task2_solution.R

User

2025-06-16

```r
#### --- Step 1: Load required libraries and datasets ---
options(repos = c(CRAN = "https://cran.rstudio.com/"))

# Install packages
install.packages("data.table")
```

```
## Installing package into 'C:/Users/User/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)

## package 'data.table' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'data.table'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\User\AppData\Local\R\win-library\4.4\00LOCK\data.table\libs\x64\data_table.dll
## to
## C:\Users\User\AppData\Local\R\win-library\4.4\data.table\libs\x64\data_table.dll:
## Permission denied

## Warning: restored 'data.table'

##
## The downloaded binary packages are in
##  C:\Users\User\AppData\Local\Temp\RtmpUT2Mv3\downloaded_packages
```

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

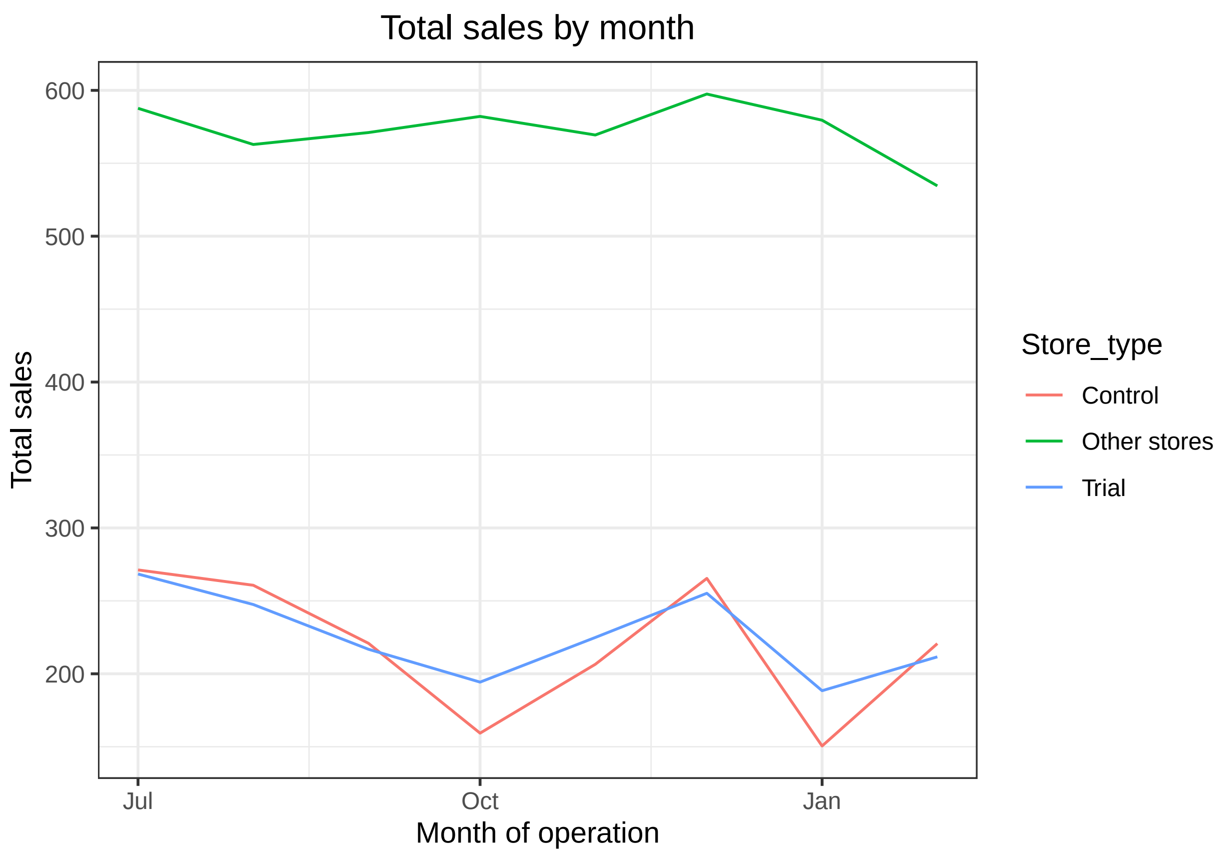
```
## Warning: package 'data.table' was built under R version 4.4.3
```

```r
library(ggplot2)
library(tidyr)

# Point the filePath to where you have downloaded the datasets to and
# assign the data files to data.tables
filePath <- "C:/Users/User/OneDrive - Swinburne University/Desktop/forage/quantium"
data <- fread(file.path(filePath,"QVI_data.csv"))

# Set themes for plots
```

```
theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = 0.5))

#### --- Step 2: Select control stores ---

# Client selected stores 77, 86, and 88 as trial locations. Establish control stores
# that were operational throughout the observation period and match pre-February 2019
# performance in: monthly sales revenue, customer count, and transactions per customer

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

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

# (2) Define the measure calculations
measureOverTime <- data[, .(totSales = sum(TOT_SALES),
                            nCustomers = uniqueN(LYLTY_CARD_NBR),
                            nTxnPerCust = uniqueN(TXN_ID) / uniqueN(LYLTY_CARD_NBR),
                            nChipsPerTxn = sum(PROD_QTY) / uniqueN(TXN_ID),
                            avgPricePerUnit = sum(TOT_SALES) / sum(PROD_QTY)),
                        by = .(STORE_NBR, YEARMONTH)][order(STORE_NBR, YEARMONTH)]

# (3) Filter to the pre-trial period and stores with full observation periods
# 1. Use the .N variable to count the number of rows within each group
# for the number of months in the pre-trial period.
storesWithFullObs <- measureOverTime[, .N, by = STORE_NBR][N == 12, STORE_NBR]
# 2. Then filter for stores with 12 months of pre-trial data
preTrialMeasures <- measureOverTime[YEARMONTH < 201902 & STORE_NBR %in% storesWithFullObs]

# (4) We need a function to rank potential control stores by their similarity
# to trial stores. This function will calculate the correlation of performance
# for each trial and control store pair.
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(me
                                                        inputTable[STORE_NBR == i, eval(metricCol)])
    )
    calcCorrTable <- rbind(calcCorrTable, calculatedMeasure)
  }
  return(calcCorrTable)
}

# (5) We can also measure similarity by the absolute difference between
# a trial store's performance and a control store's, using a standardied metric.
calculateMagnitudeDistance <- function(inputTable, metricCol, storeComparison) {
  calcDistTable <- data.table(Store1 = numeric(), Store2 = numeric(), YEARMONTH = numeric(), measure = n
```

