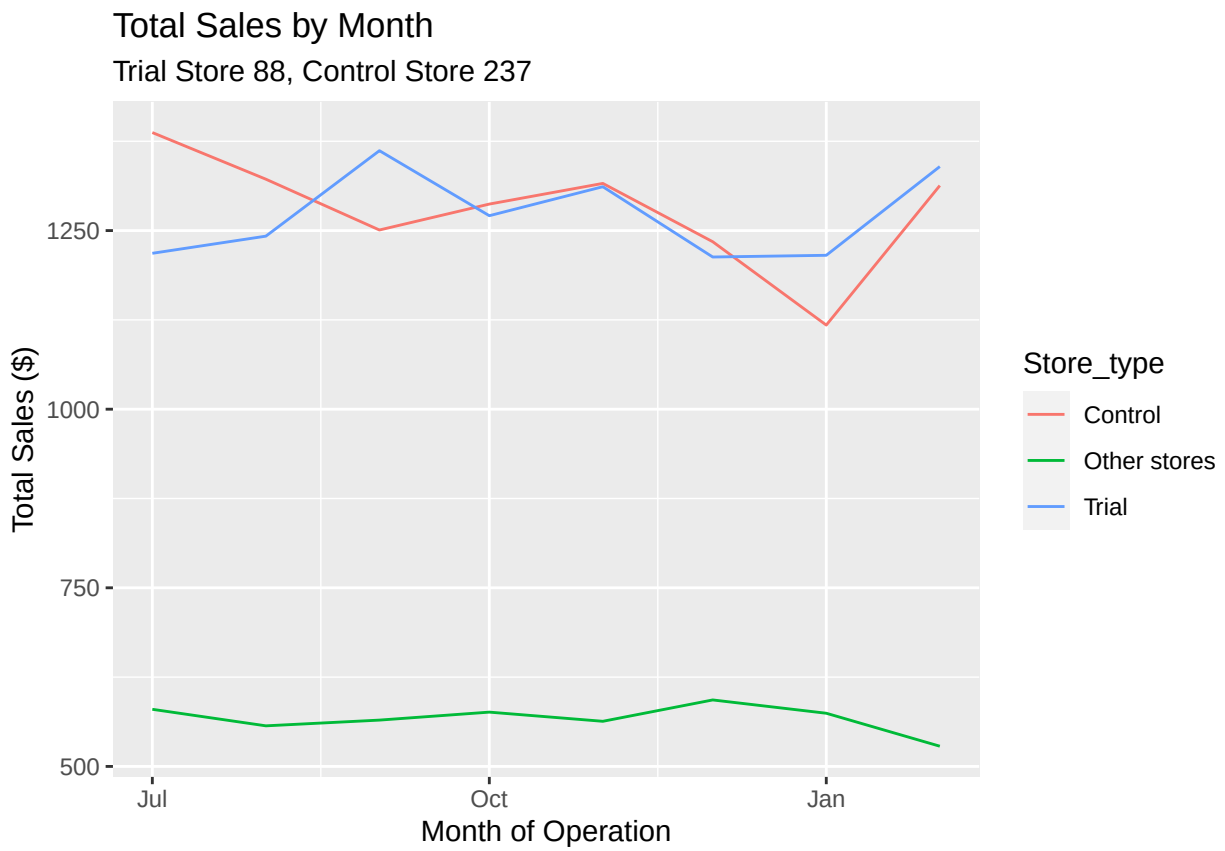
```
##    STORE_NBR YEARMONTH totSales nCustomers nTxnPerCust nChipsPerTxn
## 1:         1    201807    191.6         48    1.041667     1.180000
## 2:         1    201808    168.4         41    1.000000     1.268293
## 3:         1    201809    268.1         57    1.035088     1.203390
## 4:         1    201810    178.0         40    1.025000     1.268293
## 5:         1    201811    187.5         45    1.022222     1.217391
## 6:         1    201812    160.6         37    1.081081     1.200000
##    avgPricePerUnit
## 1:        3.247458
## 2:        3.238462
## 3:        3.776056
## 4:        3.423077
## 5:        3.348214
## 6:        3.345833
```

