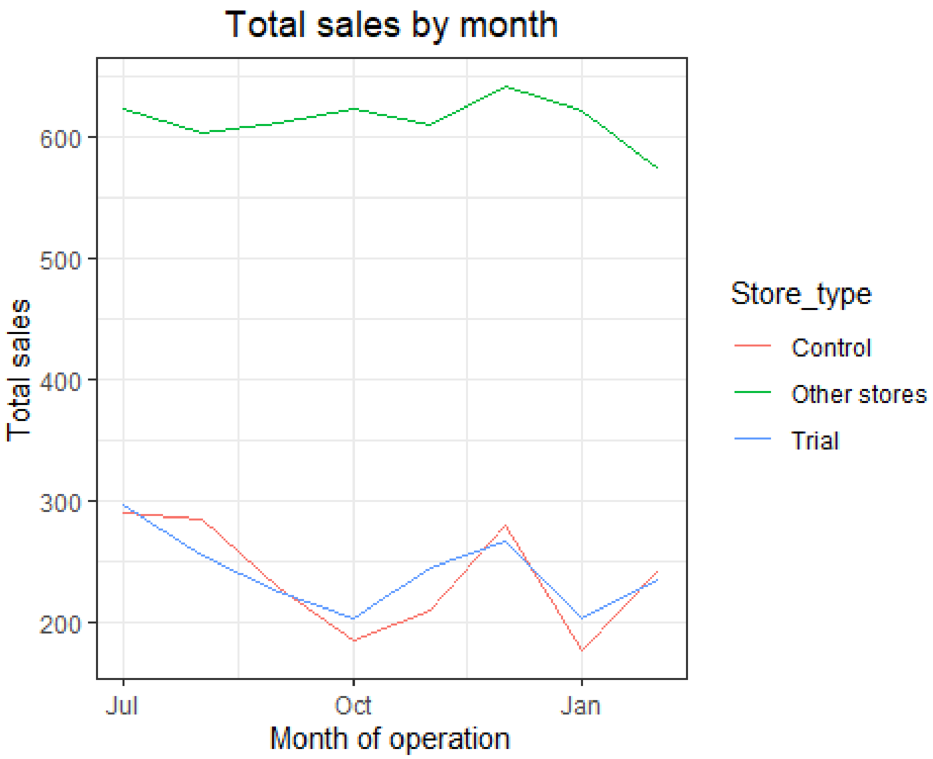
Task 2

```
library(data.table)

## Warning: package 'data.table' was built under R version 3.6.3

library(ggplot2)
library(tidyr)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

file_path = "C:/Users/jerem/Documents/R/Quantium/Task 2/"
data = fread(paste0(file_path, "QVI_data.csv"))
theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = 0.5))


data[, YEARMONTH := format(as.Date(data$DATE, "%Y%m"), "%Y%m") ]

##getting data for trial period


measure_over_time = data %>% group_by(STORE_NBR,YEARMONTH) %>%
  summarise(totSales = sum(TOT_SALES),
            nCustomers = uniqueN(LYLTY_CARD_NBR),
            nTxnPerCust = uniqueN(TXN_ID)/nCustomers,
            avgPricePerUnit = mean(TOT_SALES/PROD_QTY))
measure_over_time = data.table(measure_over_time)

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

```
## function for calculating correlation
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,
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 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)
}

### TRIAL STORE 77
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)


corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by =
intersect(names(corr_nSales),names(magnitude_nSales)))[, scoreNSales :=
corr_weight*corr_nSales$corr_measure + (1-corr_weight)*
magnitude_nSales$mag_measure]
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by =
intersect(names(corr_nCustomers),names(magnitude_nCustomers)))[, scoreNCust
:= corr_weight*corr_measure + (1-corr_weight)*mag_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[order(-finalControlScore)]
control_store = score_Control[2,]$Store2

measureOverTimeSales = measure_over_time
measureOverTimeSales$YEARMONTH = as.numeric(measureOverTimeSales$YEARMONTH)
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 = "-"))
][YEARMONTH < 201903 , ]


