```
> library(readr)
> library(tidyverse)
> library(data.table)
> library(lubridate)
> data <- read_csv("Desktop/Quantium/QVI_data1.csv")
Rows: 264834 Columns: 12
── Column specification ────────────────────────────────────────────────────────
────────────────────────────────────────────────────────────────────────────────
──────────────────

Delimiter: ","
chr  (4): PROD_NAME, BRAND, LIFESTAGE, PREMIUM_CUSTOMER
dbl  (7): LYLTY_CARD_NBR, STORE_NBR, TXN_ID, PROD_NBR, PROD_QTY, TOT_SALES,
PACK_SIZE
date (1): DATE

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
>
> theme_set(theme_bw())
> theme_update(plot.title = element_text(hjust = 0.5))
> theme(panel.grid.major = element_blank(),
+       panel.grid.minor = element_blank())
List of 2
 $ panel.grid.major: list()
  ..- attr(*, "class")= chr [1:2] "element_blank" "element"
 $ panel.grid.minor: list()
  ..- attr(*, "class")= chr [1:2] "element_blank" "element"
 - attr(*, "class")= chr [1:2] "theme" "gg"
 - attr(*, "complete")= logi FALSE
 - attr(*, "validate")= logi TRUE
>
> #We would want to match trial stores to control stores that are similar to the trial
> #store prior to the trial period of Feb 2019 in terms of :
> # Monthly overall sales revenue
> # Monthly number of customers
> # Monthly number of transactions per customer
> data <- as.data.table(data)
> data[, YEARMONTH := format(as.Date(data$DATE), "%Y%m")]
> data$YEARMONTH <- as.numeric(as.character(data$YEARMONTH))
> #### Next, we define the measure calculations to use during the analysis.
> # For each store and month calculate total sales, number of customers,
> # transactions per customer, chips per customer and the average price per unit.
> measure_whole <- data[, .(SALES = sum(TOT_SALES),
+                 CUSTOMERS = uniqueN(LYLTY_CARD_NBR),
+                 TRAN_CUS= uniqueN(TXN_ID)/uniqueN(LYLTY_CARD_NBR),
+                 PROD_QTY_CUS= sum(PROD_QTY)/uniqueN(LYLTY_CARD_NBR),
+                 AVG_PRICE= 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
> storesWithFullObs <- unique(measure_whole[, .N, STORE_NBR][N == 12, STORE_NBR])
> preTrialMeasures <- measure_whole %>% filter(YEARMONTH < '201902' &
+                                STORE_NBR %in% storesWithFullObs,)
> calCorr <- function(inputTable,metricCol,trialStoreN){
+    calTable = data.table(Store1 = numeric(),
+                      Store2 = numeric(),
+                      corr_measure = numeric())
+    stN <- unique(inputTable[,STORE_NBR])
+    for(i in stN){
+       calMeasure = data.table("Store1" = trialStoreN,
+                           "Store2" = i,
+                           "corr_measure" = cor(inputTable[STORE_NBR == trialStoreN,
eval(metricCol)],
+                                          inputTable[STORE_NBR == i, eval(metricCol)]))
+       calTable <- rbind(calTable, calMeasure) }
+    return(calTable)
+ }
> #### Create a function to calculate a standardised magnitude distance for a measure
> # looping through each control store
> calculateMagnitudeDistance <- function(inputTable,metricCol,trialStoreN){
+    calTable = data.table(Store1 = numeric(),
+                      Store2 = numeric(),
+                      YEARMONTH = numeric(),
+                      mag_measure = numeric())
+    stN <- unique(inputTable[,STORE_NBR])
+    for(i in stN){
+       calMeasure = data.table("Store1" = trialStoreN,
+                           "Store2" = i,
+                           "YEARMONTH" = preTrialMeasures$YEARMONTH ,
+                           "mag_measure" = abs(inputTable[STORE_NBR == trialStoreN,
eval(metricCol)]-
+                                          inputTable[STORE_NBR == i, eval(metricCol)]))
+       calTable <- rbind(calTable,calMeasure)
+       calTable <- unique(calTable)
+    }
+    return(calTable)
+ }
> ###Standardize sales
> standMag<- function(magnitude) {
+   minMaxDist <- magnitude[, .(minDist = min(magnitude$mag_measure),
+                               maxDist = max(magnitude$mag_measure)),
+                             by = c("Store1", "YEARMONTH")]
+   distTable <- merge(magnitude, minMaxDist, by = c("Store1", "YEARMONTH"))
+   distTable[, magnitudeMeasure := 1 - (mag_measure - minDist)/(maxDist - minDist)]
+   finalDistTable <- distTable[, .(magN_measure = mean(magnitudeMeasure)), by = .(Store1,
Store2)]
+   return(finalDistTable)
+ }

##################################################################################
> #For Sales #77
> trialStoreN <- 77
> correlation <- calCorr(preTrialMeasures,quote(SALES), trialStoreN)
 Error in eval(jsub[[2L]], parent.frame(), parent.frame()) :
  object 'metricCol' not found
> correlation_sales <- calCorr(preTrialMeasures,quote(SALES), trialStoreN)
 Error in eval(jsub[[2L]], parent.frame(), parent.frame()) :
  object 'metricCol' not found
> ?quote
>
> calCorr <- function(inputTable,metricCol,trailStoreN){
+    calTable = data.table(Store1 = numeric(),
+                          Store2 = numeric(),
+                          corr_measure = numeric())
+    stN <- unique(inputTable[,STORE_NBR])
+    for(i in stN){
+       calMeasure = data.table("Store1" = trailStoreN,
+                               "Store2" = i,
+                               "corr_measure" = cor(inputTable[STORE_NBR == trialstoreN,
eval(metricCol)],
+                                                    inputTable[STORE_NBR == i, eval(metricCol)]))
+       calTable <- rbind(calTable, calMeasure) }
+    return(calTable)
+ }
> correlation_sales <- calCorr(preTrialMeasures,quote(SALES), trialStoreN)
 Error in eval(stub[[3L]], x, enclos) : object 'trialstoreN' not found
> calCorr <- function(inputTable,metricCol,trialStoreN){
+    calTable = data.table(Store1 = numeric(),
+                          Store2 = numeric(),
+                          corr_measure = numeric())
+    stN <- unique(inputTable[,STORE_NBR])
+    for(i in stN){
```

```
+       calMeasure = data.table("Store1" = trialStoreN,
+                               "Store2" = i,
+                               "corr_measure" = cor(inputTable[STORE_NBR == trialStoreN,
eval(metricCol)],
