part2

kiran jangili

2024-08-08

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


filePath <- "E:/kiran jangili/Data analysis/Quantium/QVI_data.csv"
data <- fread(filePath)

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

Add a new month ID column in the data with the format yyyymm

```
data[, YEARMONTH := year(DATE) * 100 + month(DATE)]
```

Calculate total sales, number of customers, transactions per customer, chips per customer, and average price per unit for each store and month

```
measureOverTime <- data[, .(
  totSales = sum(TOT_SALES),
  nCustomers = uniqueN(LYLTY_CARD_NBR),
  nTxnPerCust = .N / uniqueN(LYLTY_CARD_NBR),
  nChipsPerTxn = sum(PROD_QTY) / .N,
  avgPricePerUnit = mean(TOT_SALES / PROD_QTY)
), by = .(STORE_NBR, YEARMONTH)][order(STORE_NBR, YEARMONTH)]
```

Filter to the pre-trial period and stores with full observation periods

```
storesWithFullObs <- unique(measureOverTime[, .N, by = STORE_NBR][N == 12,
STORE_NBR])
preTrialMeasures <- measureOverTime[YEARMONTH < 201902 & STORE_NBR %in%
storesWithFullObs]
```

Create a function to calculate correlation for a measure

```
calculateCorrelation<- # Create a function to calculate correlation for a
measure
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 standardised magnitude distance for a measure

```
# Create a function to calculate a standardised magnitude distance for a
measure
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)
  }

  # Standardise the magnitude distance
  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)
}
```

Use the function you created to calculate correlations and magnitude distances against store 77 using total sales and number of customers

```
trial_store <- 77
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales),
trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers),
trial_store)
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures,
quote(totSales), trial_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures,
quote(nCustomers), trial_store)
```

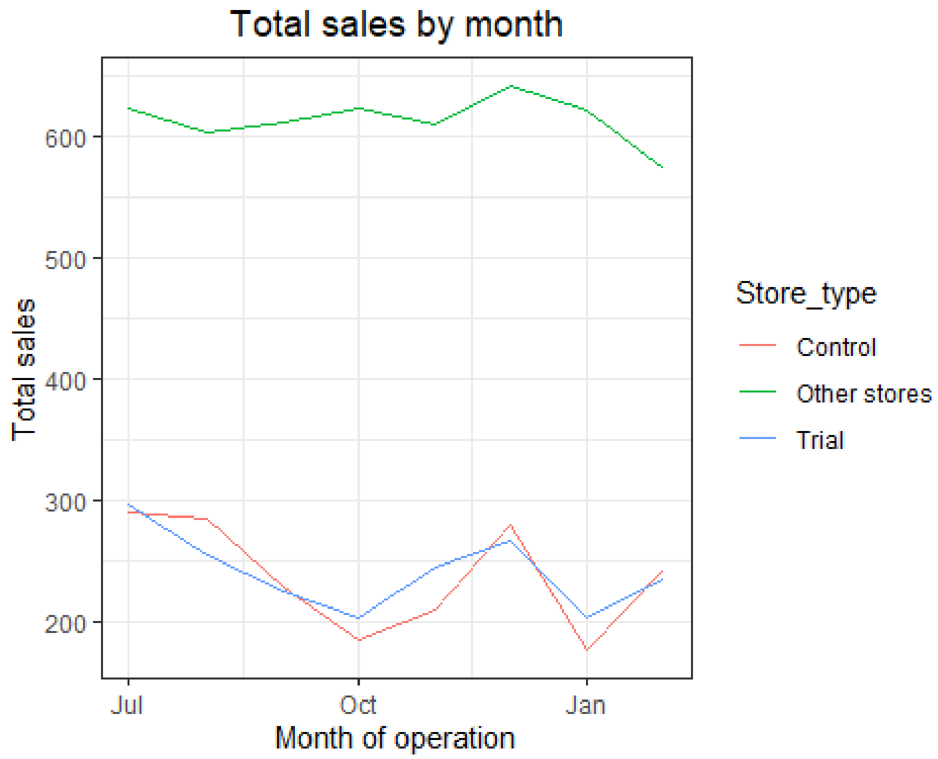
#combine scores

```
# Combine scores across the drivers
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1",
"Store2"))[, scoreNSales := corr_measure * corr_weight + mag_measure * (1 -
corr_weight)]
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by =
c("Store1", "Store2"))[, scoreNCust := corr_measure * corr_weight +
mag_measure * (1 - corr_weight)]

# Combine the 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]

# Select the most appropriate control store for trial store 77
control_store <- score_Control[order(-finalControlScore)][2, Store2]  # The
second highest store because the first will be the trial store itself
control_store