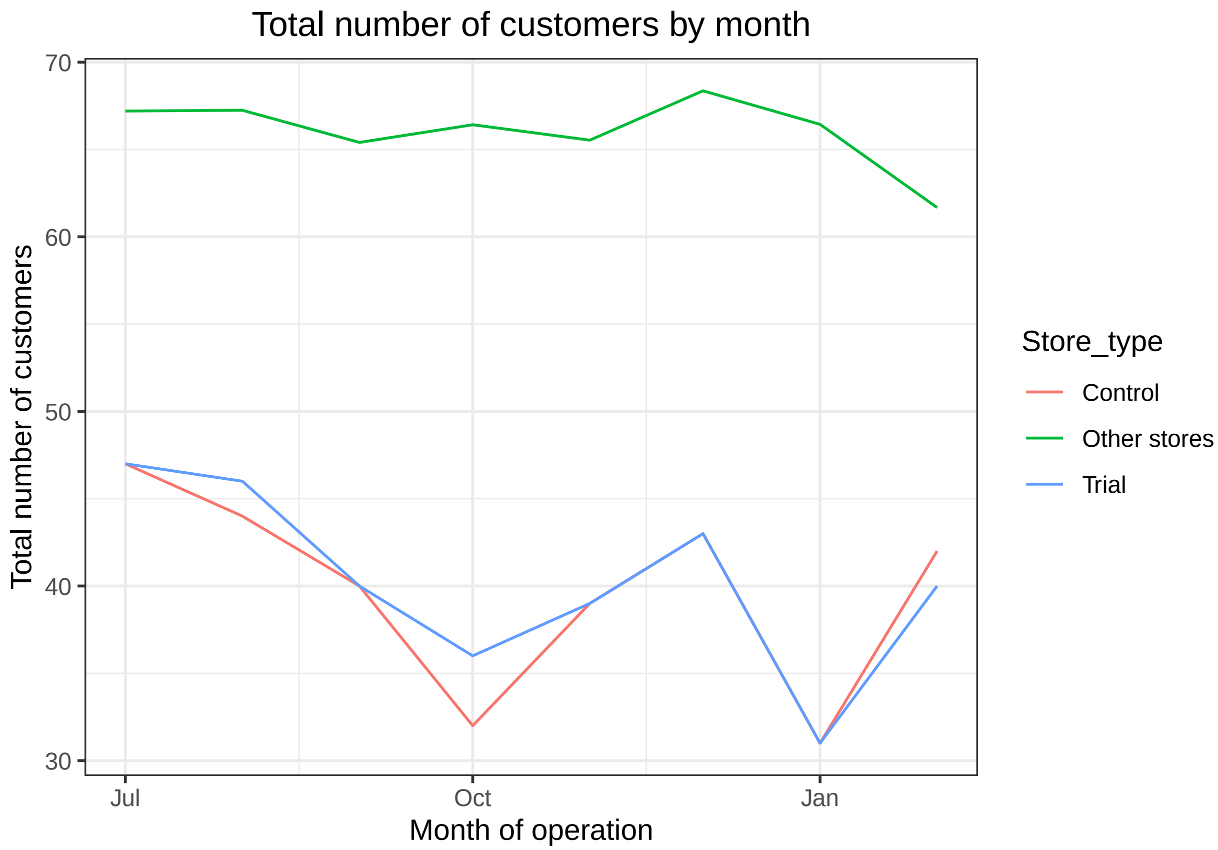
```r
  storeNumbers <- unique(inputTable[, STORE_NBR])

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

#### --- Step 3: Use the functions to find the control stores ---

# We'll select control stores based on similarity to trial stores in monthly total sales
# and customer count. This involves calculating four scores per potential control store:
# a correlation and a standardized absolute difference for each of these two metrics.

## --- 3.1. Analysis for trial store 77 ---

# Finding the control store and assessing the impact of the trial

# Find control stores for trial store 77
trial_store <- 77

# (1) Use the function to calculate correlations
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales), trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers), trial_store)
# (2) Use the functions for calculating magnitude
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales), trial_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures, quote(nCustomers), trial_store)

# All calculated scores will be combined into a composite score for ranking.
# This score will initially be a simple average (0.5 weight) of the correlation
# and magnitude scores for each driver. This weight can be adjusted
# if either trend similarity or absolute size is prioritised.

# (3) Combine Scores
corr_weight <- 0.5 # A simple average on the scores would be 0.5 * corr_measure + 0.5 * mag_measure

score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1", "Store2"))[, scoreNSales := corr_m
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = c("Store1", "Store2"))[, scoreNCus

# Now we have a score for each of total number of sales and number of customers.
# Let’s combine the two via a simple average.
score_Control <- merge(score_nSales, score_nCustomers, by = c("Store1", "Store2")) # Combine scores acr
```

```r
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5] # Combine the two via a simp

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

# Now that a control store has been identified, we will visually verify
# the similarity of key drivers in the pre-trial period, starting with total sales.

# (5) Total sales visual check
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                  ifelse(STORE_NBR == control_store, "Control",
][, totSales := mean(totSales), by = c("YEARMONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
][YEARMONTH < 201903, ] # Visual checks on trends based on the drivers

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




```r
# (6) Number of customers visual check
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                      ifelse(STORE_NBR == control_store, "Contro
```

```
][, nCusts := mean(nCustomers), by = c("YEARMONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
][YEARMONTH < 201903, ] # Conduct visual checks on customer count trends by comparing the trial store t

ggplot(pastCustomers, aes(TransactionMonth, nCusts, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Total number of customers", title = "Total number of customers by
```




```
# (7) Assess impact of trial (Trial Store 77)
# Now we'll assess the trial's impact on overall chip sales from March to June 2019.
# First, we'll scale the control store's sales to account for pre-trial differences.

# Scale pre-trial control sales to match pre-trial trial store sales
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum(tot

# Apply scaling factor and calculate percentage difference
measureOverTimeSales <- measureOverTime

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

# (8) Now with comparable scaled control sales, we can calculate the percentage difference
# from trial store sales during the trial period.

# Calculate the percentage difference between scaled control sales and trial sales
percentageDiff <- merge(scaledControlSales[, c("YEARMONTH", "controlSales")],
```

```
                          measureOverTimeSales[STORE_NBR == trial_store, c("nCustomers", "totSales", "YEA
                          by = "YEARMONTH")[, percentageDiff := abs(totSales - controlSales) / controlSal

# (9) As our null hypothesis is that the trial period is the same as the
# pre-trial period, let's take the standard deviation based on the scaled
# percentage difference in the pre-trial period

# Calculate standard deviation and degrees of freedom
stdDev <- sd(percentageDiff[YEARMONTH < 201902, percentageDiff])
# Note that there are 8 months in the pre-trial period.
# hence8-1 = 7 degrees of freedom
degreesOfFreedom <- 7

# (10) We will test with a null hypothesis of 0 difference between trial and control stores.

# Calculate the t-values for the trial months. After that, find the 95th percentile
# of the t distribution with the appropriate degrees of freedom
percentageDiff[, tValue := (percentageDiff - 0) / stdDev]
percentageDiff[, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"),
percentageDiff[YEARMONTH < 201905 & YEARMONTH > 201901, .(TransactionMonth, tValue)]
```

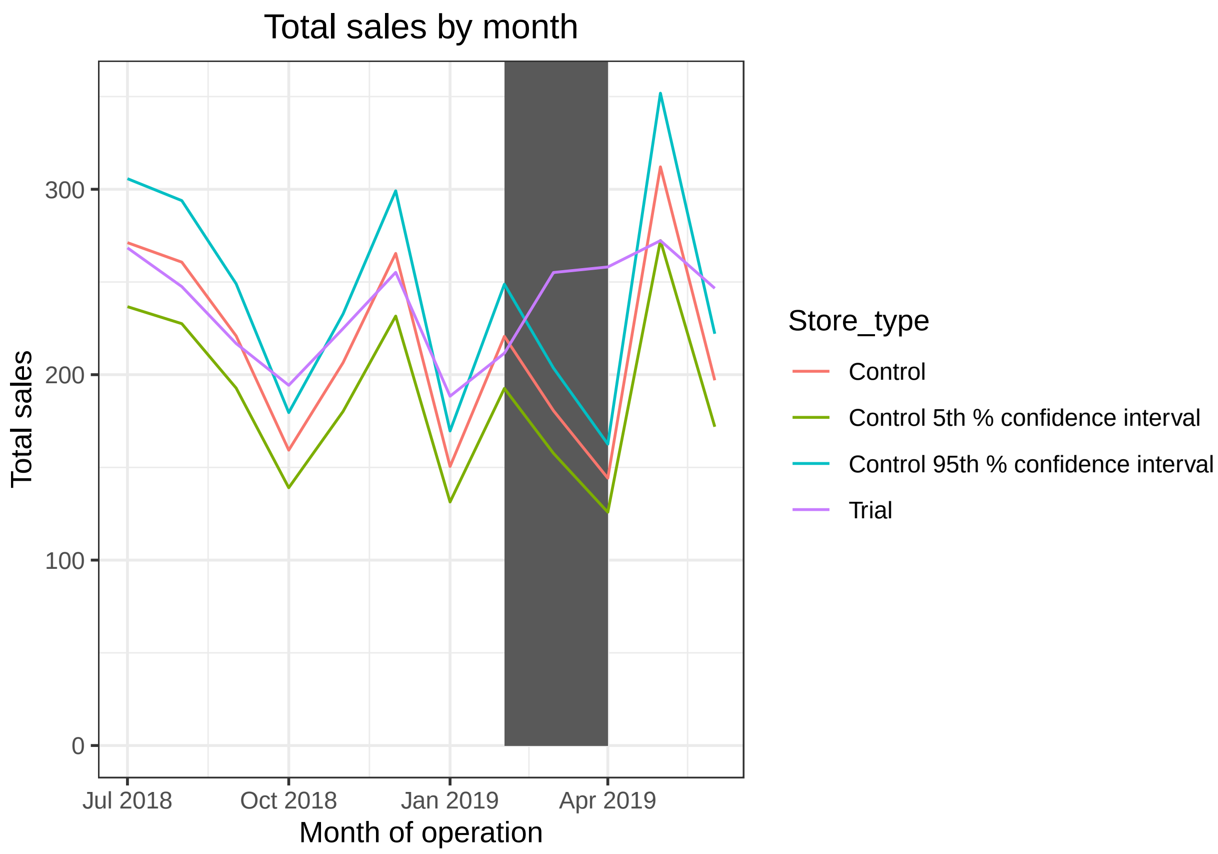
```
##    TransactionMonth    tValue
##              <Date>     <num>
## 1:       2019-02-01  1.223912
## 2:       2019-03-01  5.633494
## 3:       2019-04-01 11.336505
```