Prepare a sales data frame for plotting purposes. Demarcate the trial store, control store, and all other stores:

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

Plot the total sales by month for the pre-trial period.

```
ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line() +
  labs(x = "Month of Operation", y = "Total Sales ($)",
       title = "Total Sales by Month", subtitle = "Trial Store 88, Control Store 237")
```



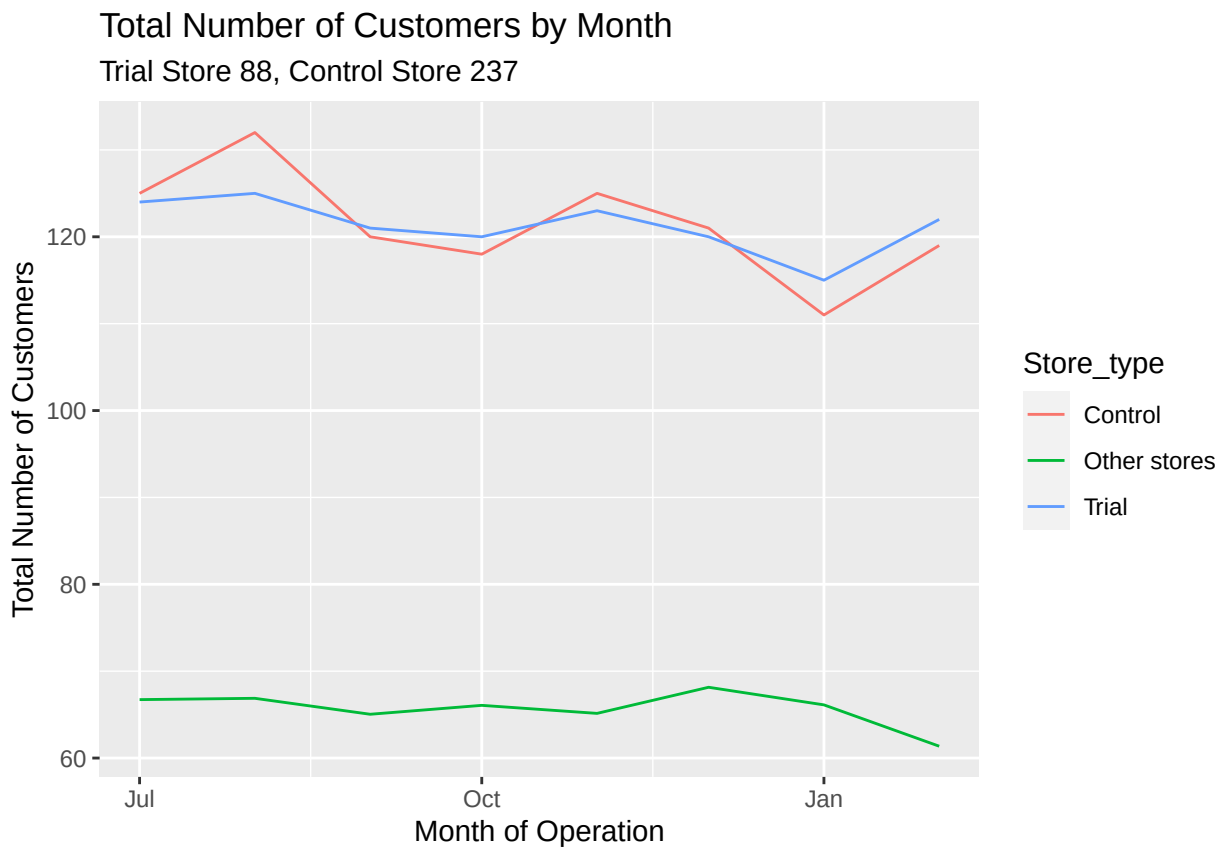
The plot shows that total sales are indeed similar between the trial and control stores.