## [1] 233
```

#Visual Checks

```
# Visual checks on total sales
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR ==
trial_store, "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]
```

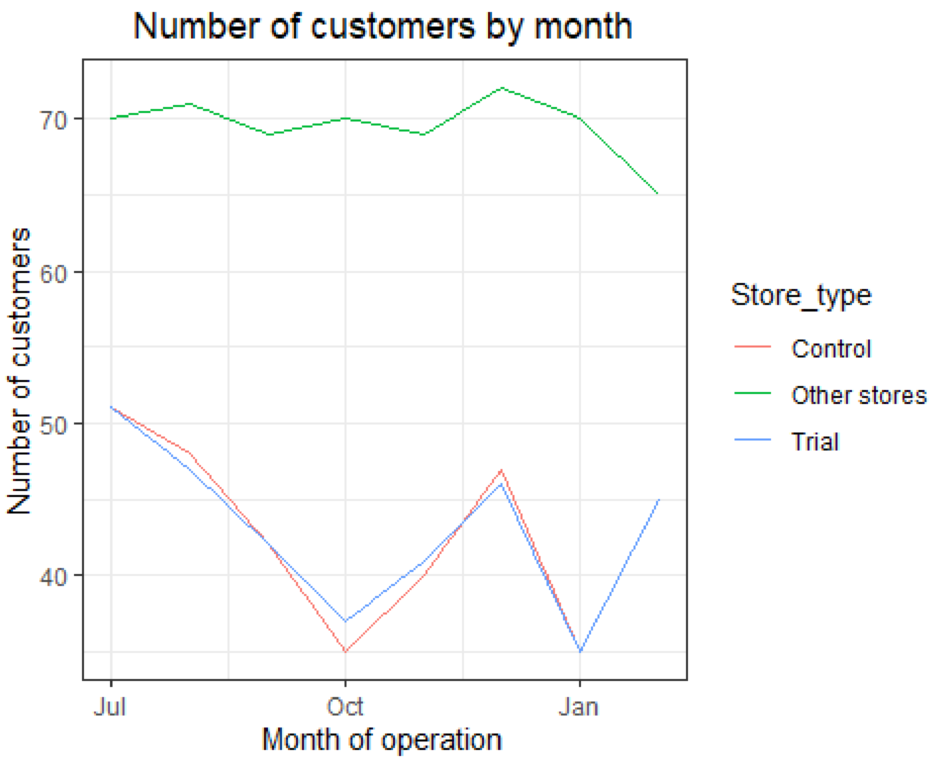
```
ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
  geom_line() +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by
month")
```



```
# Visual checks on number of customers
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR ==
trial_store, "Trial",
  ifelse(STORE_NBR == control_store, "Control", "Other stores"))
][, nCustomers := mean(nCustomers), by = c("YEARMONTH", "Store_type")
][, TransactionMonth := as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1,
sep = "-"), "%Y-%m-%d")
][YEARMONTH < 201903]
```

```
## Warning in `[.data.table`(measureOverTimeCusts[, `:=`(Store_type,
## ifelse(STORE_NBR == : 70.750000 (type 'double') at RHS position 1
## out-of-range(NA) or truncated (precision lost) when assigning to type
'integer'
## (column 4 named 'nCustomers')
```

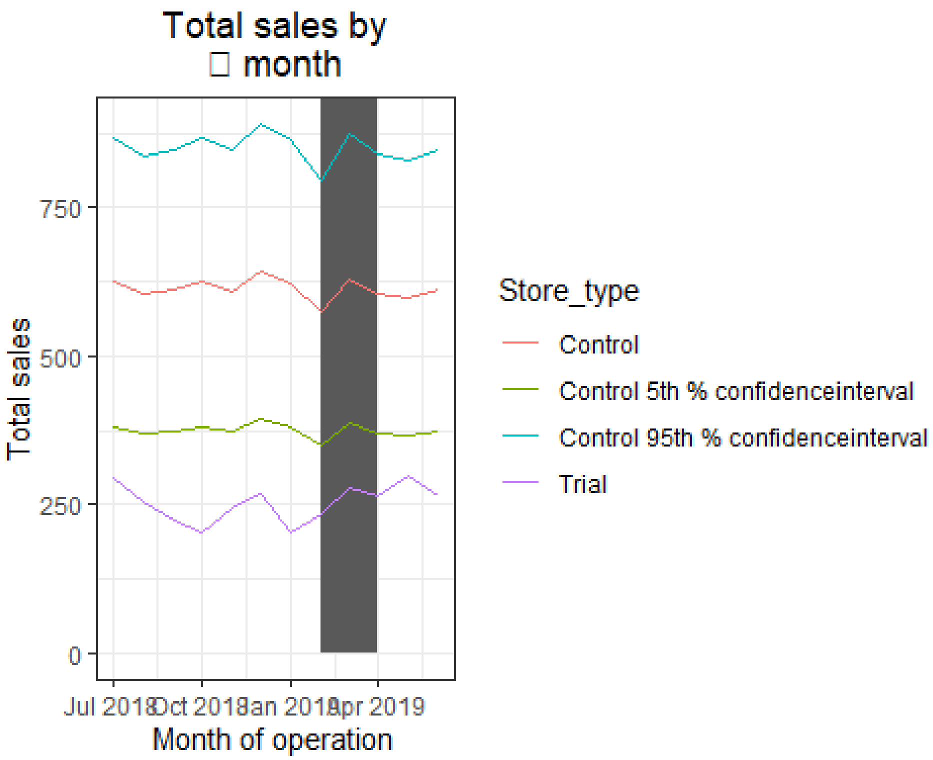
```
ggplot(pastCustomers, aes(TransactionMonth, nCustomers, color = Store_type))
+
  geom_line() +
  labs(x = "Month of operation", y = "Number of customers", title = "Number
of customers by month")
```



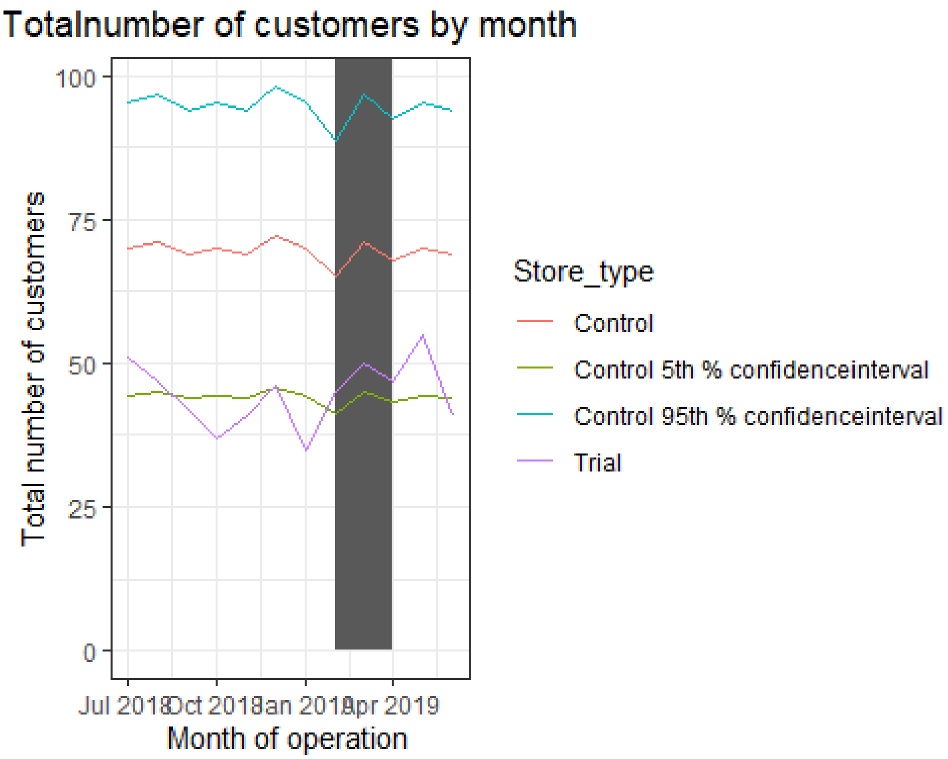
Assess Trial