ggplot(pastSales, aes(TransactionMonth, totSales, color = Store_type)) +
```

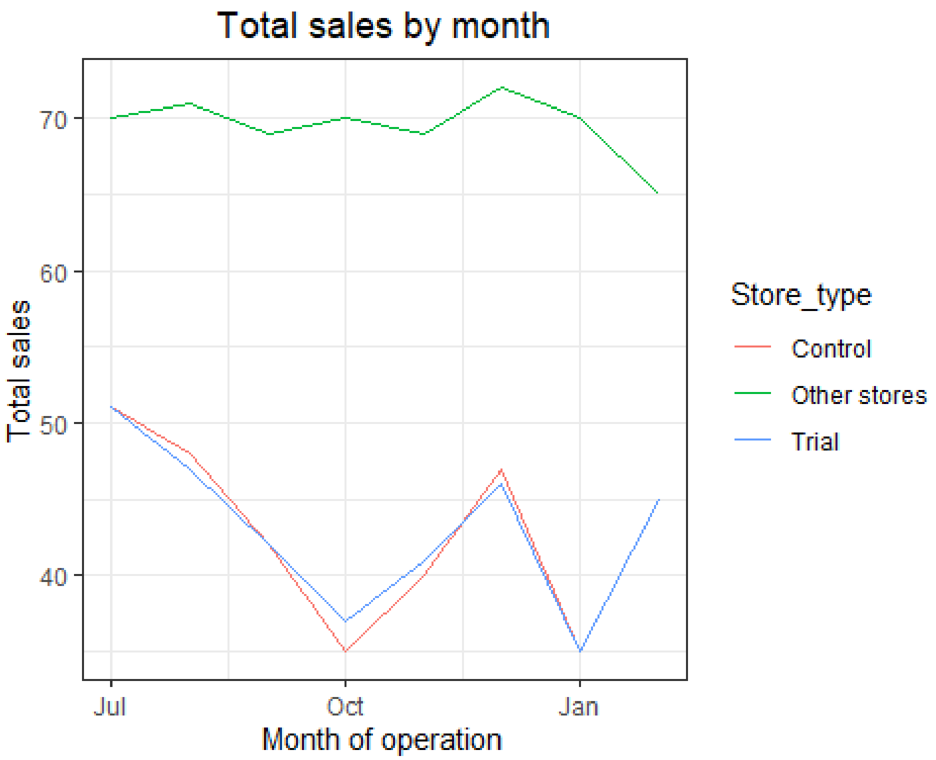
```r
geom_line() +
labs(x = "Month of operation", y = "Total sales", title = "Total sales by
month")
```