+                                                    inputTable[STORE_NBR == i, eval(metricCol)]))
+       calTable <- rbind(calTable, calMeasure) }
+    return(calTable)
+ }
> #### Create a function to calculate a standardised magnitude distance for a measure
> # looping through each control store
> calculateMagnitudeDistance <- function(inputTable,matricCol,trialStoreN){
+    calTable = data.table(Store1 = numeric(),
+                          Store2 = numeric(),
+                          YEARMONTH = numeric(),
+                          mag_measure = numeric())
+    stN <- unique(inputTable[,STORE_NBR])
+    for(i in stN){
+       calMeasure = data.table("Store1" = trialStoreN,
+                               "Store2" = i,
+                               "YEARMONTH" = preTrialMeasures$YEARMONTH ,
+                               "mag_measure" = abs(inputTable[STORE_NBR == trialStoreN,
eval(metricCol)]-
+                                                    inputTable[STORE_NBR == i, eval(metricCol)]))
+       calTable <- rbind(calTable,calMeasure)
+       calTable <- unique(calTable)
+    }
+    return(calTable)
+ }
> ###Standardize sales
> standMag<- function(magnitude) {
+   minMaxDist <- magnitude[, .(minDist = min(magnitude$mag_measure),
+                                maxDist = max(magnitude$mag_measure)),
+                            by = c("Store1", "YEARMONTH")]
+   distTable <- merge(magnitude, minMaxDist, by = c("Store1", "YEARMONTH"))
+   distTable[, magnitudeMeasure := 1 - (mag_measure - minDist)/(maxDist - minDist)]
+   finalDistTable <- distTable[, .(magN_measure = mean(magnitudeMeasure)), by = .(Store1,
Store2)]
+   return(finalDistTable)
+ }
>
####################################################################################
> #For Sales #77
> trialStoreN <- 77
> correlation_sales <- calCorr(preTrialMeasures,quote(SALES), trialStoreN)
> correlation_sales <- unique(correlation_sales)
> correlation_sales
    Store1 Store2 corr_measure
  1:     77      1   0.07521784
  2:     77      2  -0.26307873
  3:     77      3   0.80664364
  4:     77      4  -0.26329960
  5:     77      5  -0.11065231
 ---
256:     77    268   0.34475712
257:     77    269  -0.31573035
258:     77    270   0.31543042
259:     77    271   0.35548730
260:     77    272   0.11762158
> correlation_customers <- calCorr(preTrialMeasures,quote(CUSTOMERES),trialStoreN)
 Error in `[.data.table`(inputTable, STORE_NBR == i, eval(metricCol)) :
  j (the 2nd argument inside [...]) is a single symbol but column name 'CUSTOMERES' is not
found. Perhaps you intended DT[, ..CUSTOMERES]. This difference to data.frame is deliberate
and explained in FAQ 1.1.
> correlation_customers <- calCorr(preTrialMeasures,quote(CUSTOMERS),trialStoreN)
> correlation_customers
    Store1 Store2 corr_measure
  1:     77      1   0.32216828
  2:     77      2  -0.57205090
  3:     77      3   0.83420743
  4:     77      4  -0.29563870
  5:     77      5   0.37065851
 ---
256:     77    268   0.36951700
257:     77    269  -0.47429252
258:     77    270  -0.13125910
259:     77    271   0.01962906
260:     77    272   0.22321747
> mag_sales <- standMag(calculateMagnitudeDistance(preTrialMeasures, quote(SALES),
trialStoreN))
 Error in eval(jsub[[2L]], parent.frame(), parent.frame()) :
  object 'metricCol' not found
> #### Create a function to calculate a standardised magnitude distance for a measure
> # looping through each control store
> calculateMagnitudeDistance <- function(inputTable,metricCol,trialStoreN){
+   calTable = data.table(Store1 = numeric(),
+                         Store2 = numeric(),
+                         YEARMONTH = numeric(),
+                         mag_measure = numeric())
+    stN <- unique(inputTable[,STORE_NBR])
+    for(i in stN){
+       calMeasure = data.table("Store1" = trialStoreN,
+                              "Store2" = i,
+                              "YEARMONTH" = preTrialMeasures$YEARMONTH ,
+                              "mag_measure" = abs(inputTable[STORE_NBR == trialStoreN,
eval(metricCol)]-
+                                             inputTable[STORE_NBR == i, eval(metricCol)]))
+       calTable <- rbind(calTable,calMeasure)
+       calTable <- unique(calTable)
+    }
+    return(calTable)
+ }
> mag_sales <- standMag(calculateMagnitudeDistance(preTrialMeasures, quote(SALES),
trialStoreN))
> mag_sales[order(-magN_measure)] ###46
    Store1 Store2 magN_measure
  1:     77     77    1.0000000
  2:     77    233    0.9864766
  3:     77    255    0.9794790
  4:     77    188    0.9776629
  5:     77     53    0.9766779
 ---
256:     77     58    0.2325392
257:     77    165    0.2264803
258:     77    237    0.2130185
259:     77     88    0.2115205
260:     77    226    0.1237367
> mag_customers <- standMag(calculateMagnitudeDistance(preTrialMeasures,
quote(CUSTOMERS), trialStoreN))
> mag_customers
    Store1 Store2 magN_measure
  1:     77      1    0.9450549
  2:     77      2    0.9299451
  3:     77      3    0.3887363
  4:     77      4    0.2445055
  5:     77      5    0.5164835
 ---
256:     77    268    0.9436813
257:     77    269    0.3859890
258:     77    270    0.4024725
259:     77    271    0.5151099
260:     77    272    0.9519231
> mag_customers[order(-magN_measure)] ###46
```

```
    Store1 Store2 magN_measure
  1:     77     77    1.0000000
  2:     77    233    0.9931319
  3:     77     41    0.9766484
  4:     77    111    0.9684066
  5:     77    115    0.9670330
 ---
256:     77    165    0.2266484
257:     77     58    0.2101648
258:     77     88    0.2019231
259:     77    237    0.1950549
260:     77    226    0.1030220
> #average of the correlation and magnitude scores
> #### Hint: A simple average on the scores would be 0.5 * corr_measure + 0.5 *mag_measure
> corr_weight <- 0.5
> score_nSales <- merge(correlation_sales,mag_sales, by = c('Store2','Store1'))
> score_nSales <- score_nSales %>%
+   mutate(scoreNSales := (corr_weight*score_nSales$corr_sales)
+         + ((1-corr_weight) * score_nSales$magN_measure))
Error: Problem with `mutate()` column `scoreNSales`.