```
trial_store <- 77
control_store <- 86  # Replace this with the identified control store

# Scale pre-trial control sales to match pre-trial trial store sales
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
                                                 YEARMONTH < 201902,
sum(totSales)] /
                                preTrialMeasures[STORE_NBR == control_store &
                                                 YEARMONTH < 201902,
sum(totSales)]

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

scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store
&YEARMONTH < 201902, sum(totSales)]/preTrialMeasures[STORE_NBR
==control_store & YEARMONTH < 201902, sum(totSales)]
#### Apply the scaling factor
measureOverTimeSales <- measureOverTime
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][
,controlSales := totSales * scalingFactorForControlSales]
```

```
percentageDiff <- merge(scaledControlSales[, c("YEARMONTH",
"controlSales")],measureOverTime[STORE_NBR == trial_store,
c("totSales","YEARMONTH")],
by = "YEARMONTH")[, percentageDiff :=abs(controlSales-totSales)/controlSales]

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

degreesOfFreedom <- 7

#### 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)]

##    TransactionMonth   tValue
##              <Date>    <num>
## 1:       2019-02-01 2.459338
## 2:       2019-03-01 3.051140
## 3:       2019-04-01 2.919055

qt(0.95, df = degreesOfFreedom)

## [1] 1.894579

measureOverTimeSales <- measureOverTime

pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR ==
trial_store, "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_Controls95 <- pastSales[Store_type == "Control",][, totSales :=
totSales * (1 + stdDev * 2)][, Store_type := "Control 95th %
confidenceinterval"]
#### Control store 5th percentile
pastSales_Controls5 <- pastSales[Store_type == "Control",][, totSales :=
totSales * (1 - stdDev * 2)][, Store_type := "Control 5th %
confidenceinterval"]
trialAssessment <- rbind(pastSales, pastSales_Controls95,
pastSales_Controls5)
#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, totSales, color = Store_type))
+
geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0 ,
ymax = Inf, color = NULL), show.legend = FALSE) +
geom_line() +
labs(x = "Month of operation", y = "Total sales", title = "Total sales by
↪ month")
```

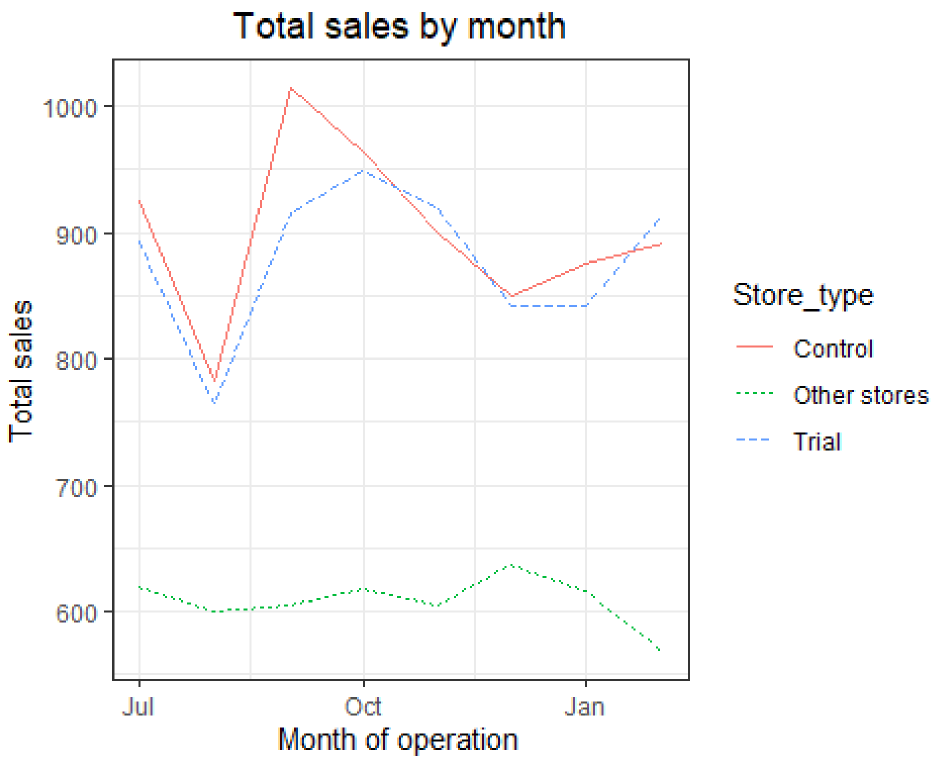


