Retail Strategy and Analytics - Task 2

Siddharth Gada

```
## ## Chunk: Loading Libraries
```

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

```
## ## Chunk: Loading Dataset
```

```
data <- read_csv(paste0(
  "C:/Users/gadas/OneDrive/Desktop/",
  "Classes Outside UNT/Forage Project/",
  "Quantium Data Analytics/QVI_data.csv"))

setDT(data)

# Deleting extra column created while importing the file
data$...1 <- NULL

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

# New month ID column in the data with the format yyyymm
data[, monthID := year(DATE_Converted)*100 + month(DATE_Converted)]
```

```
## ## Chunk: Measures
```

```
# Define the measure calculations
setDT(data)
measureOverTime <- data[
  , .(
  # Monthly overall sales revenue by store
  sales_total = sum(TOT_SALES),
  # Monthly number of customers by store
  nCustomers = uniqueN(LYLTY_CARD_NBR),
  # Monthly number of transactions per customer
  nTransactionPerCust = uniqueN(TXN_ID)/uniqueN(LYLTY_CARD_NBR),
  # Monthly number of chips bought per customer
  monthlyUnits = sum(PROD_QTY)/uniqueN(TXN_ID),
  # Monthly Average price of chips bought
  avgPricePerUnit = sum(TOT_SALES)/sum(PROD_QTY)
  ),
```

```
  by = .(STORE_NBR, monthID)
  ][order(STORE_NBR, monthID)]
```

Chunk: Pre-Trial Period

```
#### Filter to the pre-trial period and stores with full observation periods
setDT(measureOverTime)
storesWithFullObs <- unique(measureOverTime[, .N, STORE_NBR][N == 12, STORE_NBR])
preTrialMeasures <- measureOverTime[monthID < 201902 & STORE_NBR %in% storesWithFullObs, ]
```

Chunk: Function to calculate Correlation

```
# Create a function to calculate correlation for a measure, looping through each control store
calculateCorrelation <- function(inputTable, metricCol, storeComparison) {
  calcCorrTable = data.table(Store1 = numeric(), Store2 = numeric(),
                             corr_measure = numeric())
  storeNumbers <- unique(inputTable[, STORE_NBR])
    for (i in storeNumbers) {
      calculatedMeasure = data.table(
        "Store1" = storeComparison,
        "Store2" = i,
        "corr_measure" = cor(inputTable[STORE_NBR == storeComparison, eval(metricCol)],
                             inputTable[STORE_NBR == i, eval(metricCol)]))
      calcCorrTable <- rbind(calcCorrTable, calculatedMeasure)
    }
  calcCorrTable <- calcCorrTable %>%
    arrange(desc(rowMeans(select(., starts_with("corr_")))))
  return(calcCorrTable)
  }
```

Chunk: Function to calculate Magnitude Distance

```
# Create function to calculate magnitude distance
calculateMagnitudeDistance <- function(inputTable, metricCol, storeComparison) {
  calcDistTable = data.table(Store1 = numeric(), Store2 = numeric(),
                             monthID = numeric(), measure = numeric())
  storeNumbers <- unique(inputTable[, STORE_NBR])
  for (i in storeNumbers) {
    calculatedMeasure = data.table(
      "Store1" = storeComparison,
      "Store2" = i,
      "monthID" = inputTable[STORE_NBR == storeComparison, monthID],
      "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", "monthID")]
  distTable <- merge(calcDistTable, minMaxDist, by = c("Store1", "monthID"))
```

```
  distTable[, magnitudeMeasure := 1 - (measure - minDist)/(maxDist - minDist)]
  finalDistTable <- distTable[, .(mag_measure = mean(magnitudeMeasure)),
                              by = .(Store1, Store2)]
  finalDistTable <- finalDistTable %>%
  arrange(desc(rowMeans(select(., starts_with("mag_")))))
  return(finalDistTable)
  }
```

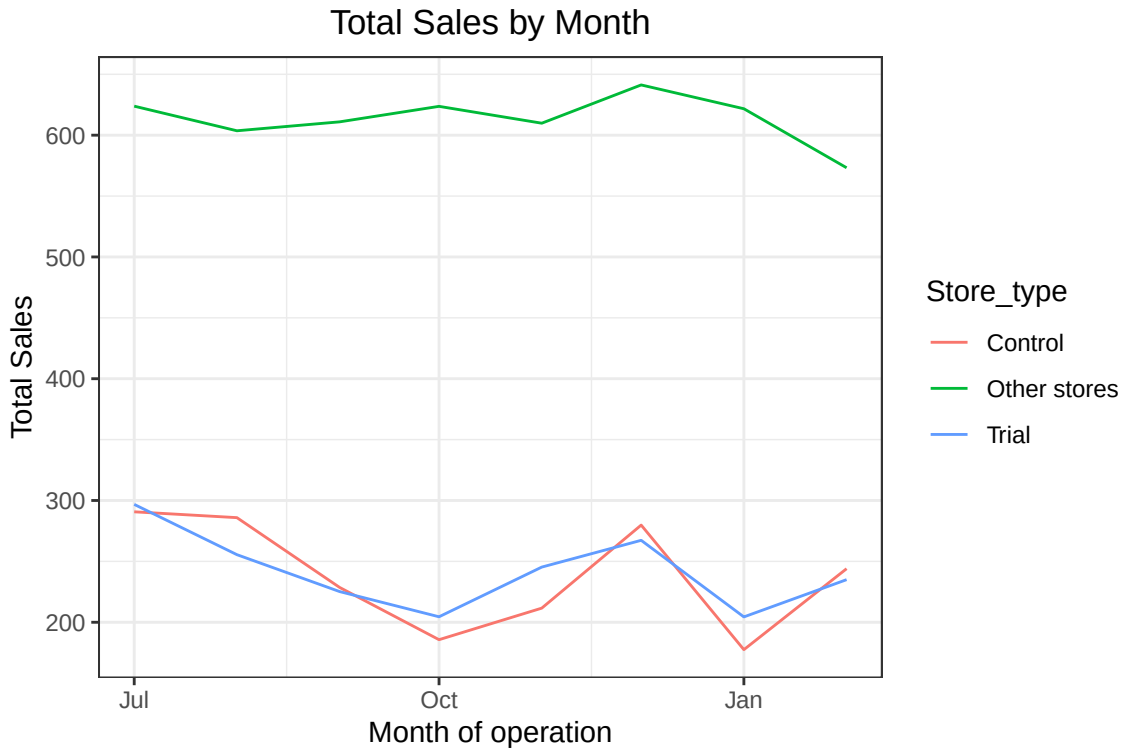
TRIAL STORE 77

```
# Use the function you created to calculate correlations against store 77
# using Total Sales and number of customers
trial_store_77 <- 77
corr_nSales <- calculateCorrelation(
  preTrialMeasures, quote(sales_total), trial_store_77
  )
corr_nCustomers <- calculateCorrelation(
  preTrialMeasures, quote(nCustomers), trial_store_77
  )

# Use the function for calculating magnitude.
magnitude_nSales <- calculateMagnitudeDistance(
  preTrialMeasures, quote(sales_total), trial_store_77
  )
magnitude_nCustomers <- calculateMagnitudeDistance(
  preTrialMeasures, quote(nCustomers), trial_store_77
  )

# Create a combined score composed of correlation and magnitude, by first merging
# the correlations table with the magnitude table
setDT(corr_nSales)
setDT(magnitude_nSales)
setDT(corr_nCustomers)
setDT(magnitude_nCustomers)

corr_weight <- 0.5

score_nSales <- merge(corr_nSales, magnitude_nSales, by = "Store2")[
  , scoreNSales := corr_measure * corr_weight + mag_measure * (1-corr_weight)] %>%
  arrange(desc(rowMeans(select(., starts_with(c("mag_", "cor_"))))))
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = "Store2")[
  , scoreNCust := corr_measure * corr_weight + mag_measure * (1-corr_weight)] %>%
  arrange(desc(rowMeans(select(., starts_with(c("mag_", "cor_"))))))

score_nSales$Store1.x <- NULL
score_nCustomers$Store1.x <- NULL
setnames(score_nSales, "Store1.y", "Store1")
setnames(score_nCustomers, "Store1.y", "Store1")

# Control stores based on the highest matching store for trial store 77
score_Control <- merge(score_nSales, score_nCustomers, by = c("Store1", "Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]
```