ℹ `scoreNSales = +...`.
ℹ `scoreNSales` must be size 260 or 1, not 0.
Run `rlang::last_error()` to see where the error occurred.
> score_nSales <- merge(correlation_sales,mag_sales, by = c('Store2','Store1')
+                   )[,scoreNSales := (corr_weight*score_nSales$corr_measure)
+                     + (1-corr_weight) * score_nSales$magN_measure]
> score_nCustomers <- merge(correlation_customers,mag_sales, by = c('Store2','Store1')
+                       )[, scoreNCust := corr_weight*corr_measure
+                         + (1-corr_weight) * magN_measure]
> ##Now we have a score for each of total number of sales and number of customers.
> ####  Combine scores across the drivers by first merging our sales scores and customer scores
into a single table
> score_Control <- merge(score_nSales, score_nCustomers, by = c('Store1','Store2'))
> score_Control[, finalControlScore := (scoreNSales * 0.5) + (scoreNCust * 0.5)]
> score_Control <- score_Control[Store2 != trialStoreN]
> Control_Store <- score_Control[finalControlScore == max(finalControlScore),]$Store2
> Control_Store  ###46
[1] 233
> #check visually if the drivers are indeed similar in the period before the trial.
> #### Visual checks on trends based on the drivers
> measureOverTime <- as.data.table(measure_whole)
> pastSales <- measureOverTime[, Store_type := ifelse(STORE_NBR == trialStoreN,"Trial",
+                                             ifelse(STORE_NBR == Control_Store,
+                                             "Control Store",
+                                  "Other Stores"))
+                    ][, mean_sales := mean(SALES), by = c("YEARMONTH","Store_type")
+                     ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
+                                                         YEARMONTH %% 100,
+                                                         1,
+                                                         sep = "-"),
+                                                     "%Y-%m-%d")
+                      ][YEARMONTH < 201902]
> ggplot(data = pastSales,
+       aes(x = TransactionMonth, y = mean_sales, color = Store_type)) +
+    geom_point() +
+    geom_line()+
+    labs(x = "Month of operation",
+        y = "Total sales ($)",
+        title = "Total sales by month of #77 Trial Store")+
+   theme(panel.grid.major = element_blank(),
+       panel.grid.minor = element_blank())
> pastCustomers <- measureOverTime[, Store_type := ifelse(STORE_NBR == trialStoreN,"Trial",
+                                  ifelse(STORE_NBR == Control_Store,
+                                      "Control Store",
+                                      "Other Stores"))
+                      ][, mean_customers :=mean(CUSTOMERS),
+                        by = c("YEARMONTH","Store_type")
+                      ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
+                                                          YEARMONTH %% 100,
+                                                          1,
+                                                          sep = "-"),
+                                                      "%Y-%m-%d")
+                      ][YEARMONTH < 201902]
> ggplot(data = pastCustomers,
+       aes(x = TransactionMonth, y = mean_customers, color = Store_type)) +
+    geom_point() +
+    geom_line()+
+    labs(x = "Month of operation",
+        y = "Total Customers",
+        title = "Total customers by month of #77 Trial Store")+
+   theme(panel.grid.major = element_blank(),
+       panel.grid.minor = element_blank())
> scalingFactorForControlSales <-
+    preTrialMeasures[STORE_NBR == trialStoreN, sum(SALES)]/
+    preTrialMeasures[STORE_NBR == Control_Store, sum(SALES)]
> scalingFactorForControlSales
[1] 1.023617
```

```
> #### Apply the scaling factor
> scaledControlSales <-
+    measureOverTime[STORE_NBR == Control_Store,
+                   ][ ,controlSales := SALES * scalingFactorForControlSales]
> ##### Calculate the percentage difference between scaled control sales and trial sales
> percentageDiff <- merge(scaledControlSales[, c("YEARMONTH", "controlSales")],
+                         measureOverTime[STORE_NBR == trialStoreN,c("mean_sales", "YEARMONTH")],
+                         by = 'YEARMONTH'
+                         )[, percentageDiff := abs(controlSales-mean_sales)/controlSales]
> percentageDiff
> stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
> stdDev
[1] 0.04994076
> #### Note that there are 8 months in the pre-trial period
> #### hence 8 - 1 = 7 degrees of freedom
> df <- 7
> #### We will test with a null hypothesis of there being 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
> #### to check whether the hypothesis is statistically significant.
> #### Hint: The test statistic here is (x - u)/standard deviation
> percentageDiff[ , tvalue := (percentageDiff - 0)/stdDev
+               ][ , TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
+                                                      YEARMONTH %% 100,
+                                                      1,
+                                                      sep = "-"),
+                                                "%Y-%m-%d")
+                 ][YEARMONTH < 201905 & YEARMONTH > 201901,
+                   .(TransactionMonth, tvalue)]
  TransactionMonth    tvalue
1:       2019-02-01  1.183534
2:       2019-03-01  7.339116
3:       2019-04-01 12.476373
> #### Also,find the 95th percentile of the t distribution with the appropriate degrees of
freedom to check whether the hypothesis is statistically significant.