```
# Find the 95th percentile of the t-distribution with the appropriate
qt(0.95,df= degreesOfFreedom)
```

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

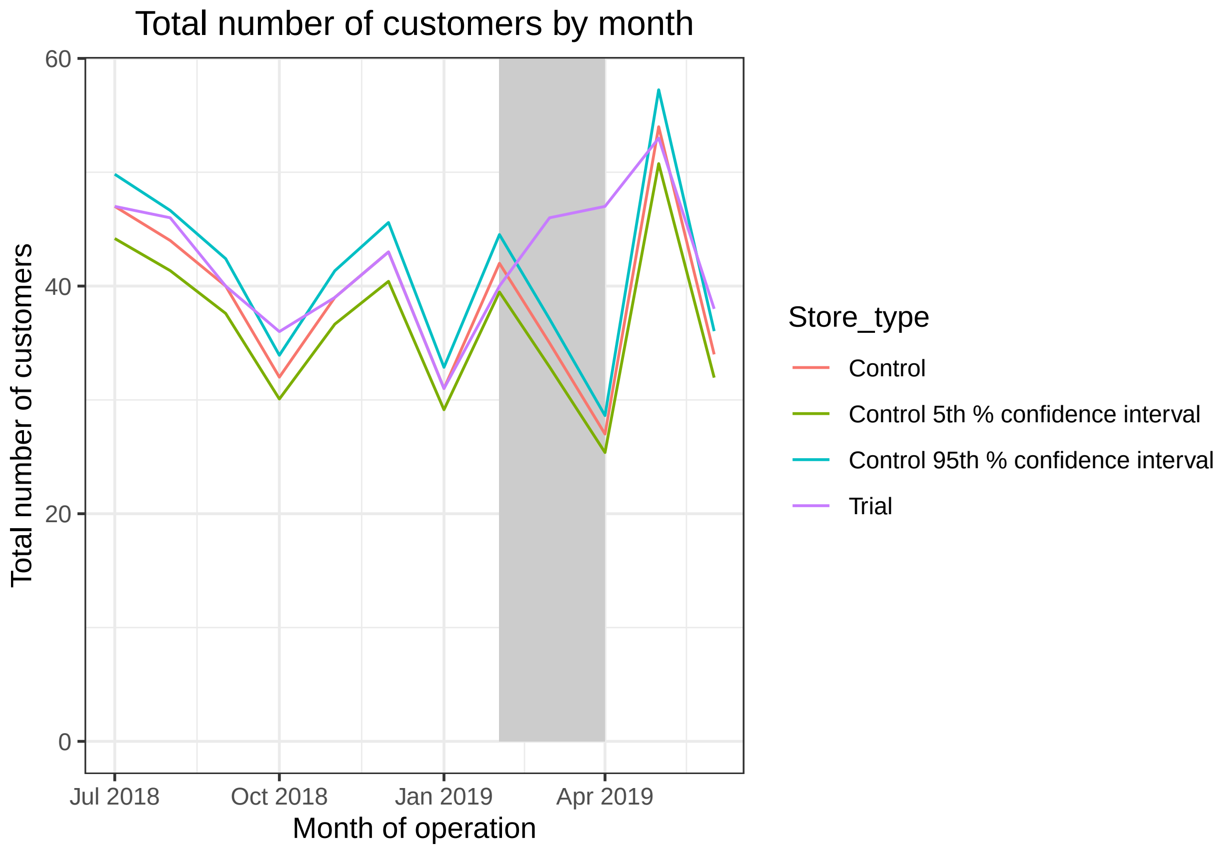
```
# A t-test showed our special store had significantly higher sales in March and April.
# We'll visualise this difference, as the t-test confirms it's a real effect, not just chance.

# (10) Visual assessment of trial impact
measureOverTimeSales <- measureOverTime

# Trial and control store totalsales
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial", ifelse(STOR
][, totSales := mean(totSales), by = c("YEARMONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
][Store_type %in% c("Trial", "Control"), ]

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

# Control store5thpercentile
pastSales_Controls5 <- pastSales[Store_type == "Control",
][, totSales := totSales * (1 - stdDev * 2)
```

```r
][, Store_type := "Control 5th % confidence interval"]

trialAssessment <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)

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




```r
# The trial in Store 77 showed a significant difference from its control,
# with trial store performance falling outside the 5-95% confidence interval
# for two of three trial months.

# Next, we will assess this for customer numbers.

# Scale pre-trial control customers to match pre-trial trial store customers.
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum(nCus
  preTrialMeasures[STORE_NBR == control_store & YEARMONTH < 201902, sum(nCustomers)]

# Apply the scaling factor
measureOverTimeCusts <- measureOverTime

scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store,
```

```
][, controlCustomers := nCustomers * scalingFactorForControlCust
][, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                         ifelse(STORE_NBR == control_store, "Control", "Otherstores"))
]

percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH", "controlCustomers")],
                        measureOverTimeCusts[STORE_NBR == trial_store, c("nCustomers", "YEARMONTH")],
                        by = "YEARMONTH"
)[, percentageDiff := abs(controlCustomers - nCustomers) / controlCustomers]

# As our null hypothesis is that the trial period is the same as the
# pre-trial period, let's take the standard deviation based on the scaled
# percentage difference in the pre-trial period
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7

# Trial and control store number of customers
pastCustomers <- measureOverTimeCusts[, nCusts := mean(nCustomers), by =
                                        c("YEARMONTH", "Store_type")
][Store_type %in% c("Trial", "Control"), ]
# Control store 95th percentile
pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",
][, nCusts := nCusts * (1 + stdDev * 2)
][, Store_type := "Control 95th % confidence interval"]
# Control store 5th percentile
pastCustomers_Controls5 <- pastCustomers[Store_type == "Control",
][, nCusts := nCusts * (1 - stdDev * 2)
][, Store_type := "Control 5th % confidence interval"]
trialAssessment <- rbind(pastCustomers, pastCustomers_Controls95,
                         pastCustomers_Controls5)

# Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, nCusts, color = Store_type)) +
  geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,], # Highlight trial perio
            aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
                ymin = 0 , ymax = Inf, color = NULL), show.legend = FALSE, fill = "grey80", alpha = 0.5)
  geom_line() +
  labs(x = "Month of operation", y = "Total number of customers", title = "Total number of customers by
```

