```
# Load required libraries
library(data.table)
library(ggplot2)
library(tidyr)

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

# Load data
QVI_data <- read.csv("intership/QVI_data.csv")
data <- as.data.table(QVI_data)



#### Calculate these measures overtime for each store
#### add a month ID
data[, YEARMONTH := year(DATE)* 100 + month(DATE)]



#### 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 = c("STORE_NBR", "YEARMONTH")][order(STORE_NBR, YEARMONTH)]

#### Filter to the pre-trial period and stores with full observation periods
storeWithFullObs <- unique(measureOverTime[, .N, STORE_NBR][N == 12, STORE_NBR])
```

```
preTrialMeasure <- measureOverTime[YEARMONTH < 201902 & STORE_NBR %in% storeWithFullObs,]

#### Create a function to calculate the 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)

 }

 return(calcCorrTable)

}
```

```
#### Create a function to calculate a standardized magnitude distance for a measure,
#### looping through each control
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)

 }

#### Standardize the magnitude distance so that the measure ranges from 0 to 1

 minMaxDist <- calcDistTable[, .(minDist = min(measure),
                                 maxDist = max(measure)),
                             by = c("Store1", "YEARMONTH")]
```

```
 distTable <- merge(calcDistTable, minMaxDist, by = c("Store1", "YEARMONTH"))

 distTable[, magnitudeMeasure := 1 - (measure - minDist)/(maxDist - minDist)]

 finalDistTable <- distTable[, .(mag_measure = mean(magnitudeMeasure)),
                      by = .(Store1, Store2)]

 return(finalDistTable)

}

#### Set up trial store
trial_store77 <- 77

#### Calculate correlation for sales and customers
corr_nSales_77 <- calculateCorrelation(preTrialMeasure, quote(totSales), trial_store77)

corr_nCustomers_77 <- calculateCorrelation(preTrialMeasure, quote(nCustomers), trial_store77)

#### Calculate absolute difference magnitude for sales and customers
magnitude_nSales_77 <- calculateMagnitudeDistance(preTrialMeasure, quote(totSales), trial_store77)
magnitude_nCustomers_77 <- calculateMagnitudeDistance(preTrialMeasure, quote(nCustomers),
trial_store77)

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

score_nSales_77 <- merge(corr_nSales_77, magnitude_nSales_77,
```

```
                    by = c("Store1", "Store2"))[, scoreNSales := corr_measure * corr_weight +
                                                mag_measure * (1 - corr_weight)]

score_nCustomers_77 <- merge(corr_nCustomers_77, magnitude_nCustomers_77,
                             by = c("Store1", "Store2"))[, scoreNCust := corr_measure * corr_weight +
                                                         mag_measure * (1 - corr_weight)]



#### Combine the scores across the drivers
score_Control <- merge(score_nSales_77,
                       score_nCustomers_77,
                       by = c("Store1", "Store2"))[, finalControlScore := scoreNSales * 0.5 +
                                                   scoreNCust * 0.5]

#### Select control store based on the highest matching store (closest to 1 but not exactly 1)
control_store <- score_Control[Store1 == trial_store77][order(-finalControlScore)][2, Store2]

score_Control[order(-finalControlScore)][2, Store2] #answer: 233



# Start with sales

#### Visual checks on trends based on the drivers
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[,
                                  Store_type := ifelse(STORE_NBR == trial_store77, "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, ]

pastSales[Store_type %in% c("Control", "Trial"), c("TransactionMonth",
                                        "STORE_NBR", "totSales", "Store_type")]


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


#### Visual checks on trends based on the drivers
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store77, "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, ]

ggplot(pastCustomers, aes(x = TransactionMonth, y = numberCustomers, color = Store_type)) +
  geom_line() + labs(x = "Month of operation", y = "Total No. of customers", title =
                       "Total No. of customers by month")

#### Scale pre-trial control sales to match pre-trial trial store sales
scalingFactoForControlSales <- preTrialMeasure[STORE_NBR == trial_store77 &
                                        YEARMONTH < 201902,sum(totSales)]/
  preTrialMeasure[STORE_NBR == control_store & YEARMONTH < 201902, sum(totSales)]
```

```
#### Apply the scaling factor
measureOverTimeSales <- measureOverTime
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store,
][, controlSales := totSales * scalingFactoForControlSales]

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

#### Calculate the percentage difference between scaled control sales and trial sales
percentageDiff <- merge(scaledControlSales[, c("YEARMONTH", "controlSales")],
                        measureOverTime[STORE_NBR == trial_store77, c("YEARMONTH", "totSales")],
                        by = "YEARMONTH")[, percentageDiff := abs(controlSales - totSales)/controlSales]

# Let's check if the difference is significant

#### Since 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])

#### There are 8 months in the pre-trial period, hence 8 - 1 = 7 degrees of freedom
degreesOfFreedom <- 7

#### We will test with a null hypothesis of there being 0 difference between trial
#### and control stores
percentageDiff[, tValue := (percentageDiff - 0)/stdDev
```