Prepare a customer data frame for plotting purposes. Demarcate the trial store, control store, and all other stores:

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

Plot the total number of customers by month for the pre-trial period.

```
ggplot(pastCustomers, aes(TransactionMonth, numberCustomers, color = Store_type)) +
  geom_line() +
  labs(x = "Month of Operation", y = "Total Number of Customers",
       title = "Total Number of Customers by Month",
       subtitle = "Trial Store 88, Control Store 237")
```



ASSESSMENT OF TRIAL STORE 88: TOTAL SALES AND NUMBER OF CUSTOMERS

Assessment 1: Has there been an uplift in overall chip sales?

Compute scaling factor for sales by dividing the sum of pre-trial trial store sales by the sum of pre-trial control store sales.

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

```
## [1] 0.9907685
```

Apply the scaling factor to the control store by multiplying the total control store sales by the scaling factor.

```
scaledControlSales <-
  measureOverTime[STORE_NBR == control_store,
                  ][ , controlSales := totSales * scalingFactorForControlSales]
head(scaledControlSales)
```

```
##    STORE_NBR YEARMONTH totSales nCustomers nTxnPerCust nChipsPerTxn
## 1:       237    201807   1387.2        125    1.248000     2.000000
## 2:       237    201808   1321.9        132    1.212121     1.900000
## 3:       237    201809   1250.8        120    1.183333     2.007042
## 4:       237    201810   1287.1        118    1.194915     2.035461
## 5:       237    201811   1316.0        125    1.224000     1.986928
```

```
## 6:       237    201812   1234.4        121    1.165289     2.007092
##    avgPricePerUnit Store_type TransactionMonth numberCustomers controlSales
## 1:        4.446154    Control       2018-07-01             125     1374.394
## 2:        4.348355    Control       2018-08-01             132     1309.697
## 3:        4.388772    Control       2018-09-01             120     1239.253
## 4:        4.484669    Control       2018-10-01             118     1275.218
## 5:        4.328947    Control       2018-11-01             125     1303.851
## 6:        4.361837    Control       2018-12-01             121     1223.005
```

Calculate the percentage difference between scaled control store sales and trial store sales

```
percentageDiff <-
  merge(scaledControlSales[, c("YEARMONTH", "controlSales")],
        measureOverTime[STORE_NBR == trial_store, c("totSales", "YEARMONTH")],
        by = "YEARMONTH")[, percentageDiff :=
                            abs(controlSales - totSales) / controlSales]
percentageDiff <- rename(percentageDiff, trialSales = totSales)
head(percentageDiff)
```

```
##    YEARMONTH controlSales trialSales percentageDiff
## 1:    201807     1374.394     1218.2    0.113645738
## 2:    201808     1309.697     1242.2    0.051536234
## 3:    201809     1239.253     1361.8    0.098887617
## 4:    201810     1275.218     1270.8    0.003464584
## 5:    201811     1303.851     1311.4    0.005789534
## 6:    201812     1223.005     1213.0    0.008180346
```

Is the percentage difference significant?

The null hypothesis is that the trial period is the same as the pre-trial period.

Determine the standard deviation based on the scaled percentage difference in the pre-trial period

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

```
## [1] 0.04907816
```

The degreesOfFreedom is still 7.

Test the null hypothesis of there being 0 difference between trial and control stores.

Calculate the t-values for the trial months.

```
percentageDiff[, tValue := (percentageDiff - 0) / stdDev
               ][, TransactionMonth := as.Date(paste(
                 YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
                 ][YEARMONTH < 201905 & YEARMONTH > 201901, .(TransactionMonth, tValue)]
```

```
##    TransactionMonth    tValue
## 1:       2019-02-01 0.6064868
## 2:       2019-03-01 5.2439100
## 3:       2019-04-01 3.1028236
```

Recall the 95th percentile of the t distribution with the appropriate degrees of freedom to check whether the hypothesis is statistically significant.

```
qt(0.95, df = degreesOfFreedom)
```

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

The t-value for March (5.24) and April (3.10) is much larger than the 95th percentile value of the tdistribution (1.89).