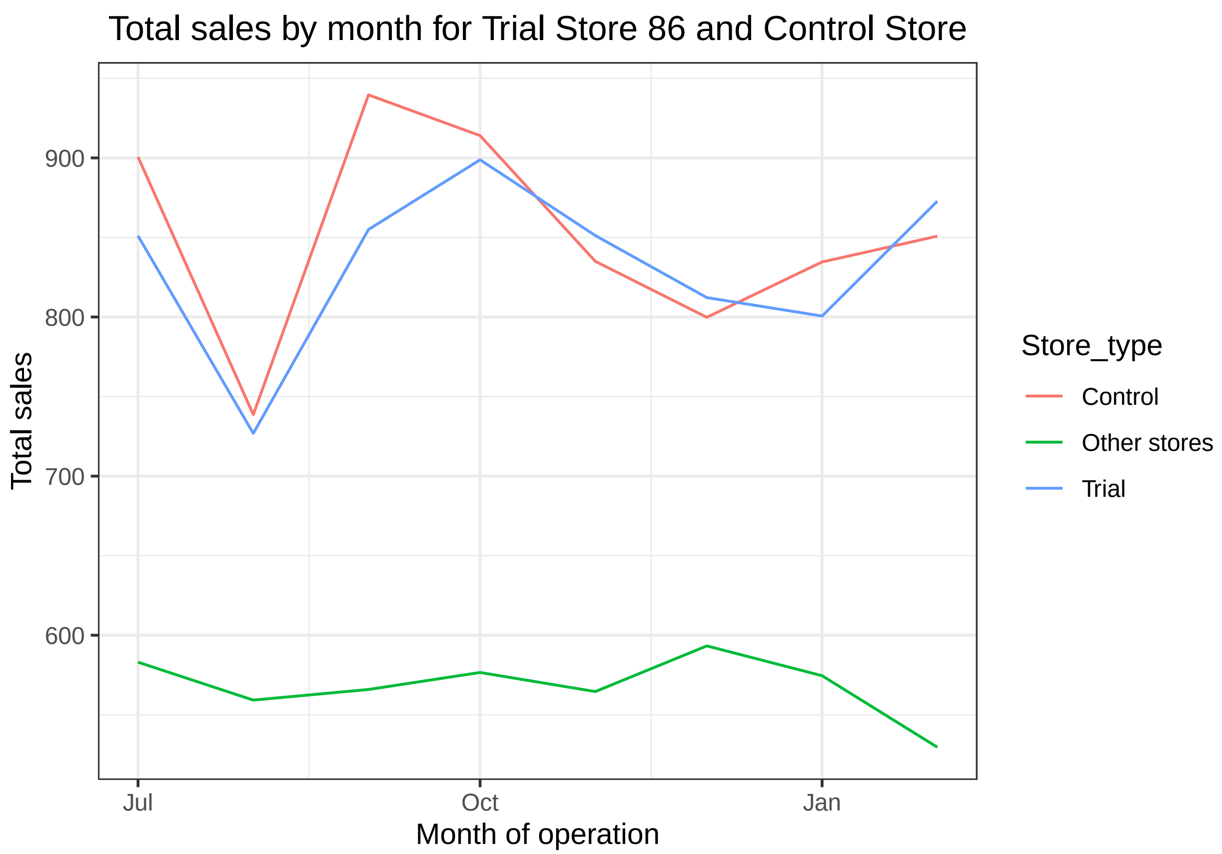



```r
# Now we have fully analysed the impact of the trial for Store 77 on both sales
# and customer numbers.

## --- 3.2 Analysis for trial store 86 ---

## (1) Control Store Selection for Trial Store 86

# Use the functions created earlier to calculate correlations
# and magnitude for each potential control store
measureOverTime <- data[,.(totSales= sum(TOT_SALES),
                          nCustomers= uniqueN(LYLTY_CARD_NBR),
                          nTxnPerCust=
                            uniqueN(TXN_ID)/uniqueN(LYLTY_CARD_NBR),
                          nChipsPerTxn= sum(PROD_QTY)/uniqueN(TXN_ID),
                          avgPricePerUnit= sum(TOT_SALES)/sum(PROD_QTY)
)
,by=c("STORE_NBR","YEARMONTH")][order(STORE_NBR,
                                      YEARMONTH)]

# Use the functions for calculating correlation
trial_store <- 86 # Set the trial store number to 86

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

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

```r
# Use the functions for calculating magnitude
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales), trial_store)

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

# Create a combined score composed of correlation and magnitude
corr_weight <- 0.5

score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1", "Store2"))[, scoreNSales := corr_n

score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = c("Store1", "Store2"))[, scoreNCus

# Combine scores across the drivers using a simple average.
score_Control <- merge(score_nSales, score_nCustomers, by = c("Store1", "Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]

# Select control store for trial store 86
# Select the control store as the second-highest ranked match
# (closest to 1, excluding the trial store itself)
control_store <- score_Control[Store1 == trial_store, ][order(-finalControlScore)][2, Store2] # Correct
control_store
```

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

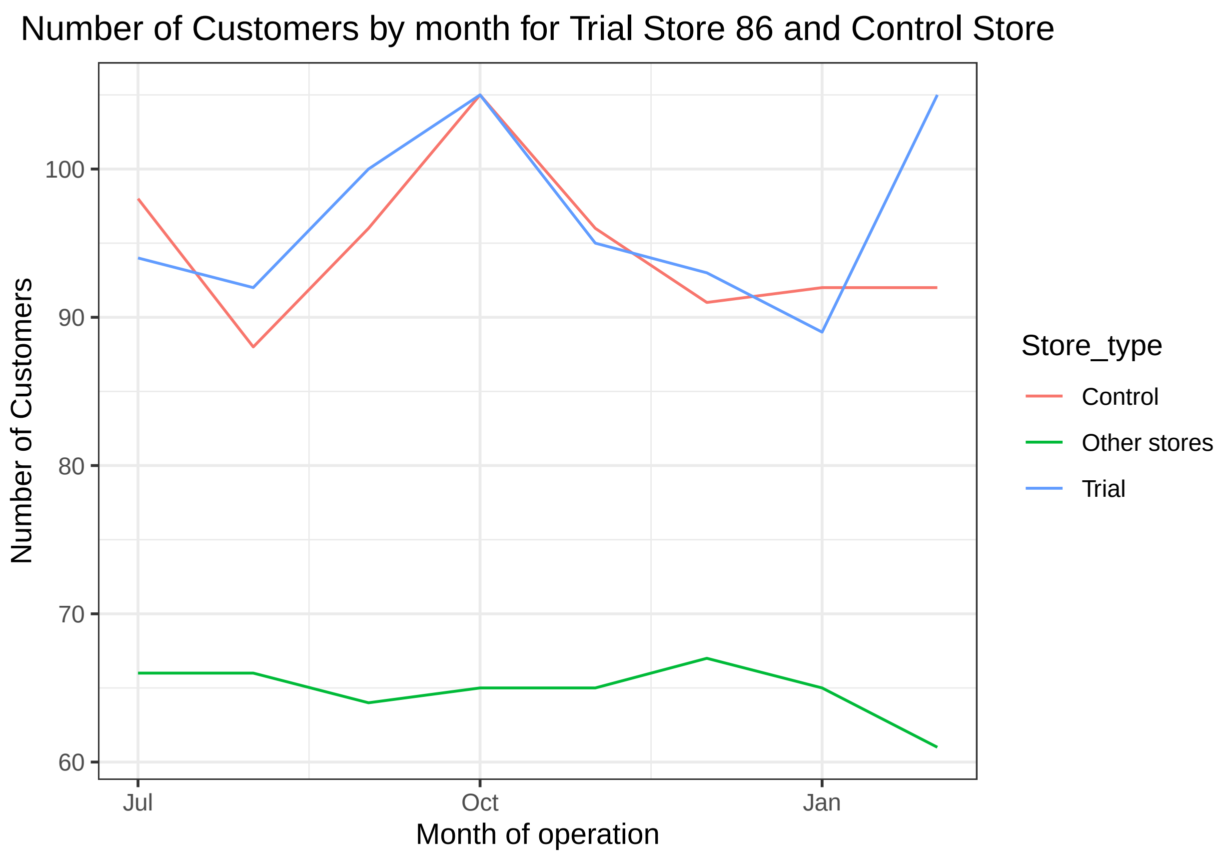
```r
# Store 155 is selected as the control for trial store 86. We'll now visually
# confirm their pre-trial similarity, starting with total sales.

# Conduct visual checks on trends based on the drivers
measureOverTimeSales <- measureOverTime # Starting with the global measureOverTime data
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                   ifelse(STORE_NBR == control_store, "Control",
][, totSales := mean(totSales), by = c("YEARMONTH", "Store_type") # Calculate mean sales per month per
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d") #
][YEARMONTH < 201903 , ] # Filter to months before and up to Feb 2019

ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by month for Trial Store 86 and
```




```
# Sales are trending in a similar way.