> qt(0.95, df = df)
[1] 1.894579
> #### Trial and control store total sales
> #### Create new variables Store_type, totSales and TransactionMonth in the data table.
> pastSales <- measureOverTime[, Store_type := ifelse(STORE_NBR == trialStoreN, 'Trial',
+                                                     ifelse(STORE_NBR == Control_Store,
+                                                            'Control', 'Others'))
+                             ][, mean_sales:= mean(SALES),
+                                   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",
+                              ][, mean_sales := mean_sales * (1 + stdDev * 2)
+                               ][, Store_type := "Control 95th % confidence interval"]
> #### Control store 5th percentile
> pastSales_Controls5 <- pastSales[Store_type == "Control",
+                             ][, mean_sales := mean_sales * (1 - stdDev * 2)
+                              ][, Store_type := "Control 5th % confidence interval"]
> trialAssessment <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)
> #### Plotting these in one nice graph
> ggplot(trialAssessment, aes(TransactionMonth, mean_sales, 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 of Trial Store #77") +
+   theme(panel.grid.major = element_blank(),
+        panel.grid.minor = element_blank())
> scalingFactorForControlCustomers<-
+    preTrialMeasures[STORE_NBR == trialStoreN, sum(CUSTOMERS)]/
+    preTrialMeasures[STORE_NBR == Control_Store, sum(CUSTOMERS)]
> scalingFactorForControlCustomers
[1] 1.003356
> scaledControlCustomers <-
+    measureOverTime[STORE_NBR == Control_Store,
+    ][ ,controlCustomers := CUSTOMERS * scalingFactorForControlSales]
> percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH", "controlCustomers")],
+                      measureOverTime[STORE_NBR == trialStoreN,c("CUSTOMERS",
"YEARMONTH")],
+                      by = 'YEARMONTH'
+                      )[, percentageDiff := abs(CUSTOMERS-controlCustomers)/controlCustomers]
> percentageDiff
> stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
> stdDev
[1] 0.01466774
> df <- 7
> percentageDiff[ , tvalue := (percentageDiff - 0)/stdDev
+               ][ , TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
+                                  YEARMONTH %% 100,
+                                  1,
+                                  sep = "-"),
+                              "%Y-%m-%d")
+                 ][YEARMONTH < 201905 & YEARMONTH > 201901,
+                  .(TransactionMonth, tvalue)]
  TransactionMonth    tvalue
1:       2019-02-01  1.573002
2:       2019-03-01 15.077949
3:       2019-04-01 36.169155
> qt(0.95, df = df)
[1] 1.894579
> pastCustomers <- measureOverTime[, Store_type := ifelse(STORE_NBR == trialStoreN, 'Trial',
+                                          ifelse(STORE_NBR == Control_Store,
+                                              'Control', 'Others'))
+                               ][, mean_customers := mean(CUSTOMERS),
+                                 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
> pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",
+                                    ][, mean_customers := mean_customers * (1 + stdDev * 2)
+                                     ][, Store_type := "Control 95th % confidence interval"]
> #### Control store 5th percentile
> pastCustomers_Controls5 <- pastCustomers[Store_type == "Control",
+                                   ][, mean_customers := mean_customers * (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, mean_customers, 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 = "Number of Customers",
+         title = "Number of Customers by month of Trial Store #77")+
+   theme(panel.grid.major = element_blank(),
+        panel.grid.minor = element_blank())
>
################################################################################
> #For Sales #86
> trialStoreN <- 86
> correlation_sales <- calCorr(preTrialMeasures,quote(SALES), trialStoreN)
> correlation_sales <- unique(correlation_sales)
> correlation_sales
    Store1 Store2 corr_measure
  1:     86      1  0.445631778
  2:     86      2 -0.403834784
  3:     86      3 -0.261283895
  4:     86      4 -0.039035475
  5:     86      5  0.235159195
 ---
256:     86    268 -0.452181631
257:     86    269  0.697055188
258:     86    270 -0.730678772
259:     86    271  0.527637266
260:     86    272  0.004925858
> correlation_customers <- calCorr(preTrialMeasures,quote(CUSTOMERS),trialStoreN)
> correlation_customers
    Store1 Store2 corr_measure
  1:     86      1   0.48583112
  2:     86      2  -0.08616076
  3:     86      3  -0.35378564
  4:     86      4  -0.16960837
  5:     86      5  -0.25322924
 ---
256:     86    268  -0.03427345
257:     86    269  -0.09858652
258:     86    270  -0.76726726
259:     86    271   0.26739300
260:     86    272  -0.35381524
> mag_sales <- standMag(calculateMagnitudeDistance(preTrialMeasures, quote(SALES),
trialStoreN))
```

```
> mag_sales[order(-magN_measure)] ###233
    Store1 Store2 magN_measure
  1:     86     86   1.00000000
  2:     86    109   0.96657804
  3:     86    155   0.96567169
  4:     86    222   0.96207647
  5:     86    225   0.96030907
 ---
256:     86    146   0.09943504
257:     86    198   0.09901207
258:     86    140   0.09819635
259:     86    177   0.09576429
260:     86     99   0.09192737
> mag_customers <- standMag(calculateMagnitudeDistance(preTrialMeasures,
quote(CUSTOMERS), trialStoreN))
> mag_customers[order(-magN_measure)] ###233
    Store1 Store2 magN_measure
  1:     86     86    1.0000000
  2:     86    155    0.9867725
  3:     86    109    0.9695767
  4:     86    225    0.9695767
  5:     86    229    0.9642857
 ---
256:     86    244    0.1190476
257:     86    146    0.1177249
258:     86     99    0.1164021
259:     86    258    0.1137566
260:     86    198    0.1124339
> corr_weight <- 0.5