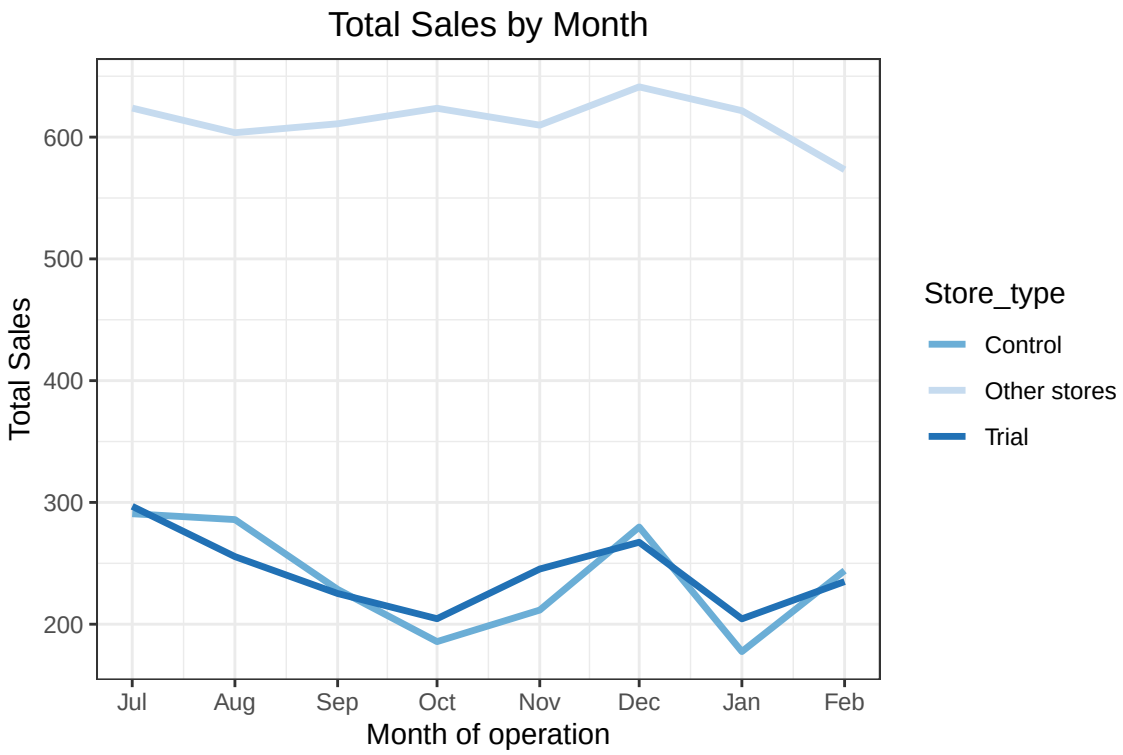
```
score_Control <-
  score_Control %>%
  arrange(desc(rowMeans(select(., starts_with(c("mag_","cor_"))))))

# Most appropriate control store for trial store 77 by finding the store with
# the highest final score
control_store <- score_Control$Store2[2]
control_store
```

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

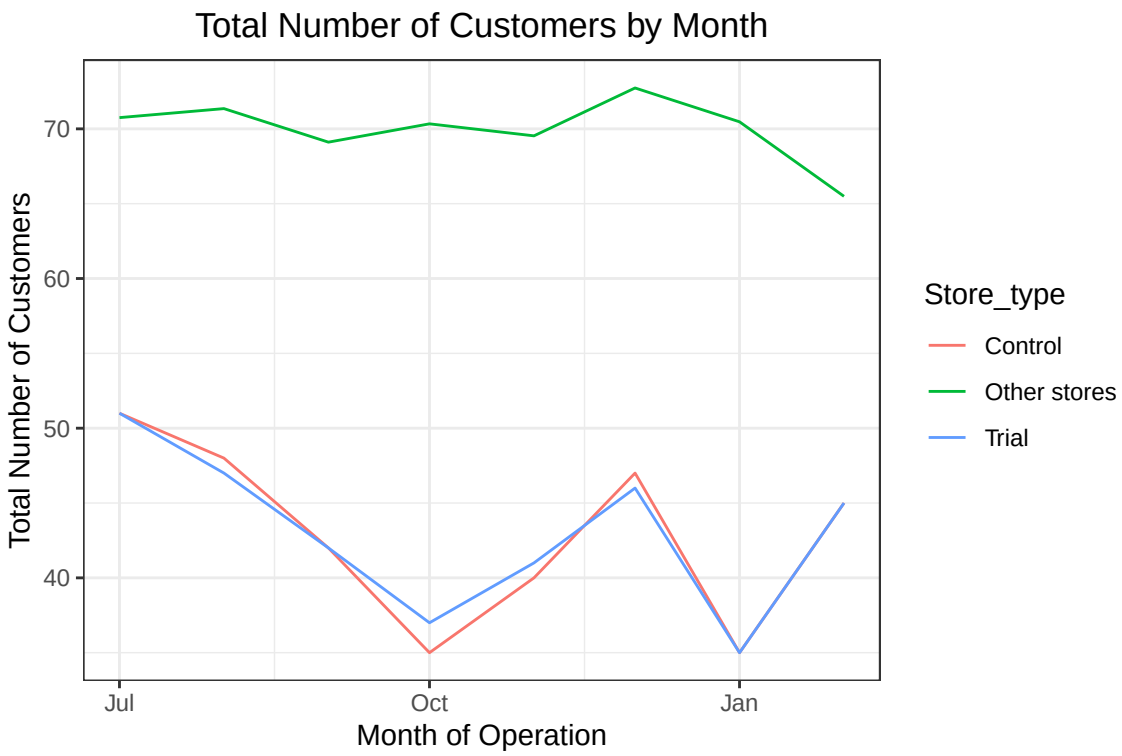
```
# Store 233 is the most appropriate control store for trial store 77 according
# to Total Sales

# Conduct visual checks on sales trends by comparing the trial store
# to the control store and other stores
measureOverTimeSales <- measureOverTime
setDT(measureOverTimeSales)
pastSales <- measureOverTimeSales[
  , Store_type := ifelse(STORE_NBR == trial_store_77, "Trial",
                         ifelse(STORE_NBR == control_store, "Control", "Other stores"))
][, totSales := mean(sales_total), by = c("monthID", "Store_type")
][, TransactionMonth := as.Date(paste(monthID %/% 100,
                                      monthID %% 100, 1, sep = "-"), "%Y-%m-%d")
][monthID < 201903 , ]
ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Total Sales", title = "Total Sales by Month")
```



```
# Plot for Presentation
ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line(size = 1.2) +
  labs(x = "Month of operation", y = "Total Sales",
       title = "Total Sales by Month") +
  scale_x_date(date_breaks = "1 month", date_labels = "%b") +
  scale_color_manual(values = c("Other stores" = "#c6dbef",    # light blue
                                "Control" = "#6baed6", # medium blue
                                "Trial" = "#2171b5")) # dark blue
```




```
# Conduct visual checks on customer count trends by comparing the trial store
# to the control store and other stores
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[
  , Store_type := ifelse(STORE_NBR == trial_store_77, "Trial",
                         ifelse(STORE_NBR == control_store,"Control", "Other stores"))
][, numCustomers := mean(nCustomers), by = c("monthID","Store_type")
][, TransactionMonth :=
    as.Date(paste(monthID %/% 100,
                  monthID %% 100, 1, sep = "-"), "%Y-%m-%d")
][monthID < 201903 , ]

ggplot(pastCustomers, aes(TransactionMonth, numCustomers, color = Store_type)) +
  geom_line() +
  labs(x = "Month of Operation", y = "Total Number of Customers",
       title = "Total Number of Customers by Month")
```




```
# Scale pre-trial control sales to match pre-trial trial store sales
scalingFactorForControlSales <- preTrialMeasures[
  STORE_NBR == trial_store_77 &
    monthID < 201902, sum(sales_total)]/preTrialMeasures[
      STORE_NBR == control_store & monthID < 201902, sum(sales_total)]

# Apply the scaling factor to control store sales
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store,
][ , controlSales := sales_total * scalingFactorForControlSales]

# Cleaning scaled trial and control sales Datasets - Removing unwanted columns
scaledControlSales$monthlyUnits <- NULL
scaledControlSales$avgPricePerUnit <- NULL
scaledControlSales$nTransactionPerCust <- NULL
scaledControlSales$totSales <- NULL
scaledControlSales$TransactionMonth <- NULL
scaledControlSales$numCustomers <- NULL

# Calculate the percentage difference between scaled control sales and trial sales
setDT(scaledControlSales)
percentageDiffSales <- merge(
  scaledControlSales[, c("monthID", "controlSales")],
  measureOverTime[STORE_NBR == trial_store_77, c("totSales", "monthID")],
  by = "monthID"
  )[, percentageDiff := (abs(controlSales - totSales)/(controlSales))]

# Take the standard deviation based on the scaled percentage difference
# in the pre-trial period
stdDevSales <- sd(percentageDiffSales[monthID < 201902 , percentageDiff])
```

```
# Note that there are 8 months in the pre-trial period
# hence 8 - 1 = 7 degrees of freedom
degreesOfFreedom <- 7

# Find the 95th percentile of the t distribution with the appropriate
# degrees of freedom to compare against
qt(0.95, df = degreesOfFreedom)
```