# Next, number of customers
# Conduct visual checks on trends based on the drivers
measureOverTimeCusts <- measureOverTime # Starting with the global measureOverTime data
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                       ifelse(STORE_NBR == control_store, "Contro
][, nCustomers := mean(nCustomers), by = c("YEARMONTH", "Store_type") # Calculate mean customers per mo
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d") #
][YEARMONTH < 201903 , ] # Filter to months before and up to Feb 2019
```

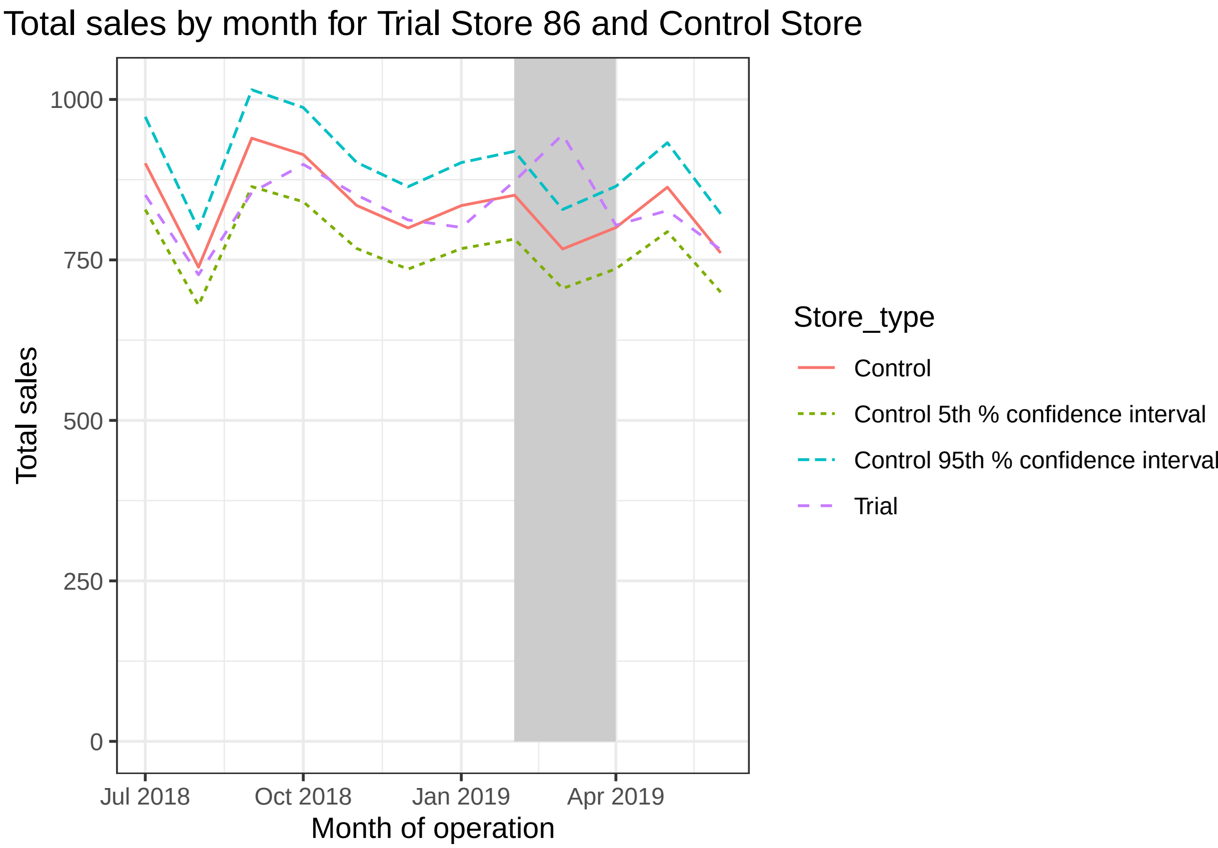
```
## Warning in `[.data.table`(measureOverTimeCusts[, `:=`(Store_type,
## ifelse(STORE_NBR == : 66.828244 (type 'double') at RHS position 1
## out-of-range(NA) or truncated (precision lost) when assigning to type 'integer'
## (column 4 named 'nCustomers')
```

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




```r
# The customer trends are also similar, which is good.
# Now, let's assess the trial's impact on sales.

# (2) Assess impact of trial on sales (trial store 86)
# Scale pre-trial control sales to match pre-trial trial store sales
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
                                YEARMONTH < 201902, sum(totSales)]/preTrialMeasures[STORE_NBR == control
                                YEARMONTH < 201902, sum(totSales)]
# Apply the scaling factor
measureOverTimeSales <- measureOverTime
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][ ,
                      controlSales := totSales * scalingFactorForControlSales]

# Calculate the percentage difference between scaled control sales and trial sales
percentageDiff <- merge(measureOverTimeSales[STORE_NBR == trial_store, .(YEARMONTH, totSales)],
                        scaledControlSales[STORE_NBR == control_store, .(YEARMONTH, controlSales)],
                        by = "YEARMONTH")[, percentageDiff := (totSales - controlSales) / controlSales]

# Our null hypothesis assumes the trial period's performance is no different
# from the pre-trial period's. So, we'll calculate the standard deviation using
# the scaled percentage difference from the pre-trial data.

# Calculate the standard deviation of percentage differences during the pre-trial period
stdDev <- sd(percentageDiff[YEARMONTH < 201902, percentageDiff])
degreesOfFreedom <- 7 # Already provided as 7
```

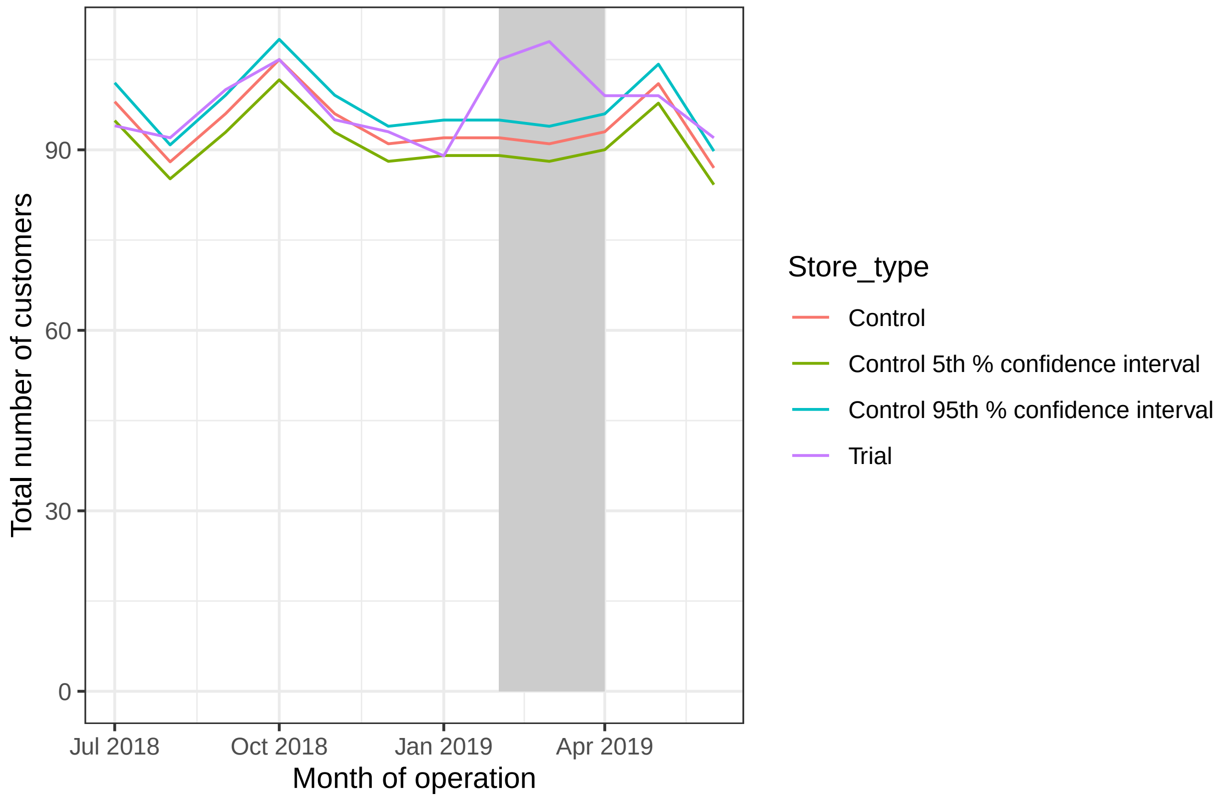
```r
#### Trial and control store total sales
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                         ifelse(STORE_NBR == control_store, "Control",
][, totSales := totSales # No aggregation needed if measureOverTime is already monthly
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
][Store_type %in% c("Trial", "Control"), ] # Filter for only Trial and Control stores