> score_nSales <- merge(correlation_sales,mag_sales, by = c('Store2','Store1')
+                     )[,scoreNSales := (corr_weight*score_nSales$corr_measure)
+                       + (1-corr_weight) * score_nSales$magN_measure]
> score_nCustomers <- merge(correlation_customers,mag_sales, by = c('Store2','Store1')
+                         )[, scoreNCust := corr_weight*corr_measure
+                           + (1-corr_weight) * magN_measure]
> score_Control <- merge(score_nSales, score_nCustomers, by = c('Store1','Store2'))
> score_Control[, finalControlScore := (scoreNSales * 0.5) + (scoreNCust * 0.5)]
> score_Control <- score_Control[Store2 != trialStoreN]
> Control_Store <- score_Control[finalControlScore == max(finalControlScore),]$Store2
> Control_Store
[1] 184
>
################################################################################
> #For Sales #86
> trialStoreN <- 86
> correlation_sales <- calCorr(preTrialMeasures,quote(SALES), trialStoreN)
> correlation_sales <- unique(correlation_sales)
> correlation_sales
     Store1 Store2 corr_measure
  1:     86      1  0.445631778
  2:     86      2 -0.403834784
  3:     86      3 -0.261283895
  4:     86      4 -0.039035475
  5:     86      5  0.235159195
 ---
256:     86    268 -0.452181631
257:     86    269  0.697055188
258:     86    270 -0.730678772
259:     86    271  0.527637266
260:     86    272  0.004925858
> correlation_customers <- calCorr(preTrialMeasures,quote(CUSTOMERS),trialStoreN)
> correlation_customers
     Store1 Store2 corr_measure
  1:     86      1   0.48583112
  2:     86      2  -0.08616076
  3:     86      3  -0.35378564
  4:     86      4  -0.16960837
  5:     86      5  -0.25322924
 ---
256:     86    268  -0.03427345
257:     86    269  -0.09858652
258:     86    270  -0.76726726
259:     86    271   0.26739300
260:     86    272  -0.35381524
> mag_sales <- standMag(calculateMagnitudeDistance(preTrialMeasures, quote(SALES),
trialStoreN))
> mag_sales[order(-magN_measure)] ###109
     Store1 Store2 magN_measure
  1:     86     86   1.00000000
  2:     86    109   0.96657804
  3:     86    155   0.96567169
  4:     86    222   0.96207647
  5:     86    225   0.96030907
 ---
256:     86    146   0.09943504
257:     86    198   0.09901207
258:     86    140   0.09819635
259:     86    177   0.09576429
```

```
260:    86    99   0.09192737
> mag_customers <- standMag(calculateMagnitudeDistance(preTrialMeasures,
quote(CUSTOMERS), trialStoreN))
> mag_customers[order(-magN_measure)] ###155
    Store1 Store2 magN_measure
  1:    86     86    1.0000000
  2:    86    155    0.9867725
  3:    86    109    0.9695767
  4:    86    225    0.9695767
  5:    86    229    0.9642857
 ---
256:    86    244    0.1190476
257:    86    146    0.1177249
258:    86     99    0.1164021
259:    86    258    0.1137566
260:    86    198    0.1124339
> corr_weight <- 0.5
> score_nSales <- merge(correlation_sales,mag_sales, by = c('Store2','Store1')
+                   )[,scoreNSales := (corr_weight*score_nSales$corr_measure)
+                     + (1-corr_weight) * score_nSales$magN_measure]
> score_nCustomers <- merge(correlation_customers,mag_sales, by = c('Store2','Store1')
+                       )[, scoreNCust := corr_weight*corr_measure
+                         + (1-corr_weight) * magN_measure]
> score_Control <- merge(score_nSales, score_nCustomers, by = c('Store1','Store2'))
> score_Control[, finalControlScore := (scoreNSales * 0.5) + (scoreNCust * 0.5)]
> score_Control <- score_Control[Store2 != trialStoreN]
> Control_Store <- score_Control[finalControlScore == max(finalControlScore),]$Store2
> Control_Store  ###184
[1] 155
> measureOverTime <- as.data.table(measure_whole)
> pastSales <- measureOverTime[, Store_type := ifelse(STORE_NBR == trialStoreN,"Trial",
+                                               ifelse(STORE_NBR == Control_Store,
+                                                      "Control Store",
+                                                      "Other Stores"))
+                         ][, mean_sales := mean(SALES), by = c("YEARMONTH","Store_type")
+                          ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
+                                YEARMONTH %% 100,
+                                1,
+                                sep = "-"),
+                          "%Y-%m-%d")
+                          ][YEARMONTH < 201902]
> ggplot(data = pastSales,
+       aes(x = TransactionMonth, y = mean_sales, color = Store_type)) +
+   geom_point() +
+   geom_line()+
+   labs(x = "Month of operation",
+        y = "Total sales ($)",
+        title = "Total sales by month of #86 Trial Store")+
+   theme(panel.grid.major = element_blank(),
+         panel.grid.minor = element_blank())
> pastCustomers <- measureOverTime[, Store_type := ifelse(STORE_NBR == trialStoreN,"Trial",
+                                                  ifelse(STORE_NBR == Control_Store,
+                                                         "Control Store",
+                                                         "Other Stores"))
+                                  ][, mean_customers :=mean(CUSTOMERS),
+                                    by = c("YEARMONTH","Store_type")
+                                    ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
+                                      YEARMONTH %% 100,
+                                      1,
+                                      sep = "-"),
+                                 "%Y-%m-%d")
+                                 ][YEARMONTH < 201902]
> ggplot(data = pastCustomers,
+        aes(x = TransactionMonth, y = mean_customers, color = Store_type)) +
+   geom_point() +
+   geom_line()+
+   labs(x = "Month of operation",
+        y = "Total Customers",
+        title = "Total customers by month of #86 Trial Store")+
+   theme(panel.grid.major = element_blank(),
+         panel.grid.minor = element_blank())
> scalingFactorForControlSales <-