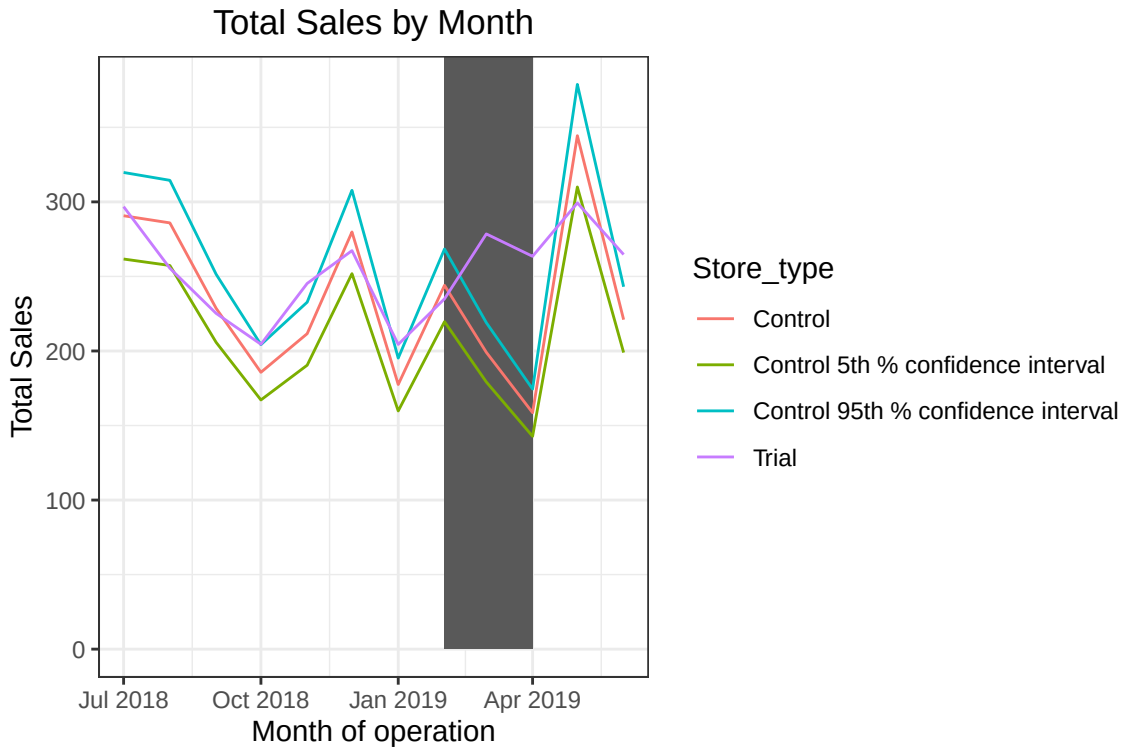
```
## [1] 1.894579
```

```
# test with a null hypothesis of there being 0 difference between trial and
# control stores
percentageDiffSales[, tValue := (percentageDiff - 0)/stdDevSales
                    ][, TransactionMonth := as.Date(paste(monthID %/% 100,
                                                          monthID %% 100, 1,
                                                          sep = "-"), "%Y-%m-%d")
                      ][monthID < 201905 & monthID > 201901,
                        .(TransactionMonth, tValue)]
```

```
##    TransactionMonth    tValue
## 1:       2019-02-01  1.183534
## 2:       2019-03-01  7.339116
## 3:       2019-04-01 12.476373
```

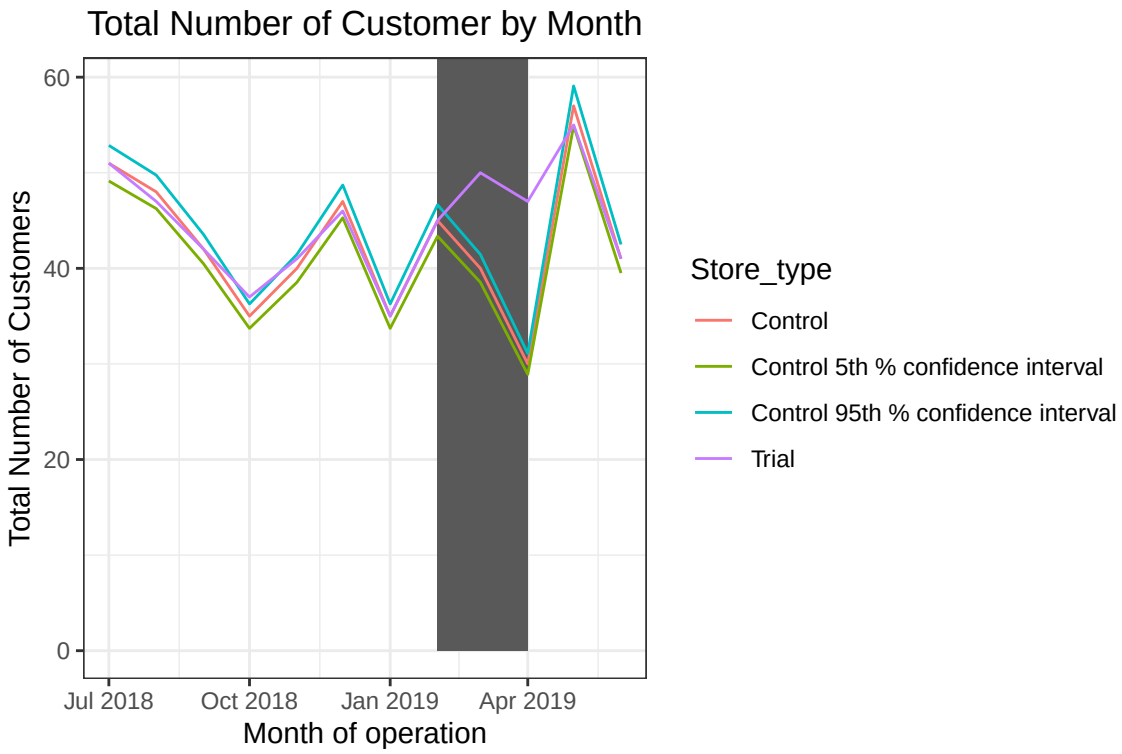
```
# We can observe that the t-value is much larger than the 95th percentile value
# of the t-distribution for March and April - i.e. the increase in sales in the
# trial store in March and April is statistically greater than in the control store.

#### Trial and control store Total Sales
pastSales <- measureOverTimeSales[
  , Store_type := ifelse(STORE_NBR == trial_store_77, "Trial",
                         ifelse(STORE_NBR == control_store,"Control", "Other stores"))
][, totSales := mean(sales_total), by = c("monthID","Store_type")
][, TransactionMonth := as.Date(paste(monthID %/% 100,
                                      monthID %% 100, 1, sep = "-"), "%Y-%m-%d")
][Store_type %in% c("Trial", "Control"), ]

## Control store 95th percentile
pastSales_Controls95 <- pastSales[Store_type == "Control",
                                  ][, totSales := sales_total * (1 + stdDevSales * 2)
                                  ][, Store_type := "Control 95th % confidence interval"]

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

trialAssessmentSales <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)

# Plotting these in a graph
ggplot(trialAssessmentSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_rect(data = trialAssessmentSales[ monthID < 201905 & monthID > 201901 ,],
```

```
            aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
                ymin = 0 , ymax = Inf, color = NULL), show.legend = FALSE) +
  geom_line() +
  labs(x = "Month of operation", y = "Total Sales", title = "Total Sales by Month")
```