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

#### compare against
qt(0.95, df = degreesOfFreedom) # 1.894579



measureOverTimeSales <- measureOverTime
#### Trial and Control store total sales
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store77, "Trial",
                                                         ifelse(STORE_NBR == control_store, "Control", "Other stores"))
][, totSales := mean(totSales), by = c("YEARMONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1,
                                      sep = "-"), "%Y-%m-%d")][Store_type %in% c("Trial", "Control")]

#### Control store 95th percentile
pastSales_Conrtols95 <- pastSales[Store_type == "Control",
][, totSales := totSales * (1 + stdDev * 2)
][, Store_type := "Conrol 95th % confidence interval"]

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

trialAssessment <- rbind(pastSales, pastSales_Conrtols95, pastSales_Controls5)
```

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

#### Scale pre-trial control customers, to math the pre-trial trial store customers

scalingFactoForControlCust <- preTrialMeasure[STORE_NBR == trial_store77 &
                                    YEARMONTH < 201902, sum(nCustomers)]/preTrialMeasure[
                                     STORE_NBR == control_store & YEARMONTH < 201902, sum(nCustomers)]

#### Apply the scaling factor
measureOverTimeCusts <- measureOverTime
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store,
][, controlCustomers := nCustomers*scalingFactoForControlCust
][, Store_type := ifelse(STORE_NBR == trial_store77, "Trial",
                  ifelse(STORE_NBR == control_store,"Control", "Other stores"))]

#### Calculate the percentage difference between scaled control sales and trial sales
percentageDiffCust <- merge(scaledControlCustomers[, c("YEARMONTH", "controlCustomers")],
                    measureOverTimeCusts[STORE_NBR == trial_store77, c("nCustomers", "YEARMONTH")],
                    by = "YEARMONTH")[, percentageDiff := abs(controlCustomers - nCustomers)/
controlCustomers]

# Let's again see if the difference is significant visually
```

```
#### As our null hypothesis is that the trial period is the same as the pre-trial period,
#### let's take the standard deviation based on the scaled percentage difference in the
#### pre-trial period
stdDev2 <- sd(percentageDiffCust[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 + stdDev2*2)][, Store_type := "Control 95th
% CI"]


#### Control store 5th percentile
pastCustomers_Controls5 <- pastCustomers[Store_type == "Control",][,
                                               nCusts := nCusts * (1 - stdDev2*2)][, Store_type := "Control 5th %
CI"]

trialAssessmentCust <- rbind(pastCustomers, pastCustomers_Controls95, pastCustomers_Controls5)

#### Plotting these in one nice graph
ggplot(trialAssessmentCust, aes(x = TransactionMonth, y = 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), show.legend = FALSE) +
  geom_line() + labs(x = "Month of operation", y = "Total number of customers",
                     title = "Total no. of customers per month")

## Trial Store 86
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 to calculate correlation for Sales and Total Customers
trial_store86 <- 86

corr_nSales86 <- calculateCorrelation(preTrialMeasure, quote(totSales), trial_store86)
corr_nCustomers86 <- calculateCorrelation(preTrialMeasure, quote(nCustomers), trial_store86)

#### Use the functions to calculate absolute measures for Sales and Total Customers
magnitude_nSales86 <- calculateMagnitudeDistance(preTrialMeasure, quote(totSales), trial_store86)
magnitude_nCustomer86 <- calculateMagnitudeDistance(preTrialMeasure, quote(nCustomers),
trial_store86)

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

#### Sales score based on correlation and magnitude
score_nSales86 <- merge(corr_nSales86,
```

```
                magnitude_nCustomer86,
                by = c("Store1", "Store2"))[, scoreNSales := (corr_measure * corr_weight) +
                                        (mag_measure * (1-corr_weight))]


#### Customer score based on correlation and magnitude
score_nCustomer86 <- merge(corr_nCustomers86, magnitude_nCustomer86,
                    by = c("Store1", "Store2"))[, scoreNCust := (corr_measure * corr_weight) +
                                            (mag_measure * (1-corr_weight))]



#### Combine scores across the drivers
score_Control86 <- merge(score_nSales86,
                    score_nCustomer86,
                    by = c("Store1", "Store2"))[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]



#### Select control stores based on the highest matching store
#### (closest to 1 but not the store itself, i.e. the second ranked highest store)
#### Select control store for trial store 86
control_store86 <- score_Control86[Store1 == trial_store86, ][order(-finalControlScore)][2, Store2]

control_store86   #155

#### Visual checks on trends based on the drivers
measureOverTimeSales <- measureOverTime