+   preTrialMeasures[STORE_NBR == trialStoreN, sum(SALES)]/
+   preTrialMeasures[STORE_NBR == Control_Store, sum(SALES)]
> scalingFactorForControlSales
[1] 0.9700651
> scaledControlSales <-
+   measureOverTime[STORE_NBR == Control_Store,
+   ][ ,controlSales := SALES * scalingFactorForControlSales]
> percentageDiff <- merge(scaledControlSales[, c("YEARMONTH", "controlSales")],
+                         measureOverTime[STORE_NBR == trialStoreN,c("mean_sales", "YEARMONTH")],
+                         by = 'YEARMONTH'
+                         )[, percentageDiff := abs(controlSales-mean_sales)/controlSales]
> percentageDiff
   YEARMONTH controlSales mean_sales percentageDiff
 1:    201807     896.9222     892.20    0.005264934
 2:    201808     759.2700     764.05    0.006295532
 3:    201809     984.0341     914.60    0.070560652
 4:   201810    934.9488    948.40   0.014387109
 5:   201811    871.8946    918.00   0.052879611
 6:   201812    824.3614    841.20   0.020426281
 7:   201901    848.4190    841.40   0.008273010
 8:   201902    864.5221    913.20   0.056306186
 9:   201903    780.3204   1026.80   0.315869729
10:    201904    819.3170    848.20   0.035252503
11:    201905    895.2246    889.30   0.006618028
12:    201906    831.5398    838.00   0.007768906
> stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
> stdDev
[1] 0.02583395
> percentageDiff[ , tvalue := (percentageDiff - 0)/stdDev
+              ][ , TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
+                                   YEARMONTH %% 100,
+                                   1,
+                                   sep = "-"),
+                              "%Y-%m-%d")
+                ][YEARMONTH < 201905 & YEARMONTH > 201901,
+                 .(TransactionMonth, tvalue)]
  TransactionMonth    tvalue
1:      2019-02-01  2.179542
2:      2019-03-01 12.226922
3:      2019-04-01  1.364580
> qt(0.95, df = df)
[1] 1.894579
> pastSales <- measureOverTime[, Store_type := ifelse(STORE_NBR == trialStoreN, 'Trial',
+                                             ifelse(STORE_NBR == Control_Store,
+                                                    'Control', 'Others'))
+                           ][, mean_sales:= mean(SALES),
+                             by = c('YEARMONTH','Store_type')
+                             ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
+                                  YEARMONTH %% 100,
+                                  1,
+                                  sep = "-"),
+                             "%Y-%m-%d")
+                             ][Store_type %in% c("Trial", "Control"), ]
> pastSales_Controls95 <- pastSales[Store_type == "Control",
+                               ][, mean_sales := mean_sales * (1 + stdDev * 2)
+                                 ][, Store_type := "Control 95th % confidence interval"]
> pastSales_Controls5 <- pastSales[Store_type == "Control",
+                              ][, mean_sales := mean_sales * (1 - stdDev * 2)
+                                ][, Store_type := "Control 5th % confidence interval"]
> trialAssessment <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)
> ggplot(trialAssessment, aes(TransactionMonth, mean_sales, 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 of Trial Store #86") +
+   theme(panel.grid.major = element_blank(),
+         panel.grid.minor = element_blank())
> scalingFactorForControlCustomers<-
+   preTrialMeasures[STORE_NBR == trialStoreN, sum(CUSTOMERS)]/
+   preTrialMeasures[STORE_NBR == Control_Store, sum(CUSTOMERS)]
> scalingFactorForControlCustomers
[1] 1
> scalingFactorForControlCustomers<-
+   preTrialMeasures[STORE_NBR == trialStoreN, sum(CUSTOMERS)]/
+   preTrialMeasures[STORE_NBR == Control_Store, sum(CUSTOMERS)]
> scalingFactorForControlCustomers
[1] 1
> scaledControlCustomers <-
+   measureOverTime[STORE_NBR == Control_Store,
+   ][ ,controlCustomers := CUSTOMERS * scalingFactorForControlSales]
> scalingFactorForControlCustomers<-
+   preTrialMeasures[STORE_NBR == trialStoreN, sum(CUSTOMERS)]/
+   preTrialMeasures[STORE_NBR == Control_Store, sum(CUSTOMERS)]
> scalingFactorForControlCustomers
[1] 1
> scaledControlCustomers <-
+   measureOverTime[STORE_NBR == Control_Store,
+   ][ ,controlCustomers := CUSTOMERS * scalingFactorForControlSales]
> percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH", "controlCustomers")],
+                         measureOverTime[STORE_NBR == trialStoreN,c("CUSTOMERS",
"YEARMONTH")],
+                         by = 'YEARMONTH'
+ )[, percentageDiff := abs(CUSTOMERS-controlCustomers)/controlCustomers]
> percentageDiff
> stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
> stdDev
[1] 0.01975041
> df <- 7
> percentageDiff[ , tvalue := (percentageDiff - 0)/stdDev
+ ][ , TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
+                                        YEARMONTH %% 100,
+                                        1,
+                                        sep = "-"),
+                                  "%Y-%m-%d")
+ ][YEARMONTH < 201905 & YEARMONTH > 201901,
+   .(TransactionMonth, tvalue)]
   TransactionMonth    tvalue
1:       2019-02-01  8.155393
2:       2019-03-01 13.222857
3:       2019-04-01  4.725720
> qt(0.95, df = df)
[1] 1.894579
> pastCustomers <- measureOverTime[, Store_type := ifelse(STORE_NBR == trialStoreN, 'Trial',
+                                                         ifelse(STORE_NBR == Control_Store,
+                                                                'Control', 'Others'))
+ ][, mean_customers := mean(CUSTOMERS),
+   by = c('YEARMONTH','Store_type')
+ ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
+                                       YEARMONTH %% 100,
+                                       1,
+                                       sep = "-"),
+                                 "%Y-%m-%d")
+ ][Store_type %in% c("Trial", "Control"), ]
> pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",