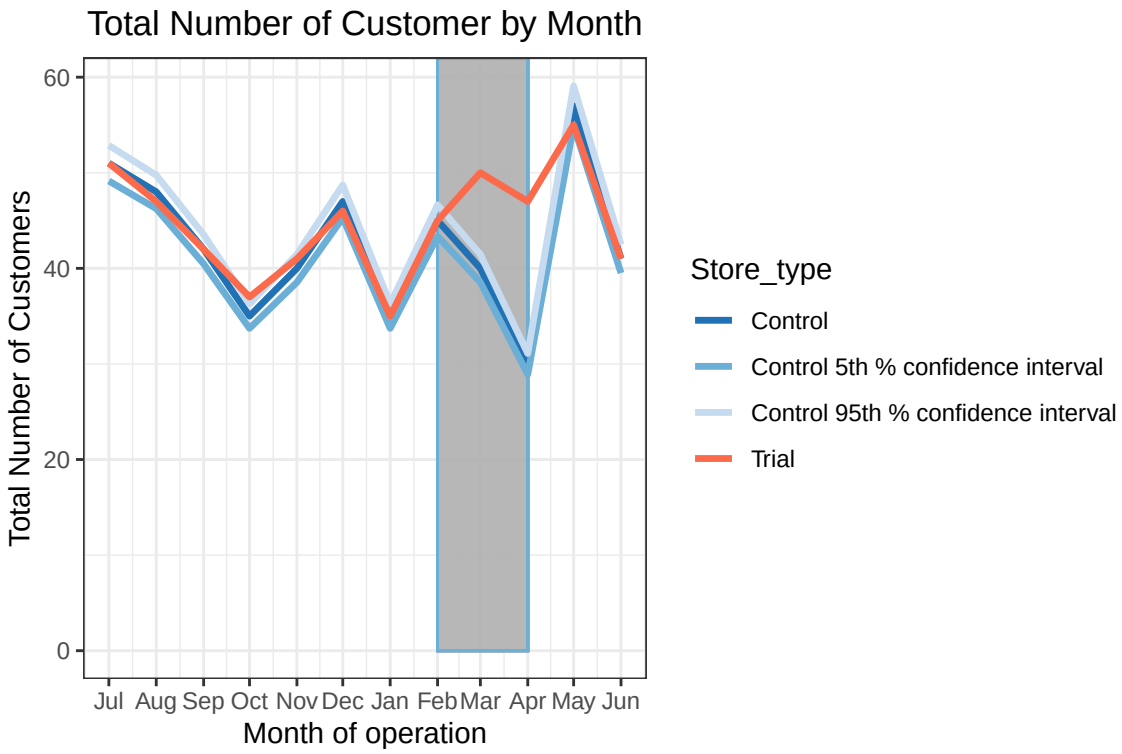



```
# The results show that the trial in store 77 is significantly different to its
# control store in the trial period as the trial store performance lies outside
# the 5% to 95% confidence interval of the control store in two of the three trial months.

# Scale pre-trial control number of customers  to match pre-trial trial
# store number of customers
scalingFactorForControlCust <- preTrialMeasures[
  STORE_NBR == trial_store_77 &
    monthID < 201902, sum(nCustomers)]/preTrialMeasures[
      STORE_NBR == control_store & monthID < 201902, sum(nCustomers)]
# Apply the scaling factor to control store sales
measureOverTimeCust <- measureOverTime
scaledControlCust <- measureOverTimeCust[
  STORE_NBR == control_store,
  ][ , controlCustomers := nCustomers * scalingFactorForControlCust
     ][, Store_type := ifelse(STORE_NBR == trial_store_77, "Trial",
                              ifelse(
                                STORE_NBR == control_store, "Control", "Other stores"))]

# Cleaning scaled trial and control sales Datasets - Removing unwanted columns
scaledControlCust$monthlyUnits <- NULL
scaledControlCust$avgPricePerUnit <- NULL
scaledControlCust$nTransactionPerCust <- NULL
```

```r
scaledControlCust$totSales <- NULL
scaledControlCust$TransactionMonth <- NULL
scaledControlCust$numCustomers <- NULL

# Calculate the percentage difference between scaled control sales and trial sales
setDT(scaledControlCust)
percentageDiffCust <- merge(scaledControlCust[
  , c("monthID", "controlCustomers")],
  measureOverTimeCust[STORE_NBR == trial_store_77, c("nCustomers", "monthID")],
  by = "monthID"
  )[, percentageDiff := (abs(controlCustomers - nCustomers)/(controlCustomers))]

# Take the standard deviation based on the scaled percentage difference in the
# pre-trial period
stdDevCust <- sd(percentageDiffCust[monthID < 201902 , percentageDiff])

# Note that there are 8 months in the pre-trial period
# hence 8 - 1 = 7 degrees of freedom
degreesOfFreedom <- 7

#### Trial and control store Total Sales
pastCustomers <- measureOverTimeCusts[, nCusts := mean(nCustomers),
                                      by = c("monthID", "Store_type")
                                      ][Store_type %in% c("Trial", "Control"), ]

## Control store 95th percentile
pastCust_Controls95 <- pastCustomers[Store_type == "Control",
                                     ][, nCusts := nCusts * (1 + stdDevCust * 2)
                                       ][, Store_type := "Control 95th % confidence interval"]

# Control store 5th percentile
pastCust_Controls5 <- pastCustomers[Store_type == "Control",
                                      ][, nCusts := nCusts * (1 - stdDevCust * 2)
                                        ][, Store_type := "Control 5th % confidence interval"]

trialAssessmentCust <- rbind(pastCustomers, pastCust_Controls95, pastCust_Controls5)

# Plotting these in one nice graph
ggplot(trialAssessmentCust, aes(TransactionMonth, nCusts, color = Store_type)) +
  geom_rect(data = trialAssessmentCust[ monthID < 201905 & monthID > 201901 ,],
            aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
                ymin = 0 , ymax = Inf, color = NULL), show.legend = FALSE) +
  geom_line() +
  labs(x = "Month of operation", y = "Total Number of Customers",
       title = "Total Number of Customer by Month")
```




```
# Plot for Presentation
ggplot(trialAssessmentCust, aes(TransactionMonth, nCusts, color = Store_type)) +
  geom_rect(data = trialAssessmentCust[ monthID < 201905 & monthID > 201901 ,],
            aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
                ymin = 0 , ymax = Inf),
            fill = "grey70", alpha = 0.2, show.legend = FALSE) +
  geom_line(size = 1.2) +
  labs(x = "Month of operation", y = "Total Number of Customers",
       title = "Total Number of Customer by Month") +
  scale_x_date(date_breaks = "1 month", date_labels = "%b") +
  scale_color_manual(values =
                       c("Control 95th % confidence interval" = "#c6dbef", # light blue
                         "Control 5th % confidence interval" = "#6baed6", # medium blue
                         "Control" = "#2171b5", # dark blue
                         "Trial" = "#fb6a4a")) # reddish
```




```
# The results show that the trial in store 77 is significantly different to its
# control store in the trial period as the trial store performance lies outside
# the 5% to 95% confidence interval of the control store in two of the three trial months.
```

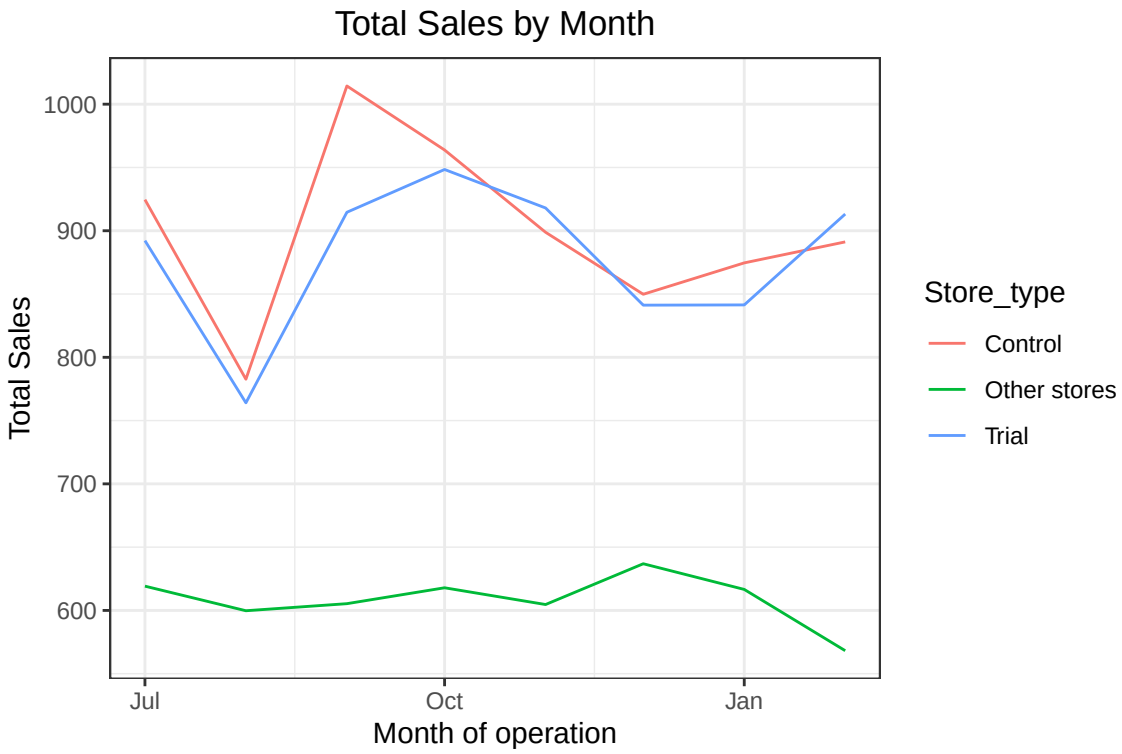
```
## TRIAL STORE 86
```

```
# Use the function you created to calculate correlations against store 86
# using Total Sales and number of customers
trial_store_86 <- 86
corr_nSales <- calculateCorrelation(
  preTrialMeasures, quote(sales_total), trial_store_86
  )
corr_nCustomers <- calculateCorrelation(
  preTrialMeasures, quote(nCustomers), trial_store_86
  )

# Then, use the functions for calculating magnitude.
magnitude_nSales <- calculateMagnitudeDistance(
  preTrialMeasures, quote(sales_total), trial_store_86
  )
magnitude_nCustomers <- calculateMagnitudeDistance(
  preTrialMeasures, quote(nCustomers), trial_store_86
  )

# Create a combined score composed of correlation and magnitude, by first merging
# the correlations table with the magnitude table
setDT(corr_nSales)
setDT(magnitude_nSales)
```