The increase in sales in the trial store in March and April is statistically greater than in the control store.

Plot 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.

Trial and control store total sales

```
pastSales <- measureOverTime[Store_type %in% c("Trial", "Control"), ]
```

Control store 5th and 95th percentile

```
pastSales_Controls5 <- pastSales[Store_type == "Control",
                                 ][, totSales := totSales * (1 - stdDev * 2)
                                 ][, Store_type := "Control 5th % confidence interval"]
pastSales_Controls95 <- pastSales[Store_type == "Control",
                                 ][, totSales := totSales * (1 + stdDev * 2)
                                 ][, Store_type := "Control 95th % confidence interval"]
```

Bind the measurements for the plot

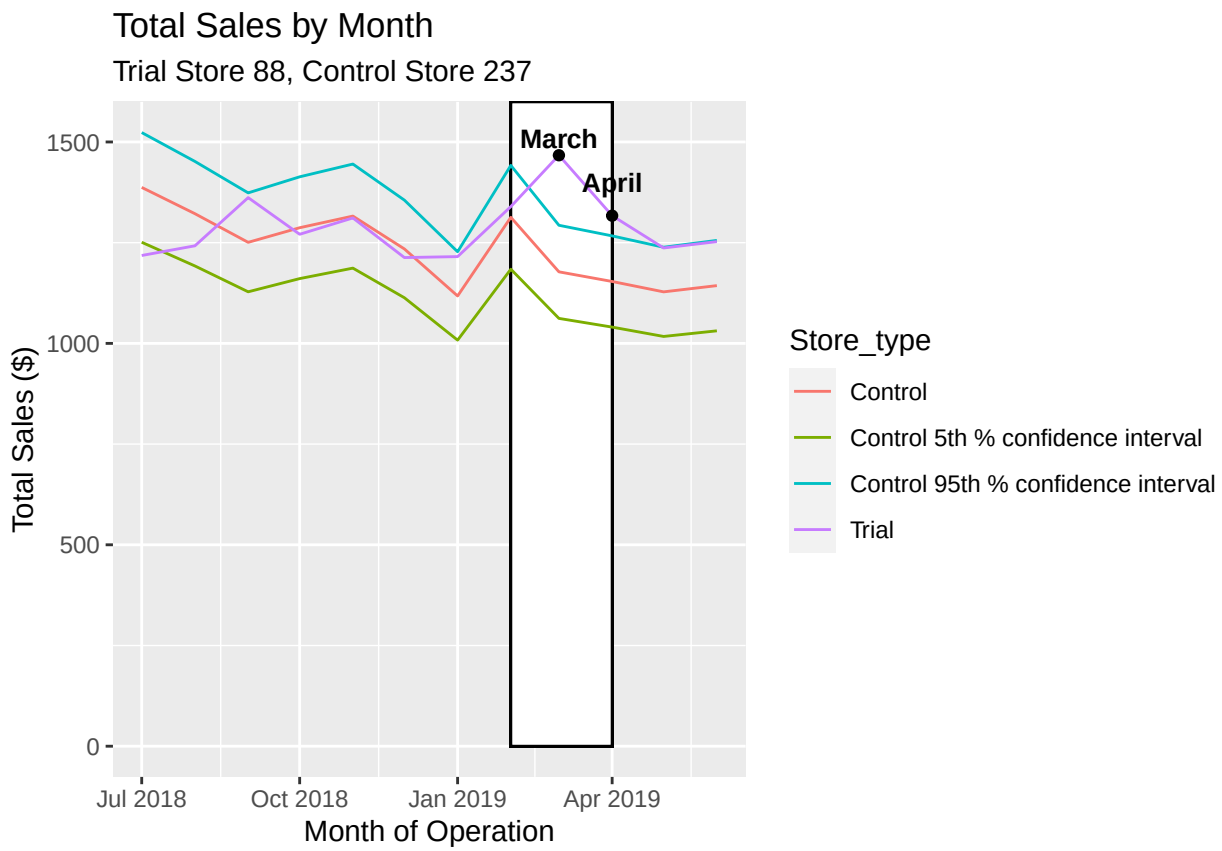
```
trialAssessmentSales <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)
```