+ ][, mean_customers := mean_customers * (1 + stdDev * 2)
+ ][, Store_type := "Control 95th % confidence interval"]
> pastCustomers_Controls5 <- pastCustomers[Store_type == "Control",
+ ][, mean_customers := mean_customers * (1 - stdDev * 2)
+ ][, Store_type := "Control 5th % confidence interval"]
> trialAssessment <- rbind(pastCustomers, pastCustomers_Controls95,
pastCustomers_Controls5)
> ggplot(trialAssessment, aes(TransactionMonth, mean_customers, 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 = "Number of Customers",
+        title = "Number of Customers by month of Trial Store #86")+
+   theme(panel.grid.major = element_blank(),
+         panel.grid.minor = element_blank())
> #For Sales #88
> trialStoreN <- 88
> correlation_sales <- calCorr(preTrialMeasures,quote(SALES), trialStoreN)
> correlation_sales <- unique(correlation_sales)
> correlation_sales
     Store1 Store2 corr_measure
  1:     88      1   0.81363605
  2:     88      2  -0.06792668
  3:     88      3  -0.50784733
  4:     88      4  -0.74556612
  5:     88      5   0.19033025
 ---
256:     88    268  -0.02142876
257:     88    269  -0.17257761
258:     88    270  -0.72327202
259:     88    271  -0.10303745
260:     88    272  -0.77277243
> correlation_customers <- calCorr(preTrialMeasures,quote(CUSTOMERS),trialStoreN)
> correlation_customers
     Store1 Store2 corr_measure
  1:     88      1   0.30533393
  2:     88      2  -0.45237858
  3:     88      3   0.52288362
  4:     88      4  -0.36150269
  5:     88      5  -0.02532022
 ---
256:     88    268   0.67267172
257:     88    269  -0.27478061
258:     88    270  -0.10303173
259:     88    271  -0.01883076
260:     88    272   0.02690878
> mag_sales <- standMag(calculateMagnitudeDistance(preTrialMeasures, quote(SALES),
trialStoreN))
> mag_sales[order(-magN_measure)] ###109
     Store1 Store2 magN_measure
  1:     88     88   1.00000000
  2:     88    237   0.95859653
  3:     88    203   0.95316043
  4:     88     40   0.94116456
  5:     88    199   0.92597992
 ---
256:     88    146   0.06610382
```

```
257:    88   198   0.06582038
258:    88   140   0.06527373
259:    88   177   0.06364391
260:    88    99   0.06107264
> mag_customers <- standMag(calculateMagnitudeDistance(preTrialMeasures,
quote(CUSTOMERS), trialStoreN))
> mag_customers[order(-magN_measure)] ###155
    Store1 Store2 magN_measure
  1:    88    88   1.00000000
  2:    88   237   0.98781838
  3:    88   203   0.94462901
  4:    88    40   0.94241417
  5:    88   165   0.93576966
 ---
256:    88   244   0.05980066
257:    88   146   0.05869324
258:    88    99   0.05758583
259:    88   258   0.05537099
260:    88   198   0.05426357
> corr_weight <- 0.5
> score_nSales <- merge(correlation_sales,mag_sales, by = c('Store2','Store1')
+ )[,scoreNSales := (corr_weight*score_nSales$corr_measure)
+   + (1-corr_weight) * score_nSales$magN_measure]
> score_nCustomers <- merge(correlation_customers,mag_sales, by = c('Store2','Store1')
+ )[, scoreNCust := corr_weight*corr_measure
+   + (1-corr_weight) * magN_measure]
> score_Control <- merge(score_nSales, score_nCustomers, by = c('Store1','Store2'))
> score_Control[, finalControlScore := (scoreNSales * 0.5) + (scoreNCust * 0.5)]
> score_Control <- score_Control[Store2 != trialStoreN]
> Control_Store <- score_Control[finalControlScore == max(finalControlScore),]$Store2
> Control_Store  ###155
[1] 178
> corr_weight <- 0.5
> score_nSales <- merge(correlation_sales,mag_sales, by = c('Store2','Store1')
+                     )[,scoreNSales := (corr_weight*score_nSales$corr_measure)
+                       + (1-corr_weight) * score_nSales$magN_measure]
> score_nCustomers <- merge(correlation_customers,mag_sales, by = c('Store2','Store1')
+                         )[, scoreNCust := corr_weight*corr_measure
+                           + (1-corr_weight) * magN_measure]
> score_Control <- merge(score_nSales, score_nCustomers, by = c('Store1','Store2'))
> score_Control[, finalControlScore := (scoreNSales * 0.5) + (scoreNCust * 0.5)]
> score_Control <- score_Control[Store2 != trialStoreN]
> Control_Store <- score_Control[finalControlScore == max(finalControlScore),]$Store2
> Control_Store  ###155
```

```
[1] 237
> measureOverTime <- as.data.table(measure_whole)
> pastSales <- measureOverTime[, Store_type := ifelse(STORE_NBR == trialStoreN,"Trial",
+                            ifelse(STORE_NBR == Control_Store,
+                                "Control Store",
+                                "Other Stores"))
+ ][, mean_sales := mean(SALES), by = c("YEARMONTH","Store_type")
+ ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
+                     YEARMONTH %% 100,
+                     1,
+                     sep = "-"),
+                  "%Y-%m-%d")
+ ][YEARMONTH < 201902]
> ggplot(data = pastSales,
+     aes(x = TransactionMonth, y = mean_sales, color = Store_type)) +
+   geom_point() +
+   geom_line()+
+   labs(x = "Month of operation",
+     y = "Total sales ($)",
+     title = "Total sales by month of #88 Trial Store")+
+   theme(panel.grid.major = element_blank(),
+      panel.grid.minor = element_blank())
> pastCustomers <- measureOverTime[, Store_type := ifelse(STORE_NBR == trialStoreN,"Trial",
+                              ifelse(STORE_NBR == Control_Store,
+                                  "Control Store",
+                                  "Other Stores"))
+ ][, mean_customers :=mean(CUSTOMERS),
+   by = c("YEARMONTH","Store_type")
+ ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
+                     YEARMONTH %% 100,
+                     1,
+                     sep = "-"),
+                  "%Y-%m-%d")
+ ][YEARMONTH < 201902]
> ggplot(data = pastCustomers,
+     aes(x = TransactionMonth, y = mean_customers, color = Store_type)) +
+   geom_point() +
+   geom_line()+
+   labs(x = "Month of operation",
+     y = "Total Customers",
+     title = "Total customers by month of #88 Trial Store")+