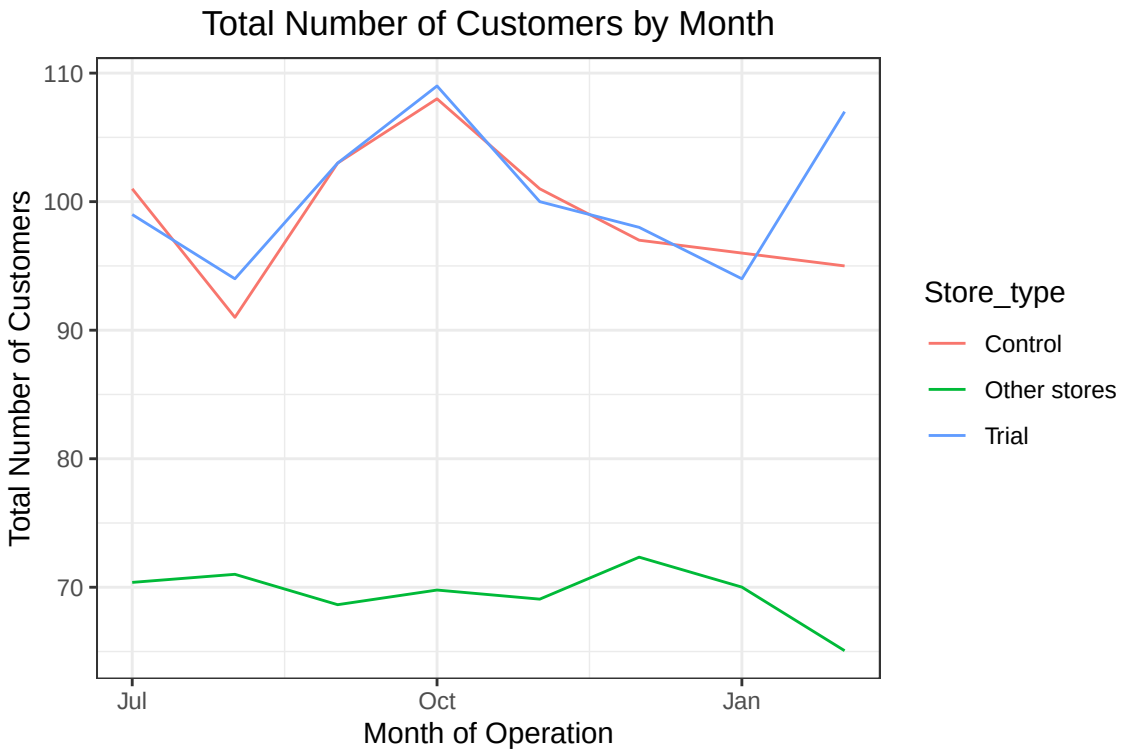
```r
setDT(corr_nCustomers)
setDT(magnitude_nCustomers)

corr_weight <- 0.5

score_nSales <- merge(corr_nSales, magnitude_nSales, by = "Store2")[
  , scoreNSales := corr_measure * corr_weight + mag_measure * (1-corr_weight)] %>%
  arrange(desc(rowMeans(select(., starts_with(c("mag_", "cor_"))))))
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = "Store2")[
  , scoreNCust := corr_measure * corr_weight + mag_measure * (1-corr_weight)] %>%
  arrange(desc(rowMeans(select(., starts_with(c("mag_", "cor_"))))))

score_nSales$Store1.x <- NULL
score_nCustomers$Store1.x <- NULL
setnames(score_nSales, "Store1.y", "Store1")
setnames(score_nCustomers, "Store1.y", "Store1")

# Control stores based on the highest matching store for trial store 86
score_Control <- merge(score_nSales, score_nCustomers, by = c("Store1", "Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]

score_Control <-
  score_Control %>%
  arrange(desc(rowMeans(select(., starts_with(c("mag_","cor_"))))))

# Most appropriate control store for trial store 86 by finding the store with
# the highest final score
control_store <- score_Control$Store2[2]
control_store
```

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

```r
# Store 155 is the most appropriate control store for trial store 86 according
# to Total Sales

# Conduct visual checks on sales trends by comparing the trial store
# to the control store and other stores
measureOverTimeSales <- measureOverTime
setDT(measureOverTimeSales)
pastSales <- measureOverTimeSales[
    , Store_type := ifelse(STORE_NBR == trial_store_86, "Trial",
                           ifelse(STORE_NBR == control_store, "Control", "Other stores"))
  ][, totSales := mean(sales_total), by = c("monthID", "Store_type")
  ][, TransactionMonth := as.Date(paste(monthID %/% 100,
                                        monthID %% 100, 1, sep = "-"), "%Y-%m-%d")
  ][monthID < 201903 , ]
ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Total Sales", title = "Total Sales by Month")
```




```
# Conduct visual checks on customer count trends by comparing the trial store
# to the control store and other stores
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[
  , Store_type := ifelse(STORE_NBR == trial_store_86, "Trial",
                         ifelse(STORE_NBR == control_store, "Control", "Other stores"))
][, numCustomers := mean(nCustomers), by = c("monthID", "Store_type")
][, TransactionMonth := as.Date(paste(monthID %/% 100,
                                      monthID %% 100, 1, sep = "-"), "%Y-%m-%d")
][monthID < 201903 , ]
ggplot(pastCustomers, aes(TransactionMonth, numCustomers, color = Store_type)) +
  geom_line() +
  labs(x = "Month of Operation", y = "Total Number of Customers", title = "Total Number of Customers by
```




```
# Scale pre-trial control sales to match pre-trial trial store sales
scalingFactorForControlSales <- preTrialMeasures[
  STORE_NBR == trial_store_86 & monthID < 201902, sum(sales_total)]/preTrialMeasures[
    STORE_NBR == control_store & monthID < 201902, sum(sales_total)]
# Apply the scaling factor to control store sales
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store,
][ , controlSales := sales_total * scalingFactorForControlSales]

# Cleaning scaled trial and control sales Datasets - Removing unwanted columns
scaledControlSales$monthlyUnits <- NULL
scaledControlSales$avgPricePerUnit <- NULL
scaledControlSales$nTransactionPerCust <- NULL
scaledControlSales$totSales <- NULL
scaledControlSales$TransactionMonth <- NULL
scaledControlSales$numCustomers <- NULL

# Calculate the percentage difference between scaled control sales and trial sales
setDT(scaledControlSales)
percentageDiffSales <- merge(scaledControlSales[
  , c("monthID", "controlSales")], measureOverTime[
    STORE_NBR == trial_store_86, c("totSales", "monthID")], by = "monthID"
)[, percentageDiff := (abs(controlSales - totSales)/(controlSales))]

# Take the standard deviation based on the scaled percentage difference in the
# pre-trial period
stdDevSales <- sd(percentageDiffSales[monthID < 201902 , percentageDiff])

# Note that there are 8 months in the pre-trial period
# hence 8 - 1 = 7 degrees of freedom
```

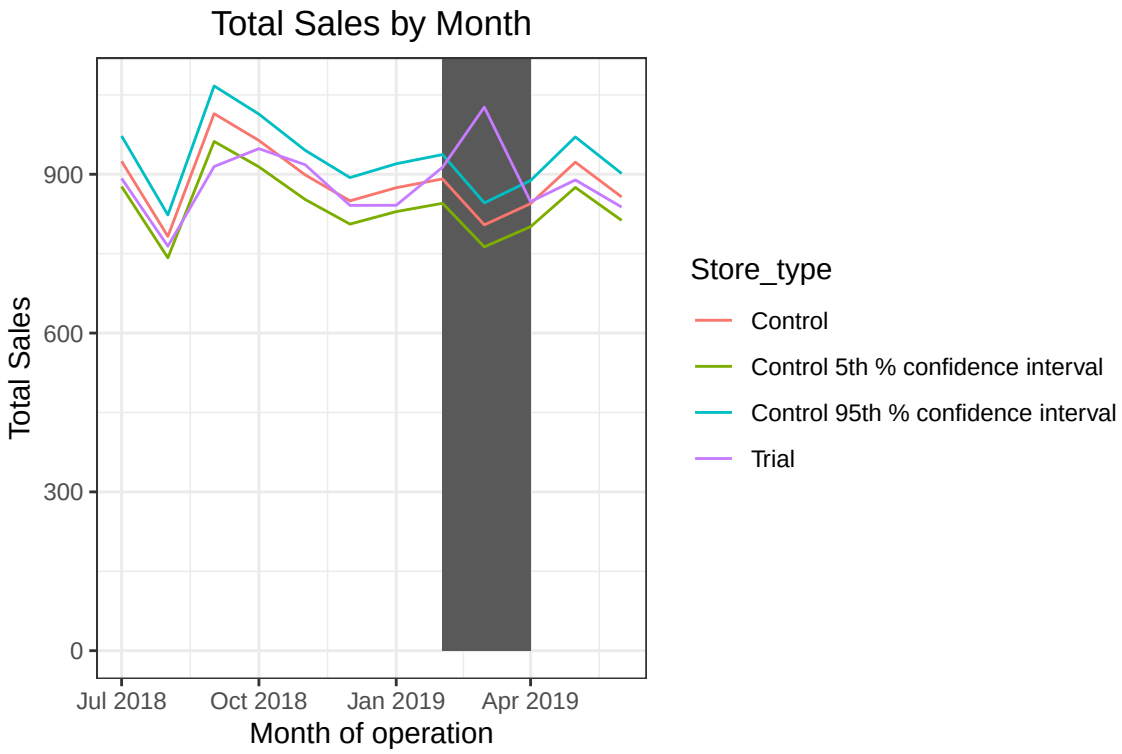
```r
degreesOfFreedom <- 7

# Find the 95th percentile of the t distribution with the appropriate
# degrees of freedom to compare against
qt(0.95, df = degreesOfFreedom)
```

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

```r
# test with a null hypothesis of there being 0 difference between trial and
# control stores
percentageDiffSales[, tValue := (percentageDiff - 0)/stdDevSales
][, TransactionMonth := as.Date(paste(monthID %/% 100,
                                      monthID %% 100, 1, sep = "-"), "%Y-%m-%d")
][monthID < 201905 & monthID > 201901, .(TransactionMonth, tValue)]
```