```r
pastCustomers <- measureOverTimeSales[, 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 = "-"))
                                      ][YEARMONTH < 201903 , ]
```

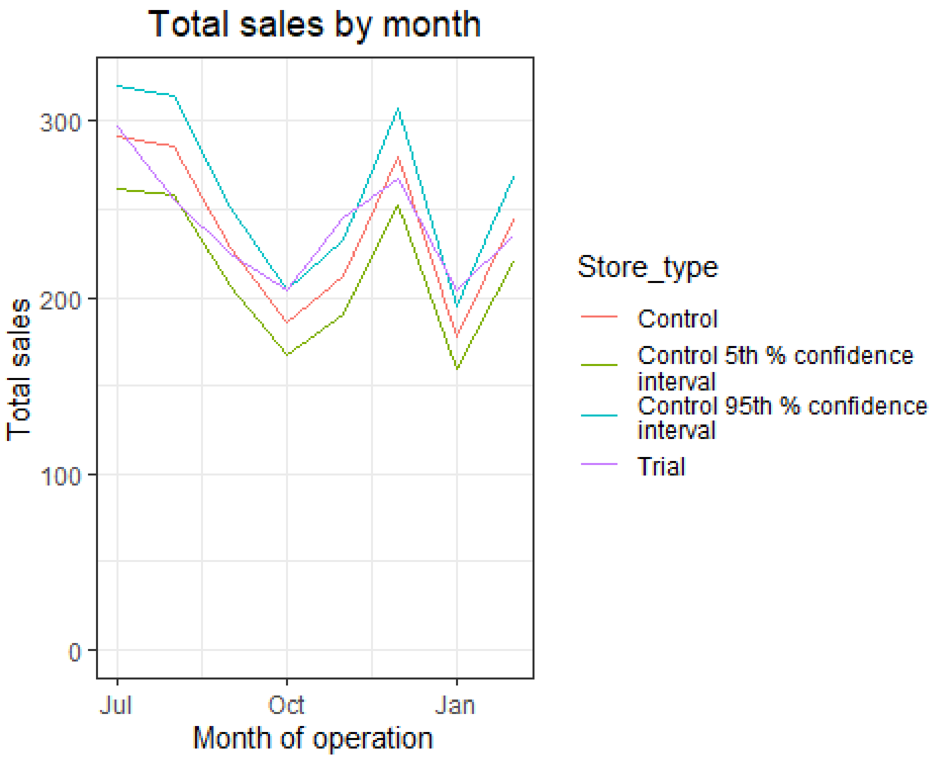
```
## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 1 column 'nCustomers': 70.750000 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 2 column 'nCustomers': 71.352490 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 3 column 'nCustomers': 69.110687 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 4 column 'nCustomers': 70.334601 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 5 column 'nCustomers': 69.534351 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 6 column 'nCustomers': 72.731801 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 7 column 'nCustomers': 70.471264 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 8 column 'nCustomers': 65.492366 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 9 column 'nCustomers': 71.509506 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 10 column 'nCustomers': 68.771863 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 11 column 'nCustomers': 70.865900 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 12 column 'nCustomers': 69.396947 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'
```

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

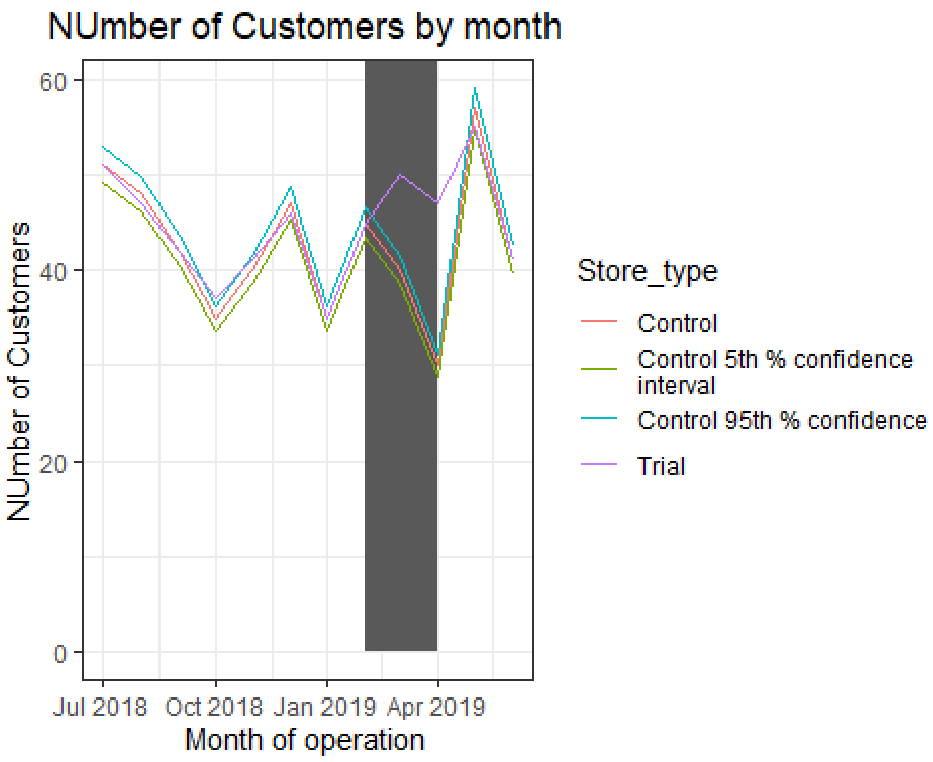