pastSales86 <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store86, "Trial",
```

```
                                        ifelse(STORE_NBR == control_store86, "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
ggplot(pastSales86, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line(aes(linetype = Store_type)) +
  labs(x = "Month", y = "Total sales",
       title = "Average total sales by month for stores 86/155 pre-trial")

# Let's check number of customers
measureOverTimeCusts <- measureOverTime
pastCustomers86 <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == trial_store86, "Trial",
                                        ifelse(STORE_NBR == control_store86, "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
ggplot(pastCustomers86, aes(TransactionMonth, numberCustomers, color = Store_type)) +
  geom_line(aes(linetype = Store_type)) +
  labs(x = "Month", y = "Number of customers", title = "Average number of customers by month for
stores 86/155")

#### Scale pre-trial control sales to match pre-trial trial store sales
scalingFactorForControlSales_86 <- preTrialMeasure[STORE_NBR == trial_store86 &
                                     YEARMONTH < 201902,
```

```
                                    sum(totSales)
]/preTrialMeasure[
  STORE_NBR == control_store86 &
    YEARMONTH < 201902, sum(totSales)]

#### Apply scaling factor to control store sales
measureOverTimeSales <- measureOverTime
scaledControlSales86 <- measureOverTimeSales[STORE_NBR == control_store86,
][, controlSales := totSales * scalingFactorForControlSales_86]



#### Calculate the percentage difference between scaled control sales and trial sales
percentageDiff86 <- merge(scaledControlSales86[, c("YEARMONTH", "controlSales")],
                          measureOverTime[STORE_NBR == trial_store86, c("YEARMONTH", "totSales")],
                          by = "YEARMONTH"
)[, percentageDiff := abs(controlSales - totSales)/controlSales]



#### Since 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.

stdDevSales_86 <- sd(percentageDiff86[, percentageDiff])

degreesOfFreedom <- 7

#### Trial and control store total sales
measureOverTimeSales <- measureOverTime
pastSales86 <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == trial_store86, "Trial",
```

```
                                        ifelse(STORE_NBR == control_store86, "Control", "Other stores"))
][, totSales := mean(totSales), by = c("YEARMONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
                                YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
][Store_type %in% c("Trial", "Control"), ]

#### Control store 95th percentile
pastSales86_Controls95 <- pastSales86[Store_type == "Control",
][, totSales := totSales * (1 + stdDevSales_86 * 2)
][, Store_type := "Control 95th % confidence interval"]


#### Control store 5th percentile
pastSales86_Controls5 <- pastSales86[Store_type == "Control",
][, totSales := totSales * (1 - stdDevSales_86 * 2)
][, Store_type := "Control 5th % confidence interval"]

#### Row bind pastSales86, Controls95, Controls5 together and call it trialAssessmentSales86
trialAssessmentSales86 <- rbind(pastSales86, pastSales86_Controls5, pastSales86_Controls95)


#### Plot these all in a nice graph
ggplot(trialAssessmentSales86, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_rect(data = trialAssessmentSales86[YEARMONTH < 201905 & YEARMONTH > 201901, ],
            aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
                ymin = 0, ymax = Inf, color = NULL), show.legend = FALSE) +
  geom_line(aes(linetype = Store_type)) +
  labs(x = "Month", y = "Average total sales", title = "Average total sales per month")
```

```
#### customer numbers
scalingFactorForControlCust_86 <- preTrialMeasure[STORE_NBR == trial_store86 &
                                    YEARMONTH < 201902, sum(nCustomers)
]/preTrialMeasure[STORE_NBR == control_store86 & YEARMONTH < 201902,
            sum(nCustomers)]

#### Apply scaling factor to the control customer numbers
measureOverTimeCusts <- measureOverTime
scaledControlCustomers86 <- measureOverTimeCusts[STORE_NBR == control_store86,
][, controlCustomers := nCustomers * scalingFactorForControlCust_86
][, Store_type := ifelse(Store_type == trial_store86, "Trial",
                  ifelse(Store_type == control_store86,
                      "Control", "Other stores"))]



#### Calculate the percentage difference
percentageDiffCust86 <- merge(scaledControlCustomers86[, c("YEARMONTH", "controlCustomers")],
                      measureOverTime[STORE_NBR == trial_store77, c("YEARMONTH", "nCustomers")],
                      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
stdDevCust_86 <- sd(percentageDiffCust86[YEARMONTH < 201902, percentageDiff])

degreesOfFreedom <- 7
```

```
#### Trial and control store number of customers
pastCustomers86 <- measureOverTimeCusts[, nCusts := mean(nCustomers), by = c("YEARMONTH",
"Store_type")
][Store_type %in% c("Trial", "Control"), ]



#### Control 95th percentile
pastCustomers86_Controls95 <- pastCustomers86[Store_type == "Control",