```
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR ==
                                               trial_store &YEARMONTH <
201902, sum(nCustomers)]/preTrialMeasures[STORE_NBR ==control_store &
YEARMONTH < 201902,sum(nCustomers)]
measureOverTimeCusts <- measureOverTime
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store,
][ , controlCustomers := nCustomers *scalingFactorForControlCust
][, Store_type := ifelse(STORE_NBR ==trial_store, "Trial",
ifelse(STORE_NBR == control_store,"Control", "Other stores"))
]
percentageDiff <- merge(scaledControlCustomers[, c("YEARMONTH",
"controlCustomers")],measureOverTimeCusts[STORE_NBR == trial_store,
c("nCustomers", "YEARMONTH")],by = "YEARMONTH")[, percentageDiff :=
abs(controlCustomers-nCustomers)/controlCustomers]

stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7
#### Trial and control store number of customers
pastCustomers <- measureOverTimeCusts[, nCusts := mean(nCustomers), by
=c("YEARMONTH", "Store_type")][Store_type %in% c("Trial", "Control"), ]
#### Control store 95th percentile
pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",][, nCusts
:= nCusts * (1 + stdDev * 2)][, Store_type := "Control 95th %
confidenceinterval"]
#### Control store 5th percentile
pastCustomers_Controls5 <- pastCustomers[Store_type == "Control",][, nCusts
:= nCusts * (1 - stdDev * 2)][, Store_type := "Control 5th %
confidenceinterval"]
trialAssessment <- rbind(pastCustomers, pastCustomers_Controls95,
pastCustomers_Controls5)
#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, nCusts, color = Store_type)) +
geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0 ,
ymax = Inf, color = NULL), show.legend = FALSE) +

geom_line() +
labs(x = "Month of operation", y = "Total number of customers", title =
"Totalnumber of customers by month")
```

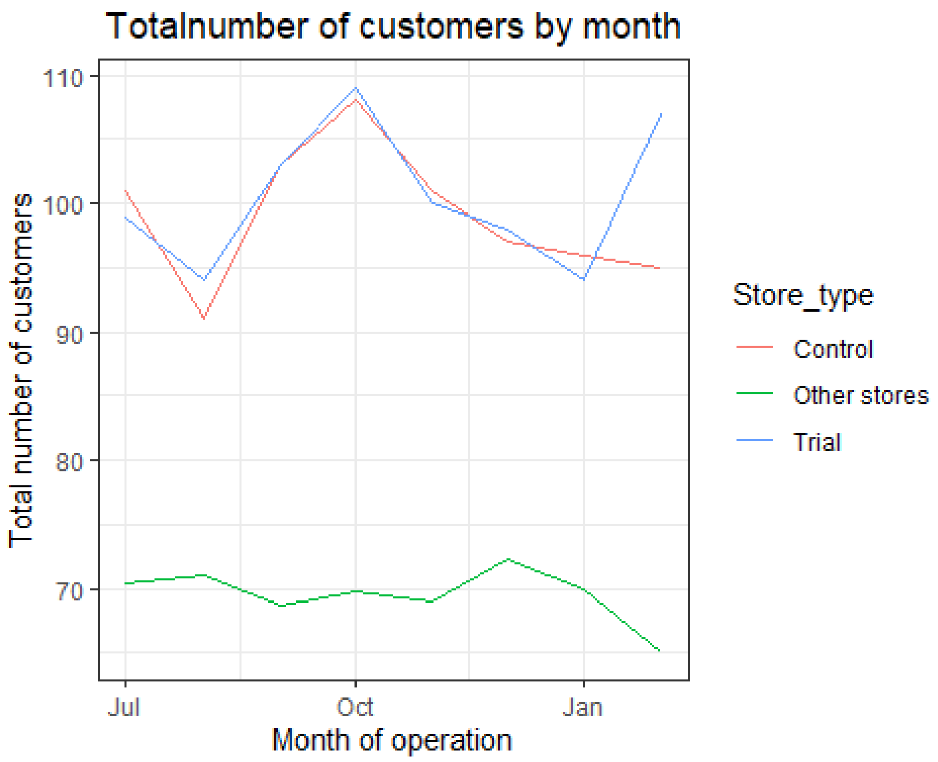


```
measureOverTime <- data[, .(totSales = sum(TOT_SALES),nCustomers =
uniqueN(LYLTY_CARD_NBR),nTxnPerCust =uniqueN(TXN_ID)/uniqueN(LYLTY_CARD_NBR),
             nChipsPerTxn = sum(PROD_QTY)/uniqueN(TXN_ID),
                                   avgPricePerUnit =
sum(TOT_SALES)/sum(PROD_QTY)), by = c("STORE_NBR",
"YEARMONTH")][order(STORE_NBR, YEARMONTH)]
#### Use the functions for calculating correlation
trial_store <- 86
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales),
trial_store)
```