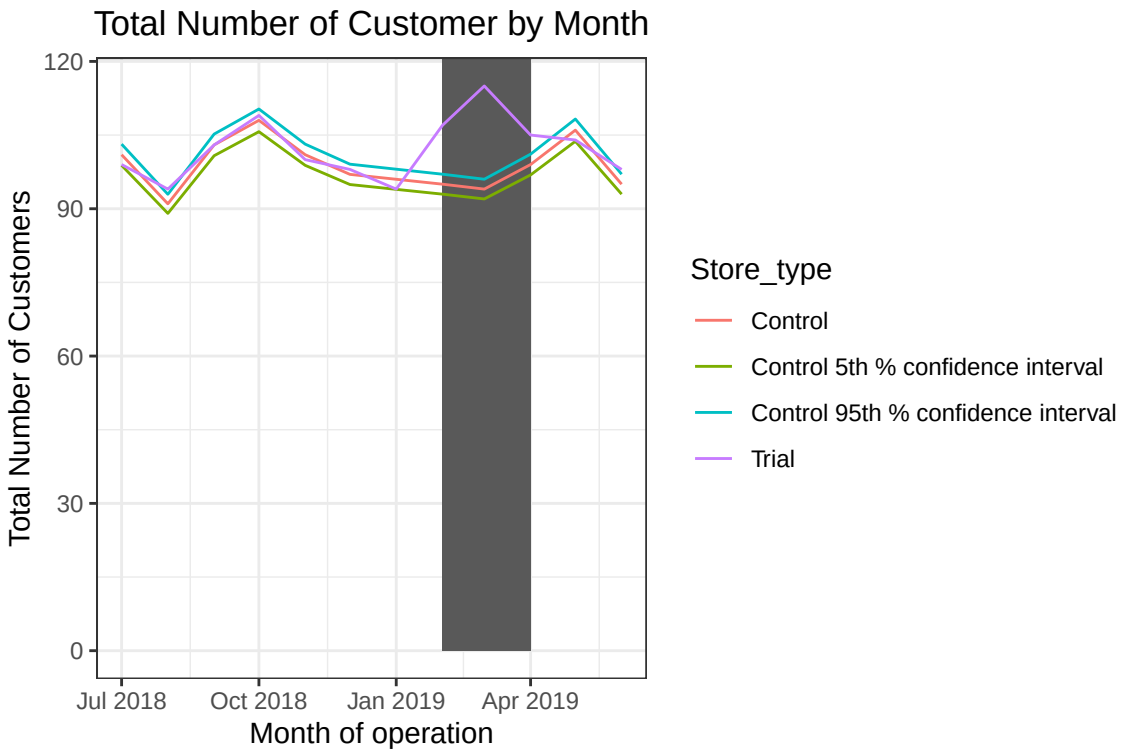
```
##    TransactionMonth    tValue
## 1:       2019-02-01  2.179542
## 2:       2019-03-01 12.226922
## 3:       2019-04-01  1.364580
```

```r
# 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.

#### Trial and control store Total Sales
pastSales <- measureOverTimeSales[
  , Store_type := ifelse(STORE_NBR == trial_store_86, "Trial",
                         ifelse(STORE_NBR == control_store,"Control", "Other stores"))
][, totSales := mean(sales_total), by = c("monthID","Store_type")
][, TransactionMonth :=
    as.Date(paste(monthID %/% 100,
                  monthID %% 100, 1, sep = "-"), "%Y-%m-%d")
][Store_type %in% c("Trial", "Control"), ]

## Control store 95th percentile
pastSales_Controls95 <- pastSales[Store_type == "Control",
][, totSales := sales_total * (1 + stdDevSales * 2)
][, Store_type := "Control 95th % confidence interval"]

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

trialAssessmentSales <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)

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




```
# The results show that the trial in store 86 is not significantly different to its
# control store in the trial period as the trial store performance lies outside
# the 5% to 95% confidence interval of the control store in two of the three trial months.

# Scale pre-trial control number of customers  to match pre-trial trial
# store number of customers
scalingFactorForControlCust <- preTrialMeasures[
  STORE_NBR == trial_store_86 & monthID < 201902, sum(nCustomers)]/preTrialMeasures[
    STORE_NBR == control_store & monthID < 201902, sum(nCustomers)]
# Apply the scaling factor to control store sales
measureOverTimeCust <- measureOverTime
scaledControlCust <- measureOverTimeCust[STORE_NBR == control_store,
][ , controlCustomers := nCustomers * scalingFactorForControlCust
][, Store_type := ifelse(STORE_NBR == trial_store_86, "Trial",
                         ifelse(STORE_NBR == control_store, "Control", "Other stores"))]

# Cleaning scaled trial and control sales Datasets - Removing unwanted columns
scaledControlCust$monthlyUnits <- NULL
scaledControlCust$avgPricePerUnit <- NULL
scaledControlCust$nTransactionPerCust <- NULL
scaledControlCust$totSales <- NULL
scaledControlCust$TransactionMonth <- NULL
scaledControlCust$numCustomers <- NULL

# Calculate the percentage difference between scaled control sales and trial sales
setDT(scaledControlCust)
percentageDiffCust <- merge(scaledControlCust[
  , c("monthID", "controlCustomers")],
  measureOverTimeCust[STORE_NBR == trial_store_86, c("nCustomers", "monthID")],
```

```
  by = "monthID"
)[, percentageDiff := (abs(controlCustomers - nCustomers)/(controlCustomers))]

# Take the standard deviation based on the scaled percentage difference in the
# pre-trial period
stdDevCust <- sd(percentageDiffCust[monthID < 201902 , percentageDiff])

# Note that there are 8 months in the pre-trial period
# hence 8 - 1 = 7 degrees of freedom
degreesOfFreedom <- 7

#### Trial and control store Total Sales
pastCustomers <- measureOverTimeCusts[, nCusts := mean(nCustomers),
                                      by = c("monthID", "Store_type")
][Store_type %in% c("Trial", "Control"), ]

## Control store 95th percentile
pastCust_Controls95 <- pastCustomers[Store_type == "Control",
][, nCusts := nCusts * (1 + stdDevCust * 2)
][, Store_type := "Control 95th % confidence interval"]

# Control store 5th percentile
pastCust_Controls5 <- pastCustomers[Store_type == "Control",
][, nCusts := nCusts * (1 - stdDevCust * 2)
][, Store_type := "Control 5th % confidence interval"]

trialAssessmentCust <- rbind(pastCustomers, pastCust_Controls95, pastCust_Controls5)

# Plotting these in one nice graph
ggplot(trialAssessmentCust, aes(TransactionMonth, nCusts, color = Store_type)) +
  geom_rect(data = trialAssessmentCust[ monthID < 201905 & monthID > 201901 ,],
            aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
                ymin = 0 , ymax = Inf, color = NULL), show.legend = FALSE) +
  geom_line() +
  labs(x = "Month of operation", y = "Total Number of Customers",
       title = "Total Number of Customer by Month")
```




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

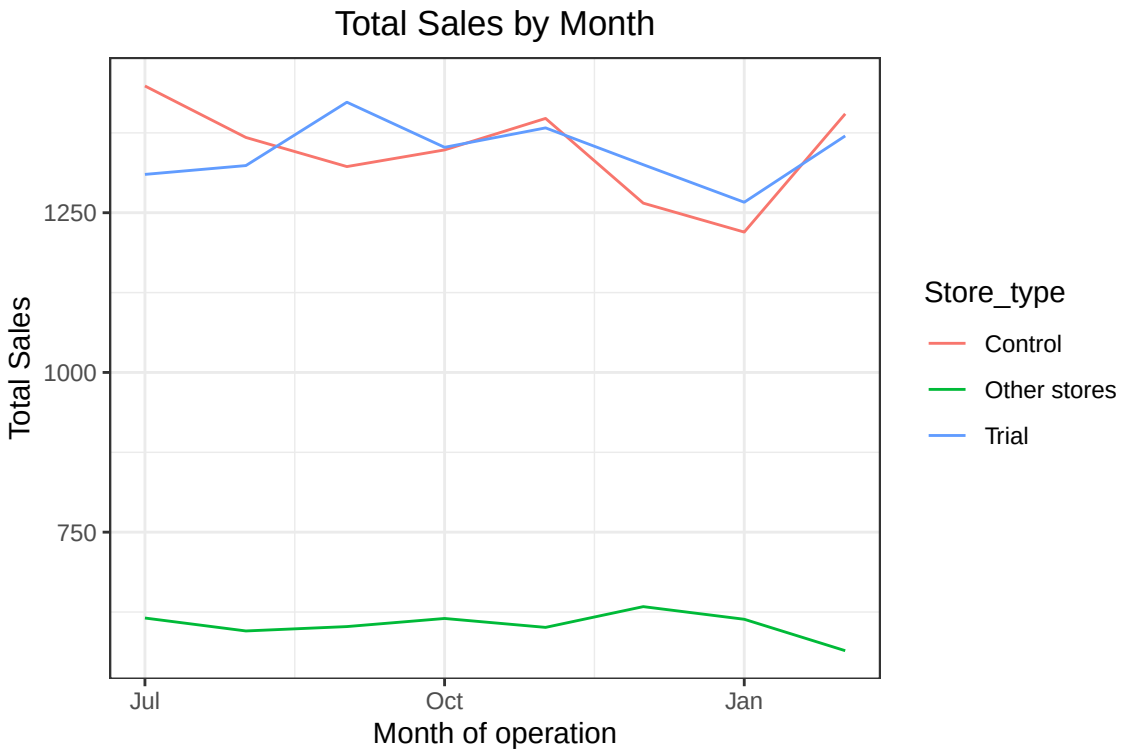
```
## TRIAL STORE 88
```