][, nCusts := nCusts * (1 + (stdDevCust_86 * 2))
][, Store_type := "Control 95th % confidence interval"]

#### Control 5th percentile
pastCustomers86_Controls5 <- pastCustomers86[Store_type == "Control",
][, nCusts := nCusts * (1 - (stdDevCust_86 * 2))
][, Store_type := "Control 5th % confidence interval"]

#### Row bind pastCustomers86, PastCustomers86_Controls95, PastCustomers_Controls5
trialAssessmentCust86 <- rbind(pastCustomers86, pastCustomers86_Controls5,
pastCustomers86_Controls95)

#### Visualize
ggplot(trialAssessmentCust86, aes(TransactionMonth, nCusts, color = Store_type)) +
 geom_rect(data = trialAssessmentCust86[YEARMONTH < 201905 & YEARMONTH > 201901, ],
         aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
             ymin = 0, ymax = Inf, color = NULL), show.legend = FALSE) +
 geom_line() +
 labs(x = "Month", y = "Average no. of customers", title = "Average no. of customers per month")
```

```
## 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)]

#### Use the functions to calculate correlation
trial_store <- 88

corr_nSales <- calculateCorrelation(preTrialMeasure, quote(totSales), trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasure, quote(nCustomers), trial_store)

#### Use the functions to calculate magnitude
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasure, quote(totSales), trial_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasure, quote(nCustomers),
trial_store)


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

score_NSales88 <- merge(corr_nSales, magnitude_nSales,
                        by = c("Store1", "Store2")
```

```
)[, scoreNSales := (corr_measure*corr_weight) +
   (mag_measure * (1 - corr_weight))]



score_NCustomers88 <- merge(corr_nCustomers, magnitude_nCustomers,
                            by = c("Store1", "Store2")
)[, scoreNCust :=(corr_measure * corr_weight) +
   (mag_measure * (1 - corr_weight))]



#### Combine scores across the drivers
score_Control88 <- merge(score_NSales88, score_NCustomers88, by = c("Store1", "Store2"))
score_Control88[, finalControlScore := (scoreNSales * 0.5) + (scoreNCust * 0.5)]

#### Select control stores based on the highest matching store (closest to 1 but not exactly 1)
#### Select control store for trial store 88
control_store88 <- score_Control88[order(-finalControlScore)][2, Store2]

control_store88 # 237

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

pastSales88 <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == 88, "Trial",
                                          ifelse(STORE_NBR == control_store88, "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, ]

ggplot(pastSales88, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line(aes(linetype = Store_type)) +
  labs(x = "Transaction Month", y = "Total sales per month",
       title = "Total sales per month Stores 88/237")




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

pastCustomers88 <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR == 88, "Trial",
                                                               ifelse(STORE_NBR == control_store88, "Control", "Other Stores"))
][, numberCustomers := mean(nCustomers), by = c("YEARMONTH", "Store_type")
][, TrsancationMonth := as.Date(paste(YEARMONTH %/% 100,
                                      YEARMONTH %% 100, 1, sep = "-"), "%Y-%m-%d")
][YEARMONTH < 201903]

ggplot(pastCustomers88, aes(TransactionMonth, numberCustomers, color = Store_type)) +
  geom_line() +
  labs(x = "Transaction Month", y = "Total customers per month",
       title = "Total customers per month Stores 88/237")




#### Scale pre-trial control store sales to match the pre-trial, trial store sales
scalingFactorForControlSales_88 <- preTrialMeasure[STORE_NBR == 88 &
```

```
                                    YEARMONTH < 201902, sum(totSales)]/preTrialMeasure[
                                     STORE_NBR == control_store88 & YEARMONTH < 201902,
                                     sum(totSales)]

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

scaledControlSales88 <- measureOverTimeSales[STORE_NBR == control_store88,
][, controlSales := totSales * scalingFactorForControlSales_88]

#### Calculate the percentage difference between the scaled control store sales, and trial store sales
percentageDiff88 <- merge(scaledControlSales88[, c("YEARMONTH", "controlSales")],
                          measureOverTime[STORE_NBR == 88, c("YEARMONTH", "totSales")], by = "YEARMONTH")

percentageDiff88[, percentageDiff := abs(controlSales - totSales)/controlSales]

#### 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.

stdDevSales_88 <- sd(percentageDiff88[YEARMONTH < 201902, percentageDiff])

degreesOfFreedom <- 7

#### Trial and control store total sales
measureOverTimeSales <- measureOverTime

pastSales88 <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR == 88, "Trial",
                                        ifelse(STORE_NBR == control_store88, "Control", "Other stores"))
```