# Calculate the 5th and 95th percentile for control store sales
pastSales_Controls95 <- pastSales[Store_type == "Control",
][, totSales := totSales * (1 + stdDev * 2) # Use stdDev in percentages to adjust sales
][, Store_type := "Control 95th % confidence interval"]
pastSales_Controls5 <- pastSales[Store_type == "Control",
][, totSales := totSales * (1 - stdDev * 2) # Use stdDev in percentages to adjust sales
][, Store_type := "Control 5th % confidence interval"]

## create a combined table with columns from pastSales,
# pastSales_Controls95 and pastSales_Controls5
trialAssessment <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)

# Plotting these in one graph
ggplot(trialAssessment, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
            aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),  ymin = 0 , ymax = Inf, col
  geom_line(aes(linetype = Store_type)) +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by month for Trial Store 86 an
```




```
# The results show that the trial in Store 86 wasn't significantly different from
# its control store during the trial period. This is because the trial store's
# performance stayed within the control store's 5% to 95% confidence interval
# for two out of the three trial months.

# (3) Assess Impact of Trial on Customers
# Scale pre-trial control customers to match pre-trial trial store customers
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store &
                                                  YEARMONTH < 201902, sum(nCustomers)]/preTrialMeasures

#### Apply the scaling factor
measureOverTimeCusts <- measureOverTime
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store,
][ , controlCustomers := nCustomers
   * scalingFactorForControlCust
][, Store_type := ifelse(STORE_NBR
                         == trial_store, "Trial",
                         ifelse(STORE_NBR == control_store,
                                "Control", "Other stores"))
]

# Calculate the percentage difference between scaled control sales and trial sales
percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH", "controlCustomers")],
                        measureOverTime[STORE_NBR == trial_store, c("nCustomers", "YEARMONTH")],
                        by = "YEARMONTH")[, percentageDiff := abs(nCustomers - controlCustomers) / cont
```

```
# As our null hypothesis is that the trial period is the same as the pre-trial
# period, let's take the standard deviation based on the scaled percentage difference
# in the pre-trial period

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

# Trial and control store number of customers
pastCustomers <- measureOverTimeCusts[, nCusts := mean(nCustomers), by =
                                        c("YEARMONTH", "Store_type")
][Store_type %in% c("Trial", "Control"), ]

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

# Control store 5th percentile
pastCustomers_Controls5 <- pastCustomers[Store_type == "Control",
][, nCusts := nCusts * (1 - stdDev * 2)
][, Store_type := "Control 5th % confidence interval"]
trialAssessment <- rbind(pastCustomers, pastCustomers_Controls95,
                         pastCustomers_Controls5)

# Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, nCusts, color = Store_type)) +
  geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
            aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),  ymin = 0 , ymax = Inf, colo
  geom_line() +
  labs(x = "Month of operation", y = "Total number of customers", title = "Total number of customers by
```

nber of customers by month for Trial Store 86 and Control Store

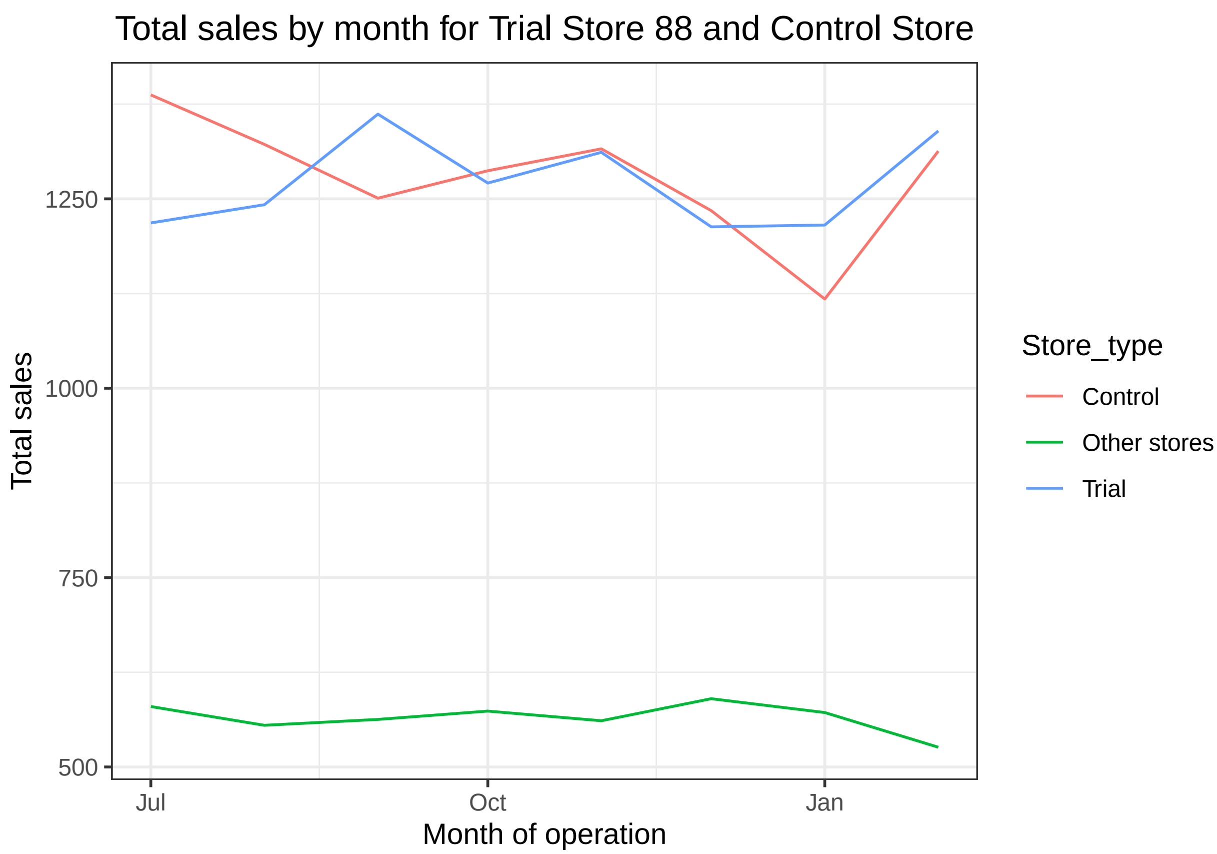



```
# It seems the trial significantly boosted customer numbers in Store 86
# across all three months, despite sales not showing a similar significant increase.
# We should ask the Category Manager if any special deals lowered prices,
# potentially affecting the sales results.

### --- 3.3. Analysis for trial store 88 ---
measureOverTime <- data[, .(
  totSales = sum(TOT_SALES),
  nCustomers = uniqueN(LYLTY_CARD_NBR),
  nTxnPerCust = uniqueN(TXN_ID) / uniqueN(LYLTY_CARD_NBR),
  nChipsPerTxn = sum(PROD_QTY) / uniqueN(TXN_ID),
  avgPricePerUnit = sum(TOT_SALES) / sum(PROD_QTY)
), by = c("STORE_NBR", "YEARMONTH")][order(STORE_NBR, YEARMONTH)]

# (1) Use the functions from earlier to calculate the correlation of the sales and
# number of customers of each potential control store to the trial store
trial_store <- 88 # Set the trial store number to 88
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales), trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers), trial_store)

# Use the functions from earlier to calculate the magnitude distance of the
# sales and number of customers of each potential control store to the trial store
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures, quote(totSales), trial_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures, quote(nCustomers), trial_store)