```
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(totSales)]/preTrialMeasures[STORE_NBR ==
control_store &
YEARMONTH < 201902, sum(totSales)]


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

percentageDiff <- merge(scaledControlSales,pastSales[Store_type == "Trial"],
by = "YEARMONTH")[,percentageDiff := abs(controlSales -
totSales.y)/controlSales ]
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])

degreesOfFreedom <- 7
percentageDiff[, tValue := abs(totSales.x - mean(totSales.y))/stdDev
][, TransactionMonth := TransactionMonth.x
]
pastSales <- pastSales[Store_type %in% c("Trial", "Control"), ]
```

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

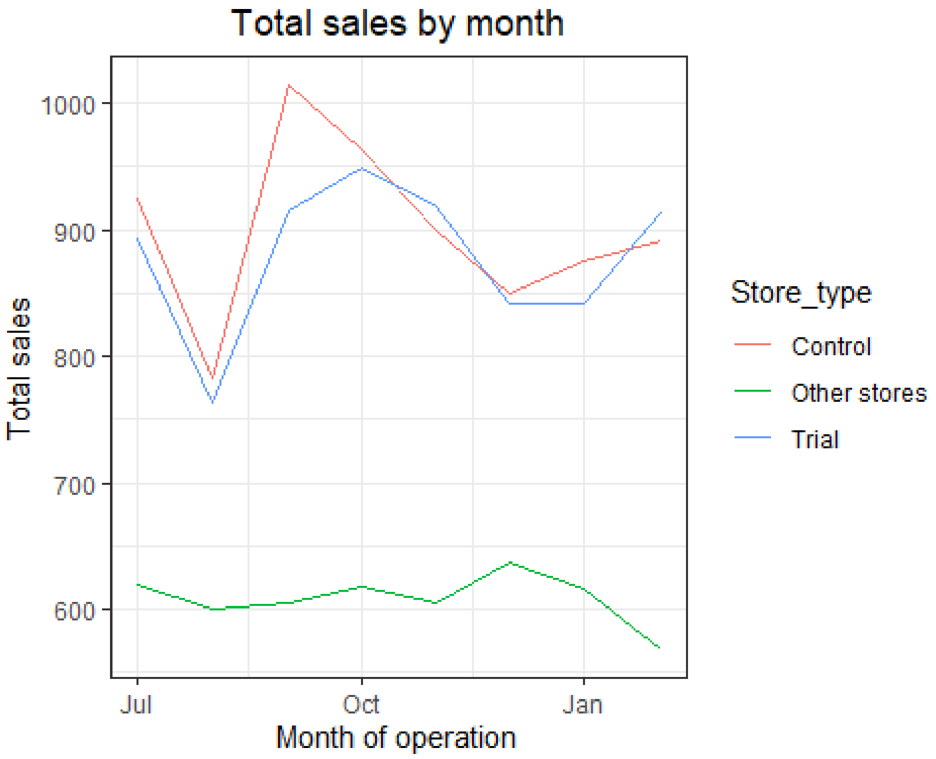


```
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(nCustomers)]/preTrialMeasures[STORE_NBR ==
control_store & YEARMONTH < 201902, sum(nCustomers)]
measureOverTimeCusts <- measureOverTimeSales
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store, ][
, controlCustomer := nCustomers * scalingFactorForControlCust]

percentageDiff <- merge(scaledControlCustomers,pastCustomers[Store_type ==
"Trial"],
                        by = "YEARMONTH")[,percentageDiff :=
abs(controlCustomer - nCustomers.y)/controlCustomer ]
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"), ]


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

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 = "NUmber of Customers", title = "NUmber
of Customers by month")
```