```
corr_nCustomers <- calculateCorrelation(preTrialMeasures, quote(nCustomers),
trial_store)
#### Use the functions for calculating magnitude
magnitude_nSales <-
calculateMagnitudeDistance(preTrialMeasures,quote(totSales), trial_store)
magnitude_nCustomers <-
calculateMagnitudeDistance(preTrialMeasures,quote(nCustomers), trial_store)
#### Create a combined score composed of correlation and magnitude
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by =
c("Store1","Store2"))[, scoreNSales := corr_measure * corr_weight +
mag_measure * (1-corr_weight)]

score_nCustomers <- 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_Control <- merge(score_nSales, score_nCustomers, by =
c("Store1","Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]

#### Select control store for trial store 86
control_store <- score_Control[Store1 == trial_store,][order(-
finalControlScore)][2, Store2]
control_store

## [1] 155

measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR
==trial_store, "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 , ]
ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
geom_line(aes(linetype = Store_type)) +
labs(x = "Month of operation", y = "Total sales", title = "Total sales by
month")
```

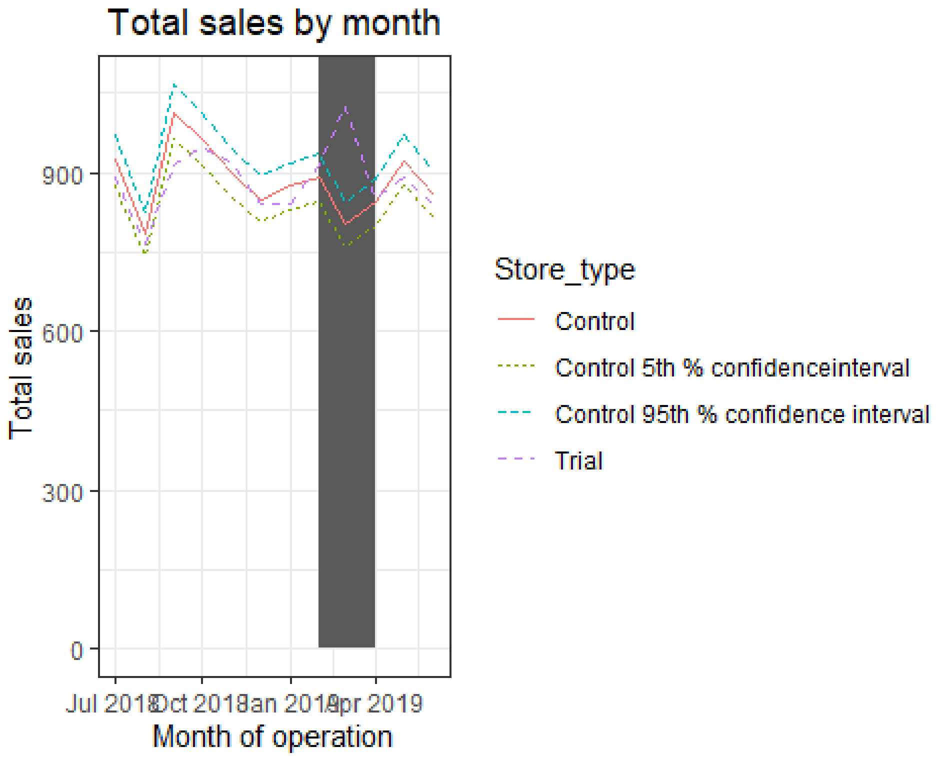


```
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR
==trial_store, "Trial",
              ifelse(STORE_NBR == control_store, "Control", "Other
stores"))][, numberCustomers := mean(nCustomers), by = c("YEARMONTH",
"Store_type")
][, TransactionMonth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

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



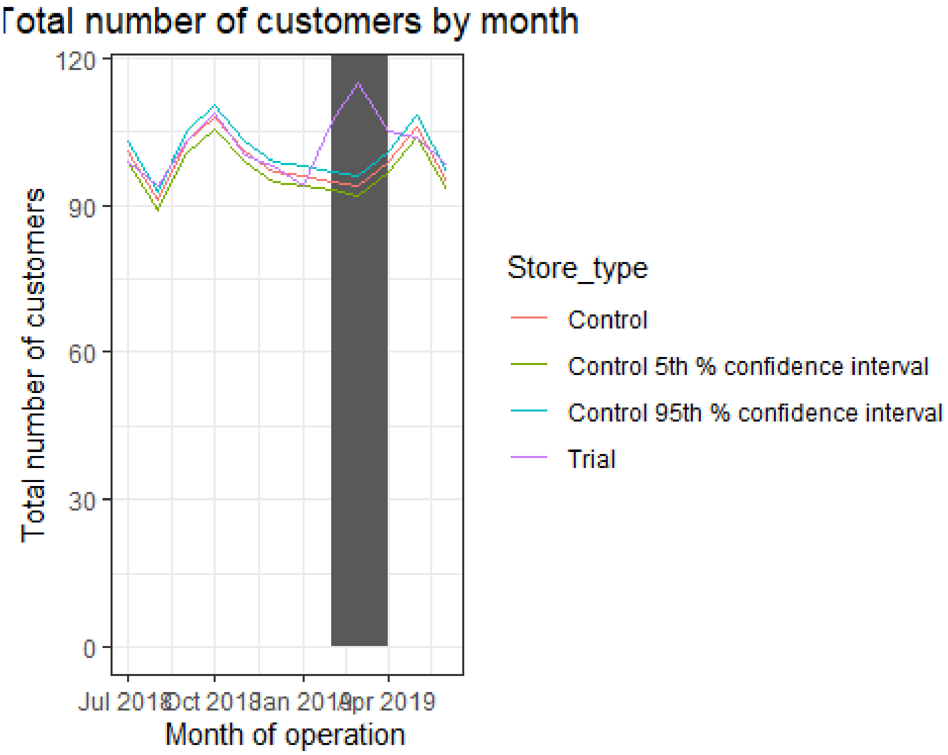
```
#### Scale pre-trial control sales to match pre-trial trial store sales
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
                            YEARMONTH < 201902,
sum(totSales)]/preTrialMeasures[STORE_NBR == control_store & YEARMONTH <
201902, sum(totSales)]

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

percentageDiff <- merge(scaledControlSales[, c("YEARMONTH",
"controlSales")],measureOverTime[STORE_NBR == trial_store,
c("totSales","YEARMONTH")],
                        by = "YEARMONTH")[,percentageDiff :=abs(controlSales-
totSales)/controlSales]

stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7
#### Trial and control store total sales
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR ==
trial_store, "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_Controls95 <- pastSales[Store_type == "Control",][, totSales :=
totSales * (1 + stdDev * 2)
                             ][, Store_type := "Control 95th % confidence
interval"]
#### Control store 5th percentile
pastSales_Controls5 <- pastSales[Store_type == "Control",][, totSales :=
totSales * (1 - stdDev * 2)
                                                         ][, Store_type :=
"Control 5th % confidenceinterval"]
trialAssessment <- rbind(pastSales, pastSales_Controls95,
pastSales_Controls5)
#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, totSales, color = Store_type))
+
geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0
,ymax = Inf, color = NULL), show.legend = FALSE) +

geom_line(aes(linetype = Store_type)) +
labs(x = "Month of operation", y = "Total sales", title = "Total sales by
month")
```