Subset total sales from months with sales increases (for plot points)

```
pointSales <- subset(trialAssessmentSales,
                     STORE_NBR == 88 & YEARMONTH == 201903 |
                     STORE_NBR == 88 & YEARMONTH == 201904)
```

Make plot with a rectangular trial period

```
ggplot(trialAssessmentSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_rect(data = trialAssessmentSales[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
            aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
                ymin = 0 , ymax = Inf, color = NULL),
            color = "black", fill = "white", show.legend = FALSE) +
geom_line() +
  labs(x = "Month of Operation", y = "Total Sales ($)",
       title = "Total Sales by Month", subtitle = "Trial Store 88, Control Store 237") +
  geom_point(data = pointSales, color = "black") +
  annotate("text", label = "March", x = as.Date("2019-03-01"), y = 1510,
           color = "black", size = 3.5, fontface = "bold") +
  annotate("text", label = "April", x = as.Date("2019-04-01"), y = 1400,
           color = "black", size = 3.5, fontface = "bold")
```



The results show that the trial in store 88 is significantly different to its control store in the trial period.

The trial store performance lies outside of the 5% to 95% confidence interval of the control store in two of the three trial months.

Assessment 2: : Has there been an uplift in overall number of chips customers?

Repeat the steps before for total sales

Compute scaling factor for number of customers by dividing the sum of pre-trial trial store number of customers by the sum of pre-trial control store number of customers.

```
scalingFactorForControlCust <-
preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum(nCustomers)] /
preTrialMeasures[STORE_NBR == control_store & YEARMONTH < 201902, sum(nCustomers)]
scalingFactorForControlCust
```

```
## [1] 0.9953052
```

Apply the scaling factor to the control store in a new data frame by multiplying the total control store sales by the scaling factor.

```
scaledControlCustomers <-
  measureOverTime[STORE_NBR == control_store,
                  ][ , controlCustomers := nCustomers * scalingFactorForControlCust
                     ][, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                              ifelse(STORE_NBR == control_store,
```

```
                                                    "Control", "Other stores"))]
head(scaledControlCustomers)
```

```
##    STORE_NBR YEARMONTH totSales nCustomers nTxnPerCust nChipsPerTxn
## 1:       237    201807   1387.2        125    1.248000     2.000000
## 2:       237    201808   1321.9        132    1.212121     1.900000
## 3:       237    201809   1250.8        120    1.183333     2.007042
## 4:       237    201810   1287.1        118    1.194915     2.035461
## 5:       237    201811   1316.0        125    1.224000     1.986928
## 6:       237    201812   1234.4        121    1.165289     2.007092
##    avgPricePerUnit Store_type TransactionMonth numberCustomers controlCustomers
## 1:        4.446154    Control       2018-07-01             125         124.4131
## 2:        4.348355    Control       2018-08-01             132         131.3803
## 3:        4.388772    Control       2018-09-01             120         119.4366
## 4:        4.484669    Control       2018-10-01             118         117.4460
## 5:        4.328947    Control       2018-11-01             125         124.4131
## 6:        4.361837    Control       2018-12-01             121         120.4319
```

Calculate the percentage difference between scaled control store number of customers and trial store number of customers.

```
percentageDiff <-
  merge(scaledControlCustomers[, c("YEARMONTH", "controlCustomers")],
        measureOverTime[STORE_NBR == trial_store, c("nCustomers", "YEARMONTH")],
        by = "YEARMONTH")[, percentageDiff :=
                            abs(controlCustomers - nCustomers)/controlCustomers]
percentageDiff <- rename(percentageDiff, trialCustomers = nCustomers)
head(percentageDiff)
```

```
##    YEARMONTH controlCustomers trialCustomers percentageDiff
## 1:    201807         124.4131            124    0.003320755
## 2:    201808         131.3803            125    0.048563465
## 3:    201809         119.4366            121    0.013089623
## 4:    201810         117.4460            120    0.021746083
## 5:    201811         124.4131            123    0.011358491
## 6:    201812         120.4319            120    0.003586465
```

Is the percentage difference significant?