+   theme(panel.grid.major = element_blank(),
+      panel.grid.minor = element_blank())
```

```
> scalingFactorForControlSales <-
+   preTrialMeasures[STORE_NBR == trialStoreN, sum(SALES)]/
+   preTrialMeasures[STORE_NBR == Control_Store, sum(SALES)]
> scalingFactorForControlSales
[1] 1.001558
> scaledControlSales <-
+   measureOverTime[STORE_NBR == Control_Store,
+   ][ ,controlSales := SALES * scalingFactorForControlSales]
> percentageDiff <- merge(scaledControlSales[, c("YEARMONTH", "controlSales")],
+                         measureOverTime[STORE_NBR == trialStoreN,c("mean_sales", "YEARMONTH")],
+                         by = 'YEARMONTH'
+ )[, percentageDiff := abs(controlSales-mean_sales)/controlSales]
> percentageDiff
    YEARMONTH controlSales mean_sales percentageDiff
 1:    201807     1450.657    1310.00    0.096960948
 2:    201808     1369.931    1323.80    0.033674301
 3:    201809     1324.260    1423.00    0.074562052
 4:    201810     1350.401    1352.40    0.001480229
 5:    201811     1399.778    1382.80    0.012129012
 6:    201812     1266.971    1325.20    0.045958983
 7:    201901     1221.601    1266.40    0.036672625
 8:    201902     1406.989    1370.20    0.026147425
 9:    201903     1210.083    1477.20    0.220742936
10:    201904     1206.477    1439.40    0.193060293
11:    201905     1201.169    1308.25    0.089147408
12:    201906     1155.398    1354.60    0.172410168
> stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
> stdDev
[1] 0.03346787
> percentageDiff[ , tvalue := (percentageDiff - 0)/stdDev
+ ][ , TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
+                                        YEARMONTH %% 100,
+                                        1,
+                                        sep = "-"),
+                                  "%Y-%m-%d")
+ ][YEARMONTH < 201905 & YEARMONTH > 201901,
+   .(TransactionMonth, tvalue)]
   TransactionMonth    tvalue
1:       2019-02-01 0.7812695
2:       2019-03-01 6.5956678
3:       2019-04-01 5.7685269
> qt(0.95, df = df)
[1] 1.894579
> pastSales <- measureOverTime[, Store_type := ifelse(STORE_NBR == trialStoreN, 'Trial',
+                                          ifelse(STORE_NBR == Control_Store,
+                                                'Control', 'Others'))
+ ][, mean_sales:= mean(SALES),
+   by = c('YEARMONTH','Store_type')
+ ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
+                                YEARMONTH %% 100,
+                                1,
+                                sep = "-"),
+                          "%Y-%m-%d")
+ ][Store_type %in% c("Trial", "Control"), ]
> pastSales_Controls95 <- pastSales[Store_type == "Control",
+ ][, mean_sales := mean_sales * (1 + stdDev * 2)
+ ][, Store_type := "Control 95th % confidence interval"]
> pastSales_Controls5 <- pastSales[Store_type == "Control",
+ ][, mean_sales := mean_sales * (1 - stdDev * 2)
+ ][, Store_type := "Control 5th % confidence interval"]
> trialAssessment <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)
> ggplot(trialAssessment, aes(TransactionMonth, mean_sales, 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 of Trial Store #88") +
+   theme(panel.grid.major = element_blank(),
+        panel.grid.minor = element_blank())
> scalingFactorForControlCustomers<-
+   preTrialMeasures[STORE_NBR == trialStoreN, sum(CUSTOMERS)]/
+   preTrialMeasures[STORE_NBR == Control_Store, sum(CUSTOMERS)]
> scalingFactorForControlCustomers
[1] 0.9943503
> scaledControlCustomers <-
+   measureOverTime[STORE_NBR == Control_Store,
+   ][ ,controlCustomers := CUSTOMERS * scalingFactorForControlSales]
> percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH", "controlCustomers")],
+                    measureOverTime[STORE_NBR == trialStoreN,c("CUSTOMERS",
"YEARMONTH")],
+                    by = 'YEARMONTH'
+ )[, percentageDiff := abs(CUSTOMERS-controlCustomers)/controlCustomers]
> percentageDiff
> stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
> stdDev
[1] 0.01057889
> df <- 7
> percentageDiff[ , tvalue := (percentageDiff - 0)/stdDev
+ ][ , TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
+                                        YEARMONTH %% 100,
+                                        1,
+                                        sep = "-"),
+                                  "%Y-%m-%d")
+ ][YEARMONTH < 201905 & YEARMONTH > 201901,
+   .(TransactionMonth, tvalue)]
   TransactionMonth    tvalue
1:       2019-02-01  1.645185
2:       2019-03-01 11.749667
3:       2019-04-01  6.144979
> qt(0.95, df = df)
[1] 1.894579
> pastCustomers <- measureOverTime[, Store_type := ifelse(STORE_NBR == trialStoreN, 'Trial',
+                                                   ifelse(STORE_NBR == Control_Store,
+                                                          'Control', 'Others'))
+ ][, mean_customers := mean(CUSTOMERS),
+   by = c('YEARMONTH','Store_type')
+ ][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100,
+                                       YEARMONTH %% 100,
+                                       1,
+                                       sep = "-"),
+                                 "%Y-%m-%d")
+ ][Store_type %in% c("Trial", "Control"), ]
> pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",
+ ][, mean_customers := mean_customers * (1 + stdDev * 2)
+ ][, Store_type := "Control 95th % confidence interval"]
> pastCustomers_Controls5 <- pastCustomers[Store_type == "Control",
+ ][, mean_customers := mean_customers * (1 - stdDev * 2)
+ ][, Store_type := "Control 5th % confidence interval"]
> trialAssessment <- rbind(pastCustomers, pastCustomers_Controls95,
pastCustomers_Controls5)
> ggplot(trialAssessment, aes(TransactionMonth, mean_customers, 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 = "Number of Customers",
+      title = "Number of Customers by month of Trial Store #88")+
+   theme(panel.grid.major = element_blank(),
+       panel.grid.minor = element_blank())
```