# 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")