The null hypothesis is that the trial period is the same as the pre-trial period.

Determine the standard deviation based on the scaled percentage difference in the pre-trial period

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

```
## [1] 0.01791538
```

The degreesOfFreedom is still 7.

Test the null hypothesis of there being 0 difference between trial and control stores.

Calculate the t-values for the trial months.

```
percentageDiff[, tValue := (percentageDiff - 0) / stdDev
               ][, TransactionMonth := as.Date(paste(
                 YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
                 ][YEARMONTH < 201905 & YEARMONTH > 201901, .(TransactionMonth, tValue)]
```

```
##    TransactionMonth   tValue
```

```
## 1:       2019-02-01 1.677105
## 2:       2019-03-01 8.482095
## 3:       2019-04-01 1.713669
```

Recall the 95th percentile of the t distribution with the appropriate degrees of freedom to check whether the hypothesis is statistically significant.

```
qt(0.95, df = degreesOfFreedom)
```

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

The t-value for March (8.48) is much larger than the 95th percentile value of the tdistribution (1.89).

The increase in number of customers in the trial store for March is statistically greater than in the control store.

Plot the number of customers in the control store, the the trial store, and the 5th and 95th percentile value of customer numbers in the control store.

Trial and control store number of customers

```
pastCustomers <-
  measureOverTime[, nCusts := mean(nCustomers), by = c("YEARMONTH", "Store_type")
                  ][Store_type %in% c("Trial", "Control"), ]
```

Control store 5th and 95th percentile

```
pastCustomers_Controls5 <-
  pastCustomers[Store_type == "Control",
                ][, nCusts := nCusts * (1 - stdDev * 2)
                  ][, Store_type := "Control 5th % confidence interval"]
pastCustomers_Controls95 <-
  pastCustomers[Store_type == "Control",
                ][, nCusts := nCusts * (1 + stdDev * 2)
                  ][, Store_type := "Control 95th % confidence interval"]
```