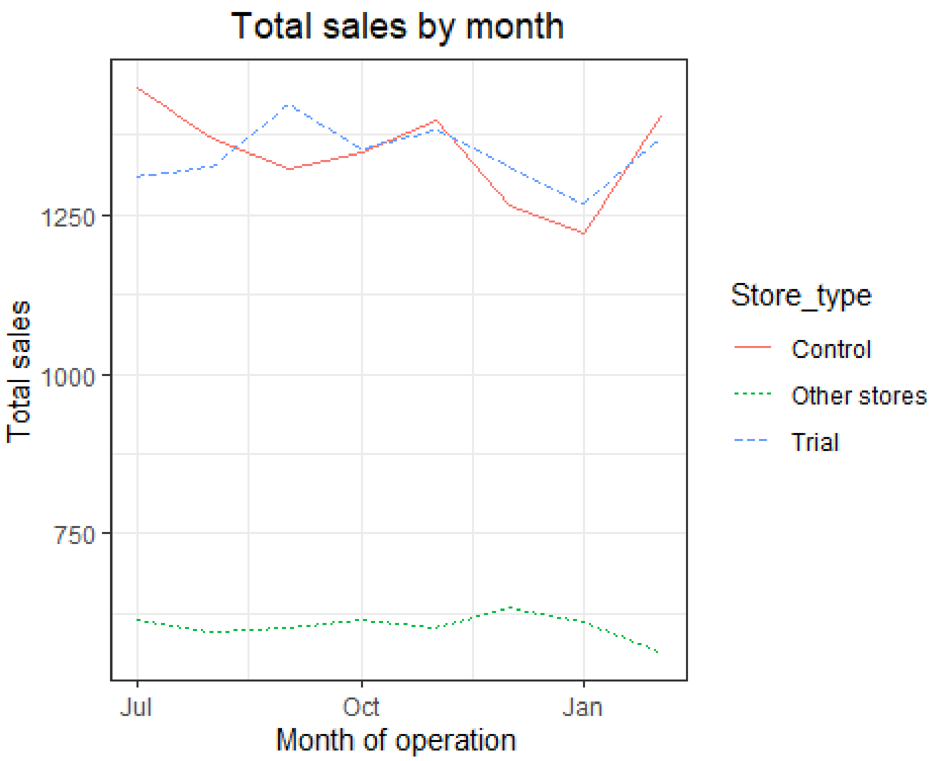


```
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(nCustomers)]/preTrialMeasures[STORE_NBR ==
                                                     control_store & YEARMONTH
< 201902, sum(nCustomers)]
```