```
### TRIAL STORE 86
trial_store <- 86
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)


corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by =
intersect(names(corr_nSales),names(magnitude_nSales)))[, scoreNSales :=
corr_weight*corr_nSales$corr_measure + (1-corr_weight)*
magnitude_nSales$mag_measure]
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by =
intersect(names(corr_nCustomers),names(magnitude_nCustomers)))[, scoreNCust
:= corr_weight*corr_measure + (1-corr_weight)*mag_measure]
score_Control <- merge(score_nSales,score_nCustomers , by = c("Store1",
"Store2"))
score_Control[, finalControlScore := scoreNSales * 0.5 + scoreNCust * 0.5]
```

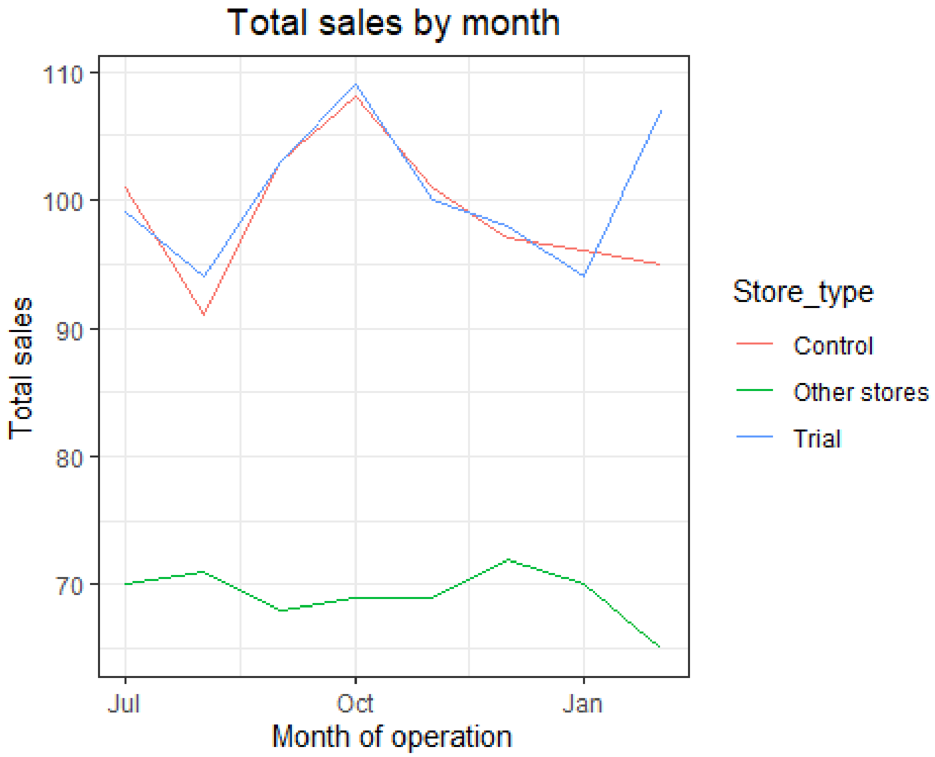
```
score_Control = score_Control[order(-finalControlScore)]
control_store = score_Control[2,]$Store2