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



#### Control store 95th percentile
pastSales88_Controls95 <- pastSales88[Store_type == "Control",
][, totSales := totSales * (1 + stdDevSales_88*2)
][, Store_type := "Control 95th % confidence interval"]

#### Control store 5th percentile
pastSales88_Controls5 <- pastSales88[Store_type == "Control",
][, totSales := totSales * (1 - stdDevSales_88*2)
][, Store_type := "Control 5th % confidence interval"]

trialAssessment88 <- rbind(pastSales88, pastSales88_Controls95, pastSales88_Controls5)

#### Visualize
ggplot(trialAssessment88, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_rect(data = trialAssessment88[YEARMONTH < 201905 & YEARMONTH > 201901, ],
            aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
                ymin = 0, ymax = Inf, color = NULL), show.legend = FALSE) +
  geom_line(aes(linetype = Store_type)) +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by month")
```

```
#### Scale pre-trial control customers to match pre-trial, trial store customers

ScalingFactorForControlCust_88 <- preTrialMeasure[STORE_NBR == 88 &
                                  YEARMONTH <  201902, sum(nCustomers)]/preTrialMeasure[
                                   STORE_NBR == control_store88 & YEARMONTH < 201902,
                                   sum(nCustomers)
                                  ]

#### Apply the scaling factor
measureOverTimeCusts <- measureOverTime
scaledControlCustomers88 <- measureOverTimeCusts[STORE_NBR == control_store88,
][, controlCustomers := nCustomers * ScalingFactorForControlCust_88
][, Store_type := ifelse(STORE_NBR == 88, "Trial",
                  ifelse(STORE_NBR == control_store88, "Control", "Other stores"))]


#### Calculate the percentage difference between scaled control sales, and trial sales
percentageDiffCust88 <- merge(scaledControlCustomers88[, c("YEARMONTH", "controlCustomers")],
                     measureOverTime[STORE_NBR == 88, c("YEARMONTH", "nCustomers")],
                     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
stdDevCust_88 <- sd(percentageDiffCust88[YEARMONTH < 201902, percentageDiff])
```

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

#### Trial and control store number of customers
pastCustomers88 <- measureOverTimeCusts[, nCusts := mean(nCustomers), by = c("YEARMONTH",
"Store_type")
][Store_type %in% c("Trial", "Control"), ]


#### Control store 95th percentile
pastCustomers88_Controls95 <- pastCustomers88[Store_type == "Control",
][, nCusts := nCusts * (1 + stdDevCust_88 * 2)
][, Store_type := "Control 95th % confidence interval"]

#### Control store 5th percentile
pastCustomers88_Controls5 <- pastCustomers88[Store_type == "Control",
][, nCusts := nCusts * (1 - stdDevCust_88 * 2)
][, Store_type := "Control 5th % confidence interval"]


trialAssessmentCust88 <- rbind(pastCustomers88, pastCustomers88_Controls5,
pastCustomers88_Controls95)

#### Plot them
ggplot(trialAssessmentCust88, aes(TransactionMonth, nCusts, color = Store_type)) +
 geom_rect(data = trialAssessmentCust88[YEARMONTH < 201905 & YEARMONTH > 201901, ],
        aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth),
           ymin = 0, ymax = Inf, color = NULL), show.legend = FALSE) +
 geom_line() +
```

```
labs(x = "Month of operation", y = "Total number of customers", title = "Total
    number of customers by month Stores 88/237")
```