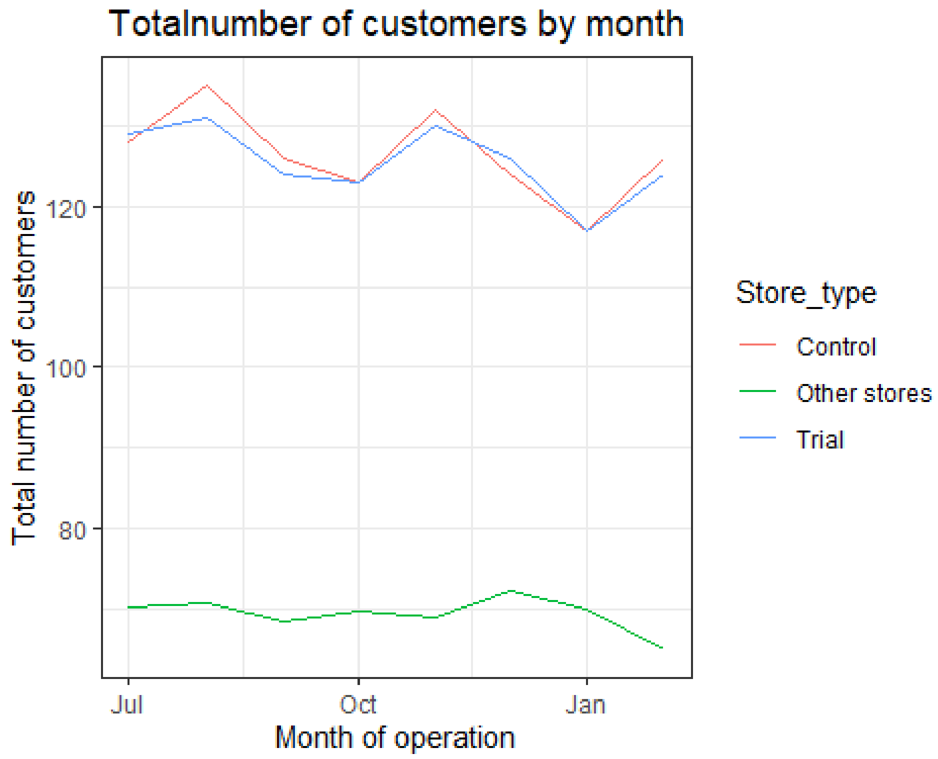
```
#### Apply the scaling factor
measureOverTimeCusts <- measureOverTime
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store,][
, controlCustomers := nCustomers* scalingFactorForControlCust
                               ][, Store_type := ifelse(STORE_NBR==
trial_store, "Trial",ifelse(STORE_NBR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

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


stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7
#### Trial and control store number of customers
pastCustomers <- measureOverTimeCusts[, nCusts := mean(nCustomers), by
=c("YEARMONTH", "Store_type")
                                            ][Store_type %in% c("Trial",
"Control"), ]
#### Control store 95th percentile
pastCustomers_Controls95 <- pastCustomers[Store_type == "Control",][, nCusts
:= nCusts * (1 + stdDev * 2)
                                           ][, Store_type := "Control 95th %
confidence interval"]
#### Control store 5th percentile
pastCustomers_Controls5 <- pastCustomers[Store_type == "Control",][, nCusts
:= nCusts * (1 - stdDev * 2)
                                            ][, Store_type := "Control 5th %
confidence interval"]

trialAssessment <- rbind(pastCustomers,
pastCustomers_Controls95,pastCustomers_Controls5)
#### Plotting these in one nice graph

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

geom_line() +
labs(x = "Month of operation", y = "Total number of customers", title =
"Total number of customers by month")
```



```
measureOverTime <- data[, .(totSales = sum(TOT_SALES),
nCustomers = uniqueN(LYLTY_CARD_NBR),
nTxnPerCust =uniqueN(TXN_ID)/uniqueN(LYLTY_CARD_NBR),nChipsPerTxn =
sum(PROD_QTY)/uniqueN(TXN_ID),
avgPricePerUnit = sum(TOT_SALES)/sum(PROD_QTY)), by = c("STORE_NBR",
"YEARMONTH")][order(STORE_NBR,YEARMONTH)]
#### Use the functions for calculating correlation
trial_store <- 88
corr_nSales <- calculateCorrelation(preTrialMeasures,
quote(totSales),trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures,
quote(nCustomers),trial_store)
#### Use the functions for calculating magnitude
magnitude_nSales <-
calculateMagnitudeDistance(preTrialMeasures,quote(totSales), trial_store)
magnitude_nCustomers <-
calculateMagnitudeDistance(preTrialMeasures,quote(nCustomers), trial_store)
#### Create a combined score composed of correlation and magnitude
corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by =
c("Store1","Store2"))[, scoreNSales := corr_measure * corr_weight +
mag_measure * (1-corr_weight)]


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

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

## [1] 237

measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR
==trial_store, "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 , ]
ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
geom_line(aes(linetype = Store_type)) +
labs(x = "Month of operation", y = "Total sales", title = "Total sales by
month")
```