# (2) Create a combined score composed of correlation and magnitude by merging the
```

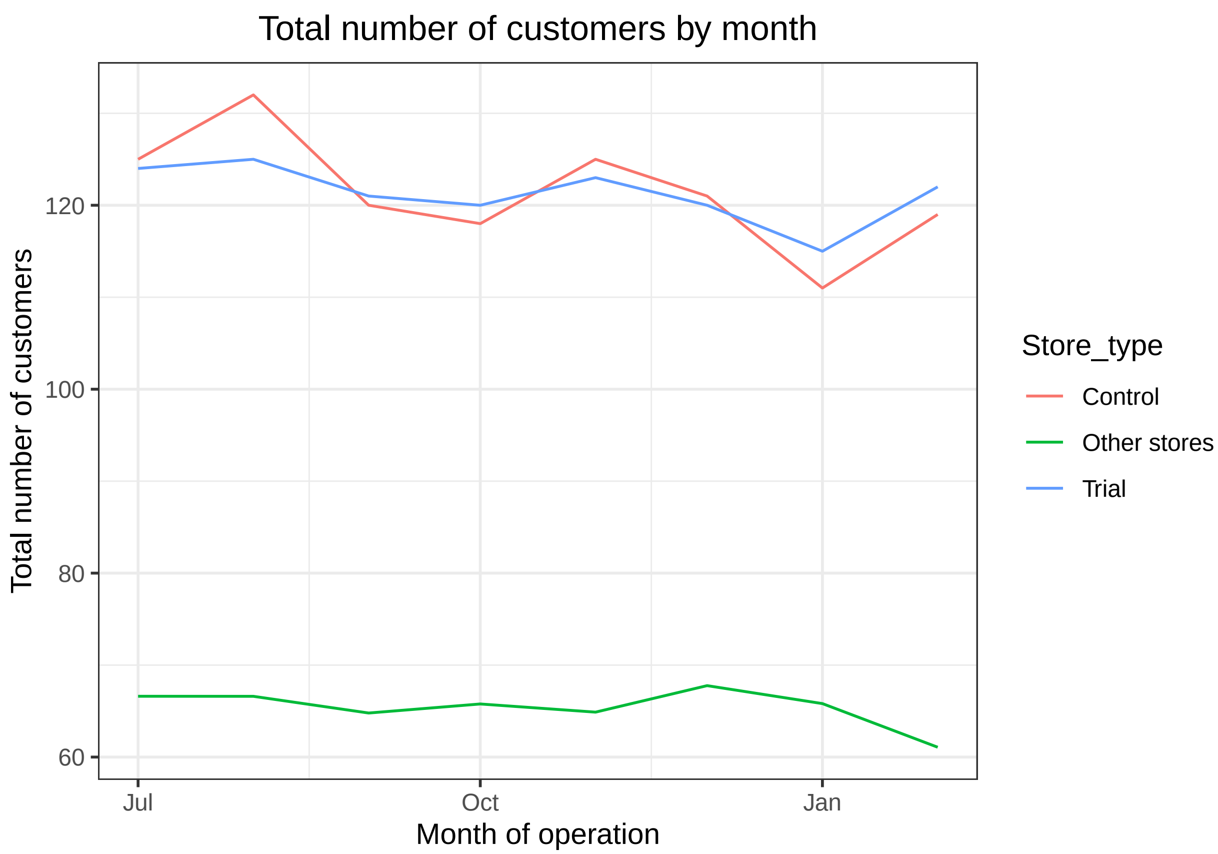
```
# correlations table and the magnitudes table, for each driver.
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1", "Store2"))[, scoreNSales := corr_m
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = c("Store1", "Store2"))[, scoreNCus

# Combine scores across the drivers by merging sales scores and customer scores,
# and compute a final combined score.
score_Control <- merge(score_nSales, score_nCustomers, by = c("Store1", "Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]

# (3) Select control store for trial store 88
control_store <- score_Control[Store1 == trial_store, ][order(-finalControlScore)][2, Store2]
control_store
```

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

```
# We've found Store 237 to be a suitable control for trial store 88.
# (4) Now, let's visually check if their pre-trial sales trends are similar.

# Visual Checks on Sales for Trial Store 88
measureOverTimeSales <- measureOverTime # Using the globally defined measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                   ifelse(STORE_NBR == control_store, "Control",
][, totSales := mean(totSales), by = c("YEARMONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
][YEARMONTH < 201903 , ] # Pre-trial period plus Feb 2019

ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by month for Trial Store 88 and
```

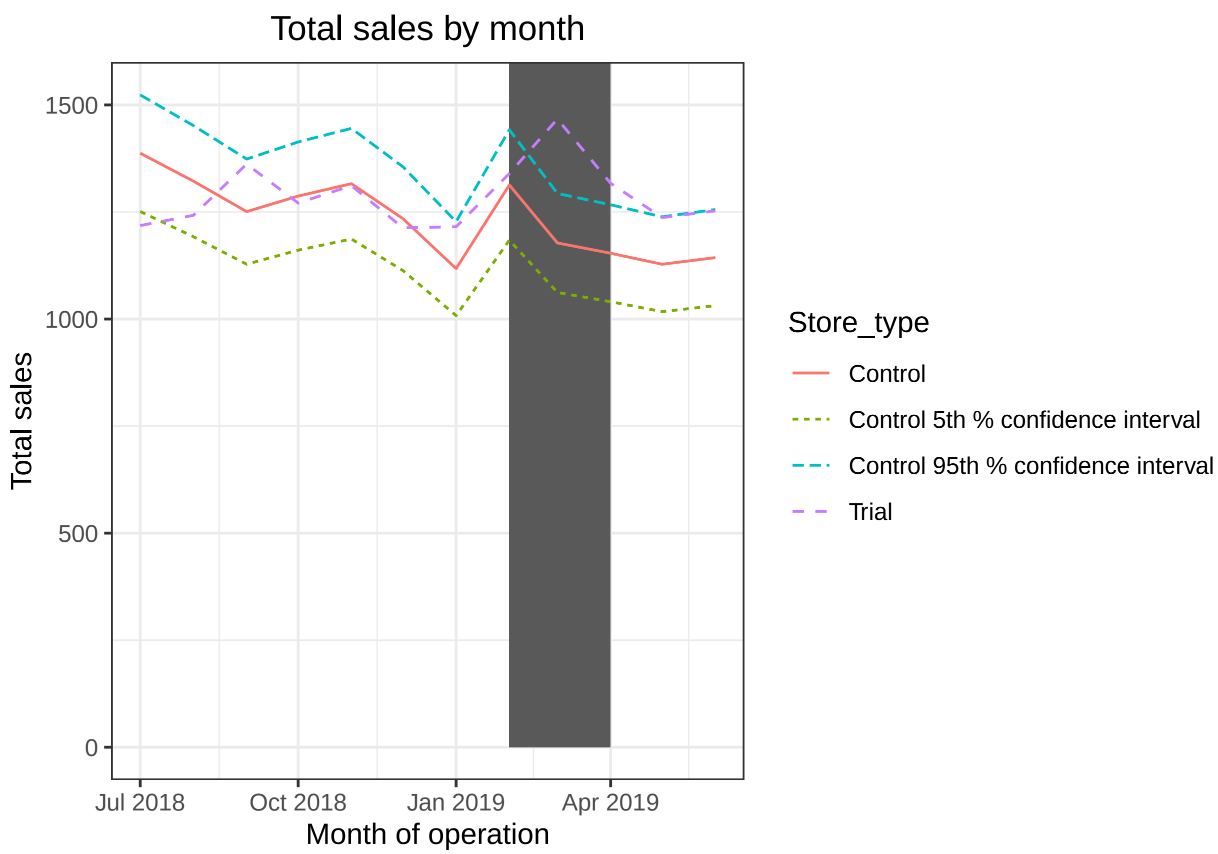



```r
# The trial and control stores have similar total sales.
# (5) Next,number of customers.