```
# Use the function you created to calculate correlations against store 88
# using Total Sales and number of customers
trial_store_88 <- 88
corr_nSales <- calculateCorrelation(
  preTrialMeasures, quote(sales_total), trial_store_88
  )
corr_nCustomers <- calculateCorrelation(
  preTrialMeasures, quote(nCustomers), trial_store_88
  )

# Then, use the functions for calculating magnitude.
magnitude_nSales <- calculateMagnitudeDistance(
  preTrialMeasures, quote(sales_total), trial_store_88
  )
magnitude_nCustomers <- calculateMagnitudeDistance(
  preTrialMeasures, quote(nCustomers), trial_store_88
  )

# Create a combined score composed of correlation and magnitude, by first merging
```

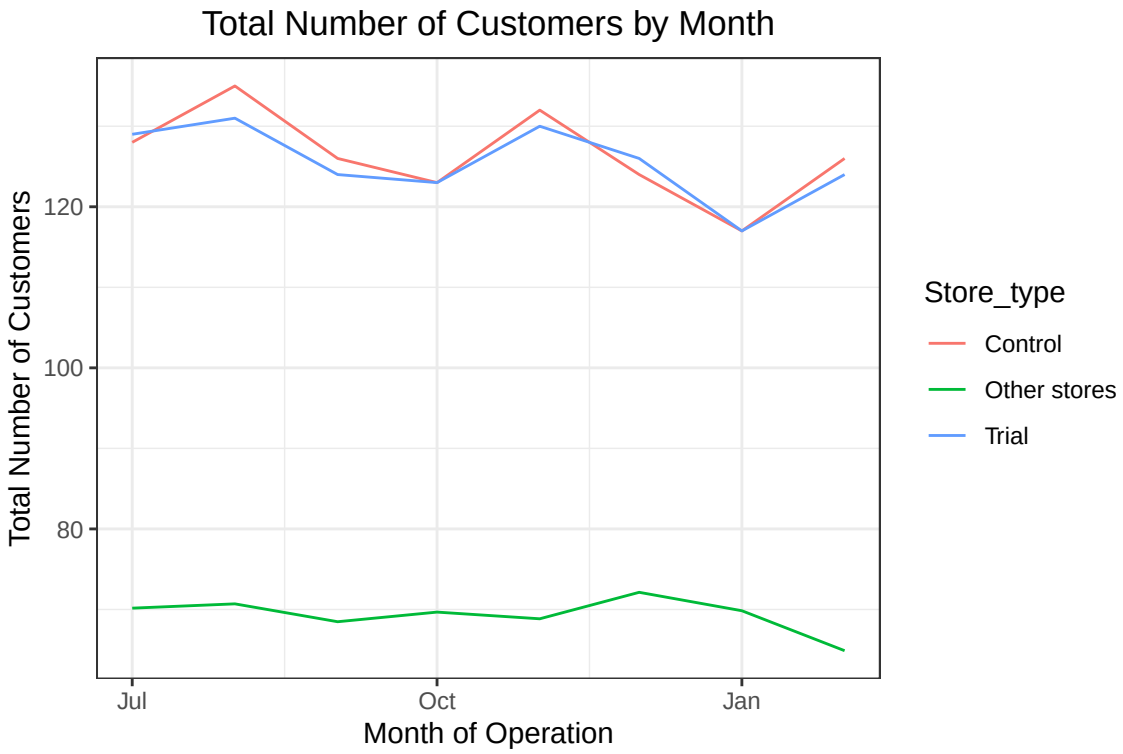
```r
# the correlations table with the magnitude table
setDT(corr_nSales)
setDT(magnitude_nSales)
setDT(corr_nCustomers)
setDT(magnitude_nCustomers)

corr_weight <- 0.5

score_nSales <- merge(corr_nSales, magnitude_nSales, by = "Store2")[
  , scoreNSales := corr_measure * corr_weight + mag_measure * (1-corr_weight)] %>%
  arrange(desc(rowMeans(select(., starts_with(c("mag_", "cor_"))))))
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = "Store2")[
  , scoreNCust := corr_measure * corr_weight + mag_measure * (1-corr_weight)] %>%
  arrange(desc(rowMeans(select(., starts_with(c("mag_", "cor_"))))))

score_nSales$Store1.x <- NULL
score_nCustomers$Store1.x <- NULL
setnames(score_nSales, "Store1.y", "Store1")
setnames(score_nCustomers, "Store1.y", "Store1")

# Control stores based on the highest matching store for trial store 88
score_Control <- merge(score_nSales, score_nCustomers, by = c("Store1", "Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]

score_Control <-
  score_Control %>%
  arrange(desc(rowMeans(select(., starts_with(c("mag_","cor_"))))))

# Most appropriate control store for trial store 86 by finding the store with
# the highest final score
control_store <- score_Control$Store2[2]
control_store
```

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

```r
# Store 237 is the most appropriate control store for trial store 88 according
# to Total Sales

# Conduct visual checks on sales trends by comparing the trial store
# to the control store and other stores
measureOverTimeSales <- measureOverTime
setDT(measureOverTimeSales)
pastSales <-
  measureOverTimeSales[
    , Store_type := ifelse(STORE_NBR == trial_store_88, "Trial",
                           ifelse(STORE_NBR == control_store, "Control", "Other stores"))
  ][, totSales := mean(sales_total), by = c("monthID", "Store_type")
  ][, TransactionMonth := as.Date(paste(monthID %/% 100,
                                        monthID %% 100, 1, sep = "-"), "%Y-%m-%d")
  ][monthID < 201903 , ]
ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Total Sales", title = "Total Sales by Month")
```




```
# Conduct visual checks on customer count trends by comparing the trial store
# to the control store and other stores
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[
  , Store_type := ifelse(STORE_NBR == trial_store_88, "Trial",
                         ifelse(STORE_NBR == control_store, "Control", "Other stores"))
][, numCustomers := mean(nCustomers), by = c("monthID", "Store_type")
][, TransactionMonth := as.Date(paste(monthID %/% 100,
                                      monthID %% 100, 1, sep = "-"), "%Y-%m-%d")
][monthID < 201903 , ]
ggplot(pastCustomers, aes(TransactionMonth, numCustomers, color = Store_type)) +
  geom_line() +
  labs(x = "Month of Operation", y = "Total Number of Customers",
       title = "Total Number of Customers by Month")
```




```
# Scale pre-trial control sales to match pre-trial trial store sales
scalingFactorForControlSales <- preTrialMeasures[
  STORE_NBR == trial_store_88 & monthID < 201902, sum(sales_total)]/preTrialMeasures[
    STORE_NBR == control_store & monthID < 201902, sum(sales_total)]
# Apply the scaling factor to control store sales
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store,
][ , controlSales := sales_total * scalingFactorForControlSales]

# Cleaning scaled trial and control sales Datasets - Removing unwanted columns
scaledControlSales$monthlyUnits <- NULL
scaledControlSales$avgPricePerUnit <- NULL
scaledControlSales$nTransactionPerCust <- NULL
scaledControlSales$totSales <- NULL
scaledControlSales$TransactionMonth <- NULL
scaledControlSales$numCustomers <- NULL

# Calculate the percentage difference between scaled control sales and trial sales
setDT(scaledControlSales)
percentageDiffSales <- merge(scaledControlSales[
  , c("monthID", "controlSales")], measureOverTime[
    STORE_NBR == trial_store_88, c("totSales", "monthID")], by = "monthID"
)[, percentageDiff := (abs(controlSales - totSales)/(controlSales))]

# Take the standard deviation based on the scaled percentage difference in the
# pre-trial period
stdDevSales <- sd(percentageDiffSales[monthID < 201902 , percentageDiff])