measureOverTimeSales = measure_over_time
measureOverTimeSales$YEARMONTH = as.numeric(measureOverTimeSales$YEARMONTH)
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 = "-"))
                                      ][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")
```



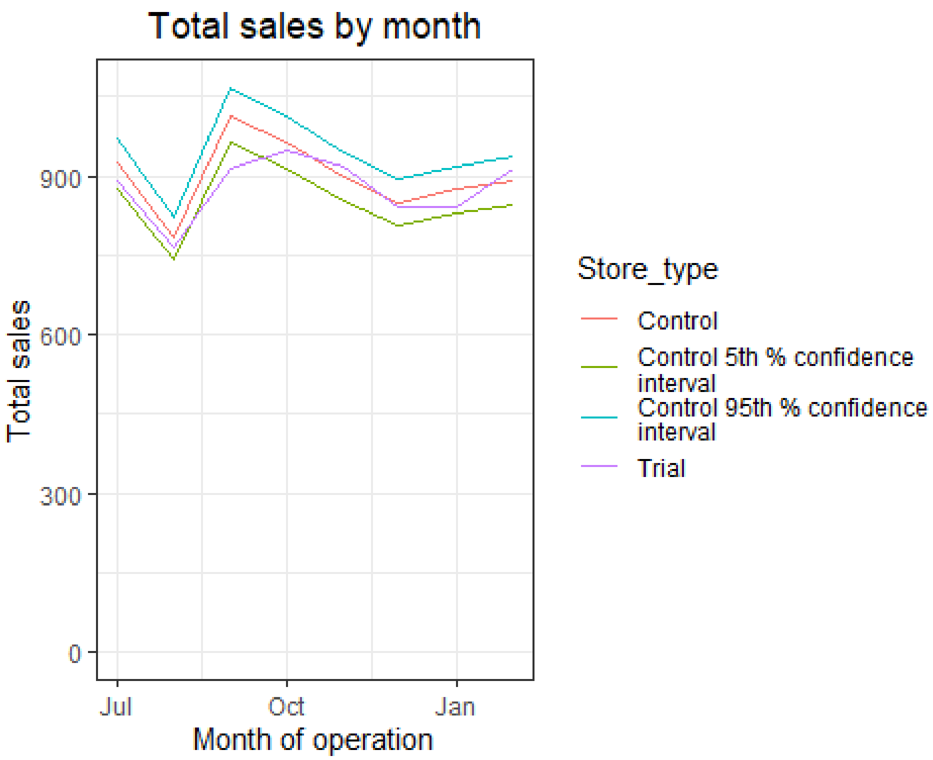
```r
pastCustomers <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR ==
trial_store,
                                                             "Trial",
                                                             ifelse(STORE_NBR
== control_store,

"Control", "Other stores"))
                                      ][, nCustomers := mean(nCustomers), by
= c("YEARMONTH",

"Store_type")
                                        ][, TransactionMonth :=
as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"))
                                          ][YEARMONTH < 201903 , ]
```