Bind the measurements for the plot

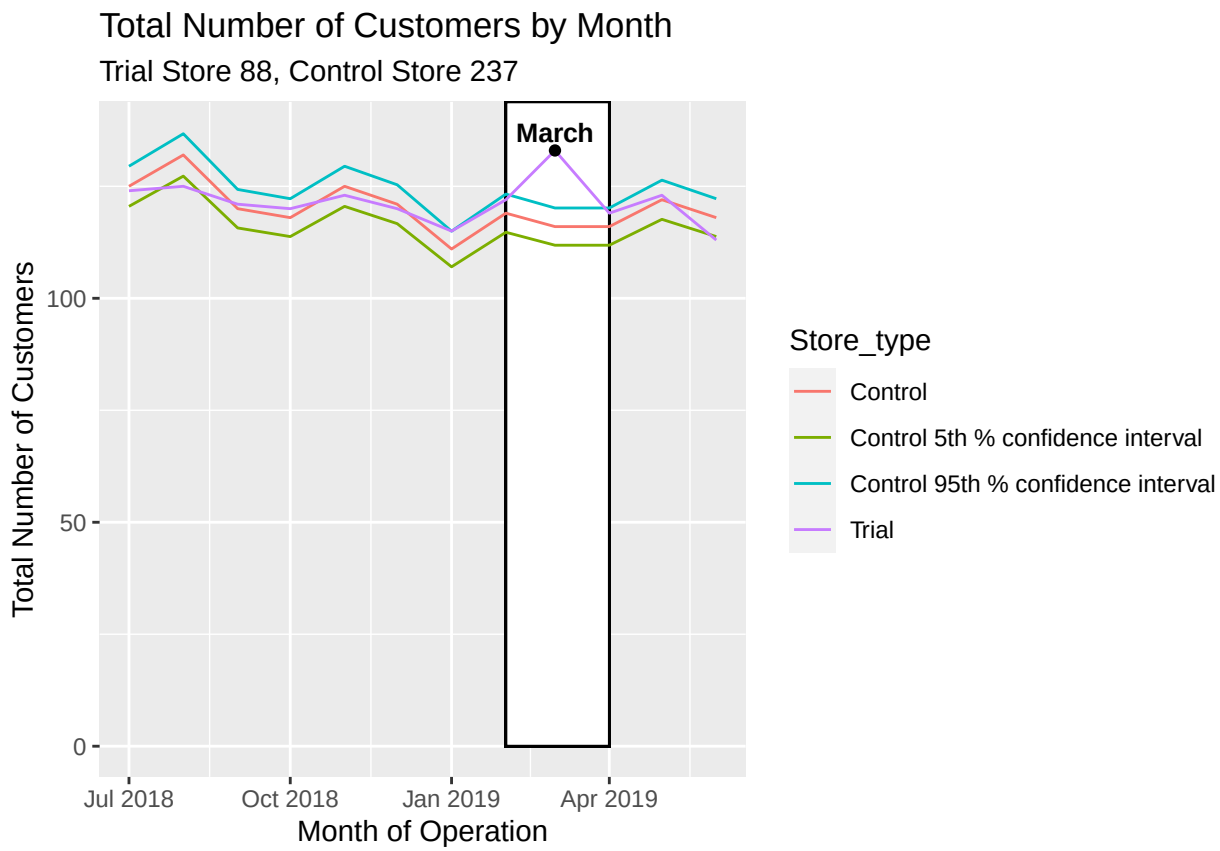
```
trialAssessmentCust <-
  rbind(pastCustomers, pastCustomers_Controls95, pastCustomers_Controls5)
```

Subset total customers from months with customer increases (for plot points)

```
pointCust <- subset(trialAssessmentCust, STORE_NBR == 88 & YEARMONTH == 201903)
```

Make plot with a rectangular trial period

```
ggplot(trialAssessmentCust, aes(TransactionMonth, nCusts, color = Store_type)) +
  geom_rect(data = trialAssessmentCust[ YEARMONTH < 201905 & YEARMONTH > 201901, ],
            aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
                ymin = 0, ymax = Inf, color = NULL),
            color = "black", fill = "white", show.legend = FALSE) +
geom_line() +
  labs(x = "Month of Operation", y = "Total Number of Customers",
       title = "Total Number of Customers by Month",
       subtitle = "Trial Store 88, Control Store 237") +
  geom_point(data = pointCust, color = "black") +
  annotate("text", label = "March",
           x = as.Date("2019-03-01"), y = 137,
           color = "black", size = 3.5, fontface = "bold")
```



Total number of customers in the trial period for the trial store were not significantly higher than the control store as the trial store performance lies inside the 5% to 95% confidence interval of the control store in two of the three trial months.

The trial had a significant impact on increasing sales in trial store 88 but number of customers were not significantly higher.

CONCLUSION

Control stores 233, 155, 237 are for trial stores 77, 86 and 88 respectively.

For trial store 77, The store layout changes during the trial period have resulted in significantly increased sales and number of customers, especially in the months of March and April.

The trial had a significant impact on increasing the number of customers in trial store 86 but sales were not significantly higher.

The trial had a significant impact on increasing sales in trial store 88 but number of customers were not significantly higher.

Check with the client if the implementation of the trial was different in trial store 86 or 88, but generally, the trial shows a noticeable increase in both sales and number of customers.

Now prepare the presentation to the Category Manager.