# Note that there are 8 months in the pre-trial period
# hence 8 - 1 = 7 degrees of freedom
```

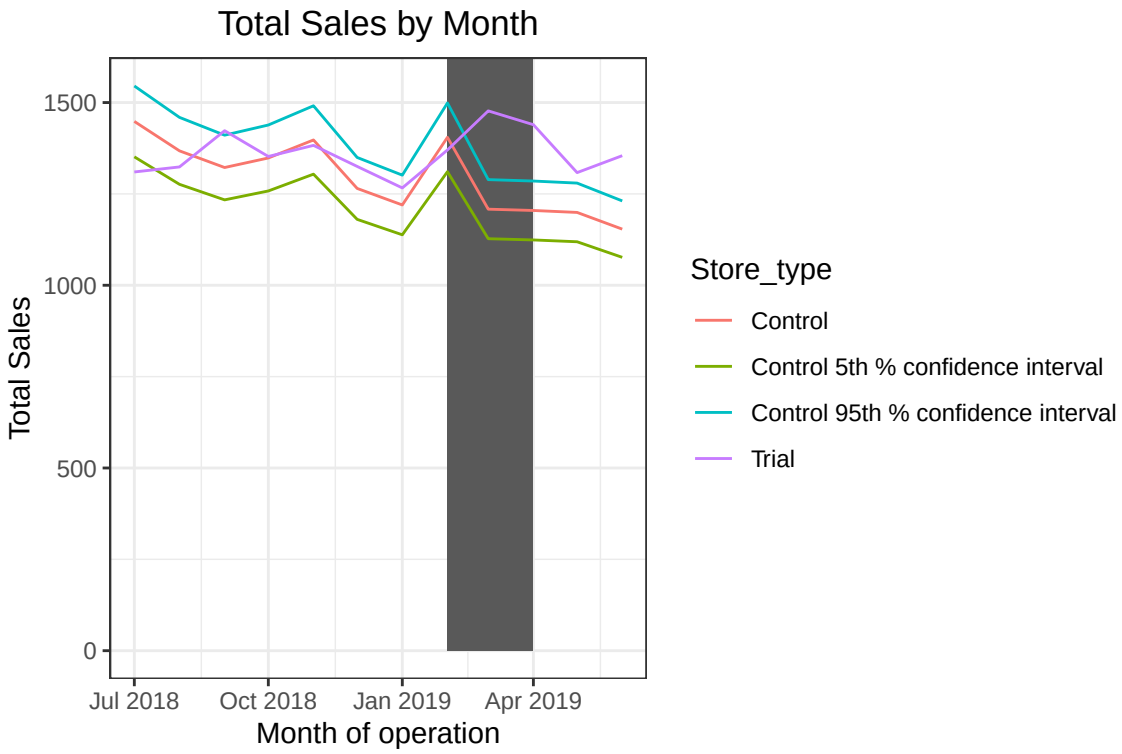
```
degreesOfFreedom <- 7

# Find the 95th percentile of the t distribution with the appropriate
# degrees of freedom to compare against
qt(0.95, df = degreesOfFreedom)
```

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

```
# test with a null hypothesis of there being 0 difference between trial and
# control stores
percentageDiffSales[, tValue := (percentageDiff - 0)/stdDevSales
][, TransactionMonth := as.Date(paste(monthID %/% 100,
                                      monthID %% 100, 1, sep = "-"), "%Y-%m-%d")
][monthID < 201905 & monthID > 201901, .(TransactionMonth, tValue)]
```

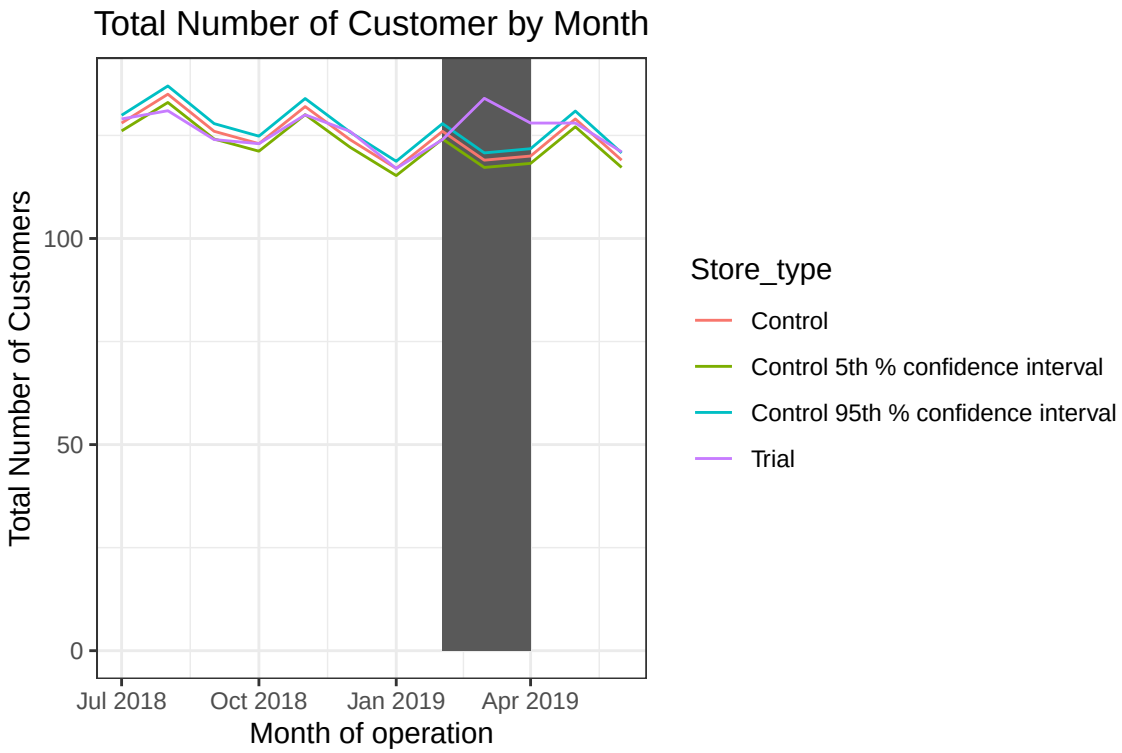
```
##    TransactionMonth    tValue
## 1:       2019-02-01 0.7812695
## 2:       2019-03-01 6.5956678
## 3:       2019-04-01 5.7685269
```

```
# 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.

#### Trial and control store Total Sales
pastSales <- measureOverTimeSales[
  , Store_type := ifelse(STORE_NBR == trial_store_88, "Trial",
                         ifelse(STORE_NBR == control_store,"Control", "Other stores"))
][, totSales := mean(sales_total), by = c("monthID","Store_type")
][, TransactionMonth :=
    as.Date(paste(monthID %/% 100,
                  monthID %% 100, 1, sep = "-"), "%Y-%m-%d")
][Store_type %in% c("Trial", "Control"), ]

## Control store 95th percentile
pastSales_Controls95 <- pastSales[Store_type == "Control",
][, totSales := sales_total * (1 + stdDevSales * 2)
][, Store_type := "Control 95th % confidence interval"]

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

trialAssessmentSales <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)

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




```
# The results show that the trial in store 88 is significantly different to its
# control store in the trial period as the trial store performance lies outside
# the 5% to 95% confidence interval of the control store in two of the three trial months.

# Scale pre-trial control number of customers  to match pre-trial trial
# store number of customers
scalingFactorForControlCust <- preTrialMeasures[
  STORE_NBR == trial_store_88 & monthID < 201902, sum(nCustomers)]/preTrialMeasures[
    STORE_NBR == control_store & monthID < 201902, sum(nCustomers)]
# Apply the scaling factor to control store sales
measureOverTimeCust <- measureOverTime
scaledControlCust <- measureOverTimeCust[STORE_NBR == control_store,
][ , controlCustomers := nCustomers * scalingFactorForControlCust
][, Store_type := ifelse(STORE_NBR == trial_store_88, "Trial",
                         ifelse(STORE_NBR == control_store, "Control", "Other stores"))]

# Cleaning scaled trial and control sales Datasets - Removing unwanted columns
scaledControlCust$monthlyUnits <- NULL
scaledControlCust$avgPricePerUnit <- NULL
scaledControlCust$nTransactionPerCust <- NULL
scaledControlCust$totSales <- NULL
scaledControlCust$TransactionMonth <- NULL
scaledControlCust$numCustomers <- NULL

# Calculate the percentage difference between scaled control sales and trial sales
setDT(scaledControlCust)
percentageDiffCust <- merge(scaledControlCust[
  , c("monthID", "controlCustomers")], measureOverTimeCust[
    STORE_NBR == trial_store_88, c("nCustomers", "monthID")], by = "monthID"
```

```
)[, percentageDiff := (abs(controlCustomers - nCustomers)/(controlCustomers))]

# Take the standard deviation based on the scaled percentage difference in the
# pre-trial period
stdDevCust <- sd(percentageDiffCust[monthID < 201902 , percentageDiff])

# Note that there are 8 months in the pre-trial period
# hence 8 - 1 = 7 degrees of freedom
degreesOfFreedom <- 7

#### Trial and control store Total Sales
pastCustomers <- measureOverTimeCusts[, nCusts := mean(nCustomers),
                                      by = c("monthID", "Store_type")
][Store_type %in% c("Trial", "Control"), ]

## Control store 95th percentile
pastCust_Controls95 <- pastCustomers[Store_type == "Control",
][, nCusts := nCusts * (1 + stdDevCust * 2)
][, Store_type := "Control 95th % confidence interval"]

# Control store 5th percentile
pastCust_Controls5 <- pastCustomers[Store_type == "Control",
][, nCusts := nCusts * (1 - stdDevCust * 2)
][, Store_type := "Control 5th % confidence interval"]

trialAssessmentCust <- rbind(pastCustomers, pastCust_Controls95, pastCust_Controls5)

# Plotting these in one nice graph
ggplot(trialAssessmentCust, aes(TransactionMonth, nCusts, color = Store_type)) +
  geom_rect(data = trialAssessmentCust[ monthID < 201905 & monthID > 201901 ,],
            aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
                ymin = 0 , ymax = Inf, color = NULL), show.legend = FALSE) +
  geom_line() +
  labs(x = "Month of operation", y = "Total Number of Customers",
       title = "Total Number of Customer by Month")
```




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

# We've found control stores 233, 155, 237 for trial stores 77, 86 and 88 respectively.

# The results for trial stores 77 and 88 during the trial period show a significant
# difference in at least two of the three trial months but this is not the case
# for trial store 86. We can check with the client if the implementation of the
# trial was different in trial store 86 but overall, the trial shows a significant
# increase in sales.
```