```
## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 1 column 'nCustomers': 70.378788 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 2 column 'nCustomers': 71.007663 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 3 column 'nCustomers': 68.645038 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 4 column 'nCustomers': 69.783270 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 5 column 'nCustomers': 69.076336 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 6 column 'nCustomers': 72.340996 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 7 column 'nCustomers': 70.011494 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 8 column 'nCustomers': 65.064885 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 9 column 'nCustomers': 71.057034 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 10 column 'nCustomers': 68.288973 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 11 column 'nCustomers': 70.490421 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 12 column 'nCustomers': 68.973282 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'
```

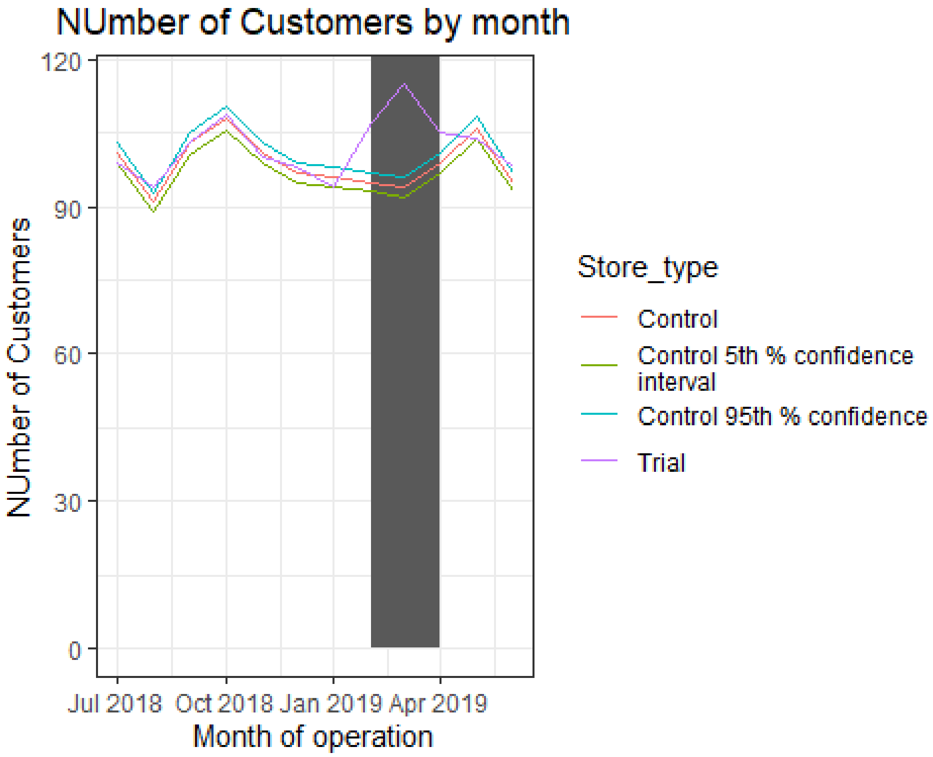
```
ggplot(pastCustomers, aes(TransactionMonth, nCustomers, color = Store_type))
+
  geom_line() +
  labs(x = "Month of operation", y = "Total sales", title = "Total sales by
month")
```



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

YEARMONTH < 201902, sum(totSales)]


scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][ ,

controlSales := totSales * scalingFactorForControlSales]

percentageDiff <- merge(scaledControlSales,pastSales[Store_type == "Trial"],
by = "YEARMONTH")[,percentageDiff := abs(controlSales -
totSales.y)/controlSales ]
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])

degreesOfFreedom <- 7
percentageDiff[, tValue := abs(totSales.x - mean(totSales.y))/stdDev
               ][, TransactionMonth := TransactionMonth.x
                 ]
pastSales <- pastSales[Store_type %in% c("Trial", "Control"), ]


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

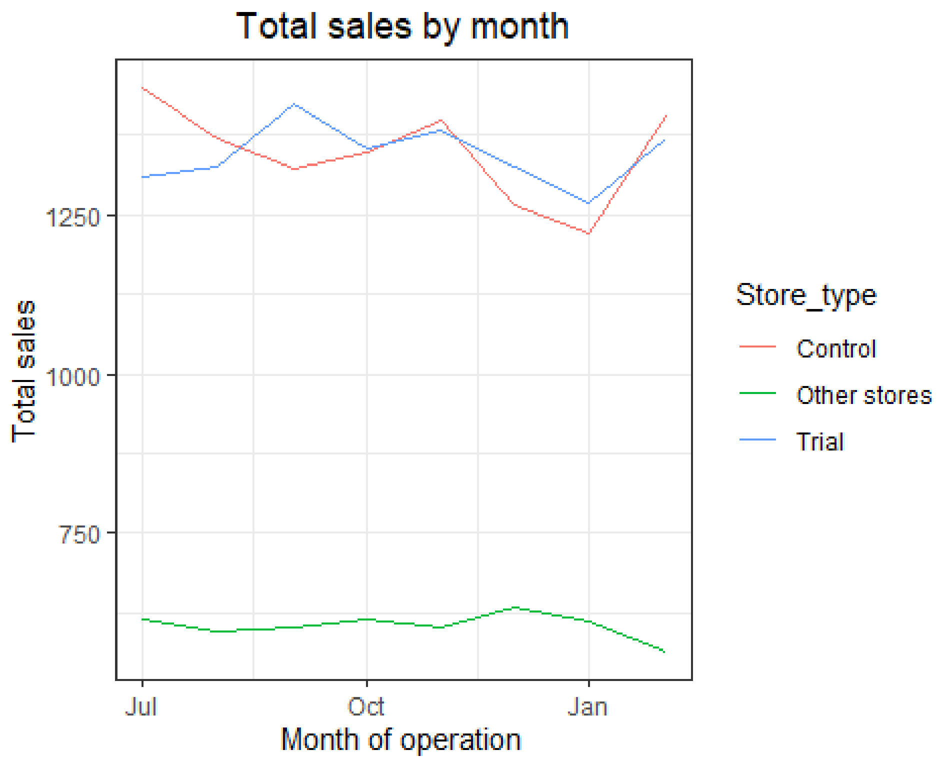