# Visual checks on trends based on the drivers
measureOverTimeCusts <- measureOverTime

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

ggplot(pastCustomers, aes(TransactionMonth, numberCustomers, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Total number of customers", title = "Total number of customers by
```



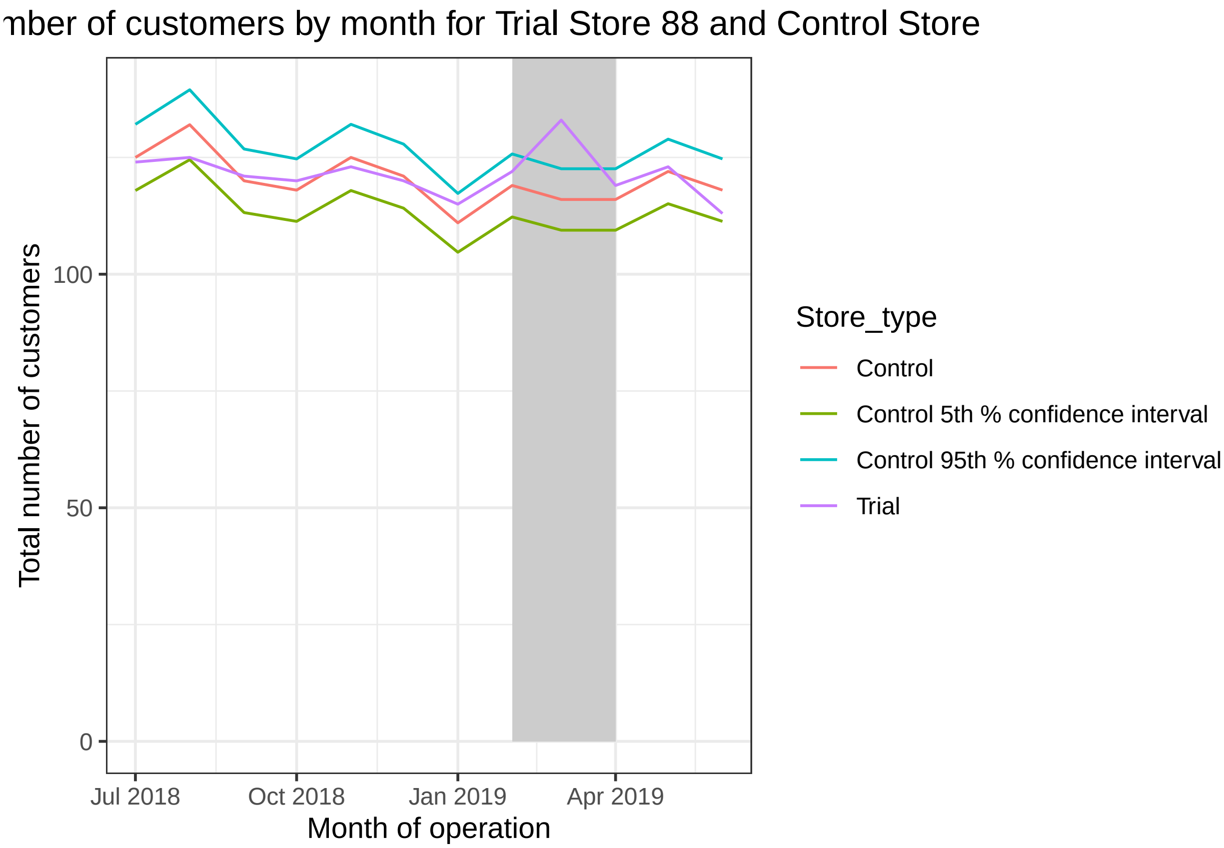

```r
# The customer numbers for both control and trial stores are similar.

# (6) Let's now assess the impact of the trial on sales.
# Scale pre-trial control sales to match pre-trial trial store sales
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store & YEARMONTH < 201902, sum(tot
  preTrialMeasures[STORE_NBR == control_store & YEARMONTH < 201902, sum(totSales)]

# Apply the scaling factor to control store sales
measureOverTimeSales <- measureOverTime # Use a copy to avoid modifying original measureOverTime direct
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store,
][, controlSales := totSales * scalingFactorForControlSales]

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

# Assuming no difference between trial and pre-trial periods (null hypothesis),
# calculate the standard deviation based on the scaled percentage difference in the pre-trial period.
stdDev <- sd(percentageDiff[YEARMONTH < 201902, percentageDiff])
degreesOfFreedom <- 7 # 8 months in pre-trial period, so 8-1 = 7 degrees of freedom

# Prepare data for visual assessment of trial impact on total sales
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial",
                                                  ifelse(STORE_NBR == control_store, "Control",
```

```r
][, totSales := mean(totSales), by = c("YEARMONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
][Store_type %in% c("Trial", "Control"), ]

# Calculate 95th percentile for control store sales
pastSales_Controls95 <- pastSales[Store_type == "Control",
][, totSales := totSales * (1 + stdDev * 2) # Assuming 2 standard deviations for ~95% CI for simplifica
][, Store_type := "Control 95th % confidence interval"]

# Calculate 5th percentile for control store sales
pastSales_Controls5 <- pastSales[Store_type == "Control",
][, totSales := totSales * (1 - stdDev * 2) # Assuming 2 standard deviations for ~5% CI for simplificat
][, Store_type := "Control 5th % confidence interval"]

# Combine all sales data for plotting
trialAssessment <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)

# Plotting the combined sales data with confidence intervals and highlighted trial period
ggplot(trialAssessment, aes(TransactionMonth, totSales, color = Store_type)) +
  # Highlight the trial period
  geom_rect(data = trialAssessment[YEARMONTH < 201905 & YEARMONTH > 201901, ],
            aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0, ymax = Inf, color
            show.legend = FALSE) +
  # Plot the sales lines, differentiating by Store_type
  geom_line(aes(linetype = Store_type)) +
  # Add labels and title
  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 from
# its control store in the trial period, as the trial store's performance lies
# outside of the 5% to 95% confidence interval of the control store in two of
# the three trial months. Let's have a look at assessing this for the number of
# customers as well.

# (7) Assess Impact of Trial on Customers
# Scale pre-trial control store customers to match pre-trial trial store customers
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store &
                                                  YEARMONTH < 201902, sum(nCustomers)]/preTrialMeasures

# Apply the scaling factor
measureOverTimeCusts <- measureOverTime
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store,
][ , controlCustomers := nCustomers
   * scalingFactorForControlCust
][, Store_type := ifelse(STORE_NBR
                         == trial_store, "Trial",
                         ifelse(STORE_NBR == control_store,
                                "Control", "Other stores"))
]

# Calculate the absolute percentage difference between scaled control sales and
# trial sales (should be customers here)
percentageDiff <- merge(measureOverTimeCusts[STORE_NBR == trial_store, .(YEARMONTH, nCustomers)],
                        scaledControlCustomers[STORE_NBR == control_store, .(YEARMONTH, controlCustomer
```

```
                         by = "YEARMONTH"
)[, percentageDiff := (nCustomers - controlCustomers) / controlCustomers] # Using signed difference for

# As our null hypothesis is that the trial period is the same as the pre-trial
# period, let's take the standard deviation based on the scaled percentage difference
# in the pre-trial period
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7 # note that there are 8 months in the pre-trial period hence
# 8 - 1 = 7 degrees of freedom

# Trial and control store number of customers
pastCustomers <- measureOverTimeCusts[, nCustomers := nCustomers, by =
                                        c("YEARMONTH", "Store_type")
][Store_type %in% c("Trial", "Control"), ] # Filter for only Trial and Control stores for plotting

# Control store 95th percentile
pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",
][, nCustomers := nCustomers * (1 + stdDev * 2)
][, Store_type := "Control 95th % confidence interval"]

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

# Combine the tables pastSales, pastSales_Controls95, pastSales_Controls5 (should be pastCustomers rela
trialAssessment <- rbind(pastCustomers, pastCustomers_Controls95, pastCustomers_Controls5)

# Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, nCustomers, color = Store_type)) +
  geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
          aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),  ymin = 0 , ymax = Inf, colo
  geom_line() +
  labs(x = "Month of operation", y = "Total number of customers", title = "Total number of customers by
```




```
# 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.

#### --- Step 4: Conclusion ---
# We successfully identified control stores: 233 for trial store 77,
# 155 for trial store 86, and 237 for trial store 88.

# Trial stores 77 and 88 demonstrated a significant sales uplift in at least two
# of the three trial months. However, trial store 86 did not show a similar significant
# difference. We recommend investigating potential variations in trial implementation
# at store 86. Overall, the trial indicates a significant increase in sales.

# With this analysis complete, we are ready to prepare our presentation
# for the Category Manager.
```