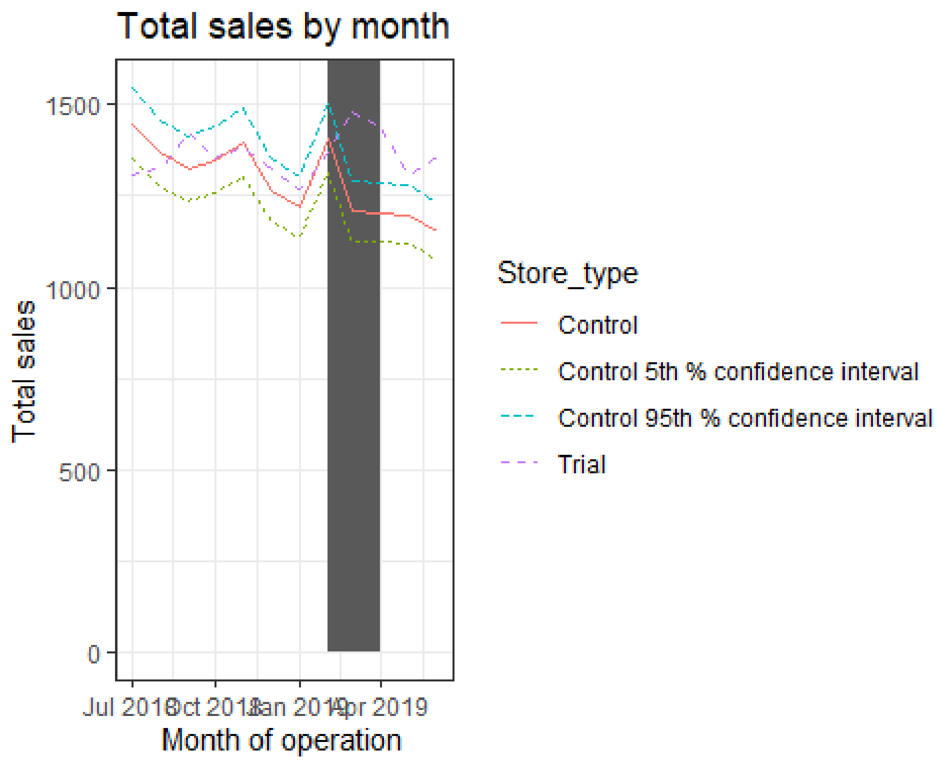


```r
measureOverTimeCusts <- measureOverTime
pastCustomers <- measureOverTimeCusts[, Store_type := ifelse(STORE_NBR
==trial_store, "Trial",
              ifelse(STORE_NBR == control_store,"Control", "Other
stores"))][, numberCustomers := mean(nCustomers), by =c("YEARMONTH",
"Store_type")
           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

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



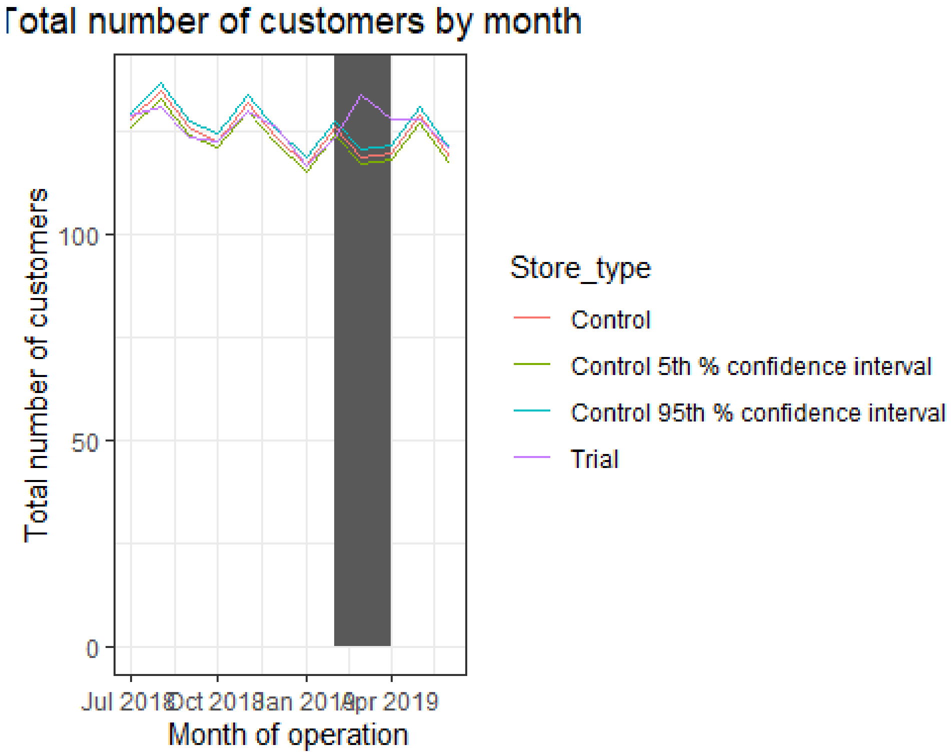
```r
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(totSales)]/preTrialMeasures[STORE_NBR ==
control_store & YEARMONTH < 201902, sum(totSales)]

#### Apply the scaling factor
measureOverTimeSales <- measureOverTime
scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][
,controlSales := totSales * scalingFactorForControlSales]
#### Calculate the percentage difference between scaled control sales and
trial
```

```
percentageDiff <- merge(scaledControlSales[, c("YEARMONTH", "controlSales")],
      measureOverTime[STORE_NBR == trial_store, c("totSales",
"YEARMONTH")],by = "YEARMONTH")[, percentageDiff :=abs(controlSales-
totSales)/controlSales]
#### As our null hypothesis is that the trial period is the same as the

stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7
#### Trial and control store total sales
measureOverTimeSales <- measureOverTime
pastSales <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR ==
trial_store, "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_Controls95 <- pastSales[Store_type == "Control",][, totSales :=
totSales * (1 + stdDev * 2)
                        ][, Store_type := "Control 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_Controls95,
pastSales_Controls5)
#### Plotting these in one nice graph

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



```
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(nCustomers)]/preTrialMeasures[STORE_NBR ==
control_store & YEARMONTH < 201902, sum(nCustomers)]

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

#### Calculate the percentage difference between scaled control sales and
trial

percentageDiff <- merge(scaledControlCustomers[,
c("YEARMONTH","controlCustomers")],

measureOverTime[STORE_NBR == trial_store, c("nCustomers", "YEARMONTH")],by =
"YEARMONTH")[, percentageDiff :=abs(controlCustomers-
nCustomers)/controlCustomers]
#### As our null hypothesis is that the trial period is the same as the

stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])
degreesOfFreedom <- 7 # note that there are 8 months in the pre-trial period
```

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

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

trialAssessment <- rbind(pastCustomers,
pastCustomers_Controls95,pastCustomers_Controls5)
#### Plotting these in one nice graph
ggplot(trialAssessment, aes(TransactionMonth, nCusts, color = Store_type)) +
geom_rect(data = trialAssessment[ YEARMONTH < 201905 & YEARMONTH > 201901 ,],
aes(xmin = min(TransactionMonth), xmax = max(TransactionMonth), ymin = 0 ,
ymax = Inf, color = NULL), show.legend = FALSE) +

geom_line() +
labs(x = "Month of operation", y = "Total number of customers", title =
"Total number of customers by month")
```


Total number of customers by month
Total number of customers
100
50
0
Jul 2018
Oct 2018
Jan 2019
Apr 2019
Month of operation
Store_type
Control
Control 5th % confidence interval
Control 95th % confidence interval
Trial