```
scalingFactorForControlCust <- preTrialMeasures[STORE_NBR == trial_store &
YEARMONTH < 201902, sum(nCustomers)]/preTrialMeasures[STORE_NBR ==
control_store & YEARMONTH < 201902, sum(nCustomers)]
measureOverTimeCusts <- measureOverTimeSales
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store, ][
, controlCustomer := nCustomers * scalingFactorForControlCust]

percentageDiff <- merge(scaledControlCustomers,pastCustomers[Store_type ==
"Trial"],
                        by = "YEARMONTH")[,percentageDiff :=
abs(controlCustomer - nCustomers.y)/controlCustomer ]
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"), ]


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

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 = "NUmber of Customers", title = "NUmber
of Customers by month")
```



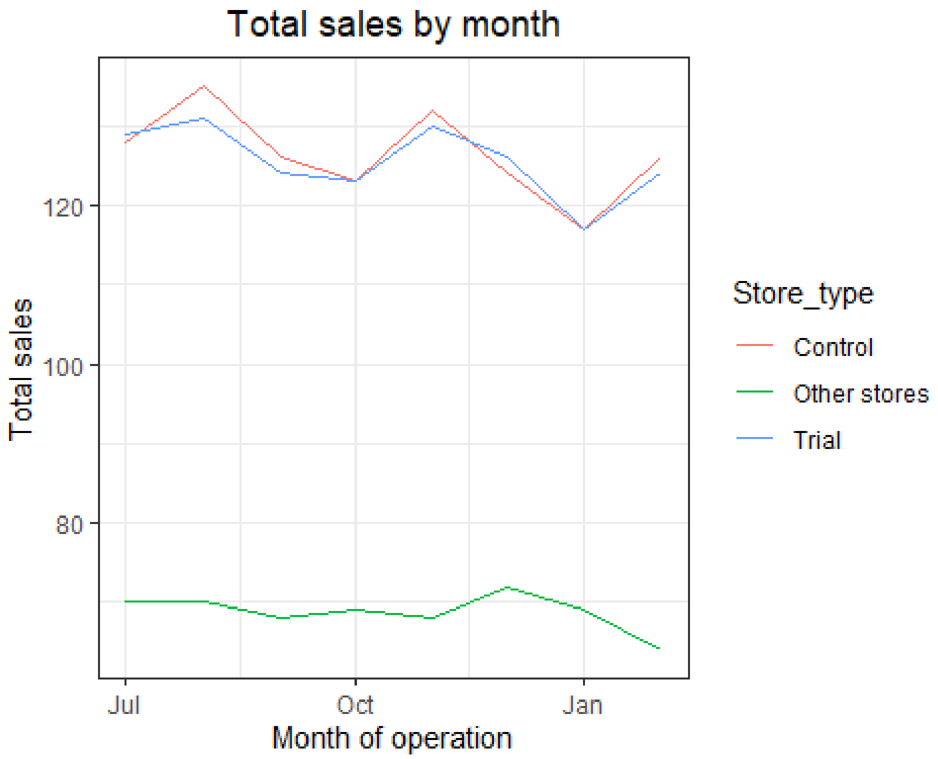
```
### TRIAL STORE 88
trial_store <- 88
corr_nSales <- calculateCorrelation(preTrialMeasures, quote(totSales),
                                    trial_store)
corr_nCustomers <- calculateCorrelation(preTrialMeasures,
quote(nCustomers),trial_store )
```

```
magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures,
quote(totSales),
                                               trial_store)
magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures,
                                                   quote(nCustomers),
trial_store)


corr_weight <- 0.5
score_nSales <- merge(corr_nSales, magnitude_nSales, by =
intersect(names(corr_nSales),names(magnitude_nSales)))[, scoreNSales :=
corr_weight*corr_nSales$corr_measure + (1-corr_weight)*
magnitude_nSales$mag_measure]
score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by =
intersect(names(corr_nCustomers),names(magnitude_nCustomers)))[, scoreNCust
:= corr_weight*corr_measure + (1-corr_weight)*mag_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[order(-finalControlScore)]
control_store = score_Control[2,]$Store2

measureOverTimeSales = measure_over_time
measureOverTimeSales$YEARMONTH = as.numeric(measureOverTimeSales$YEARMONTH)
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 = "-"))
                                      ][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")
```



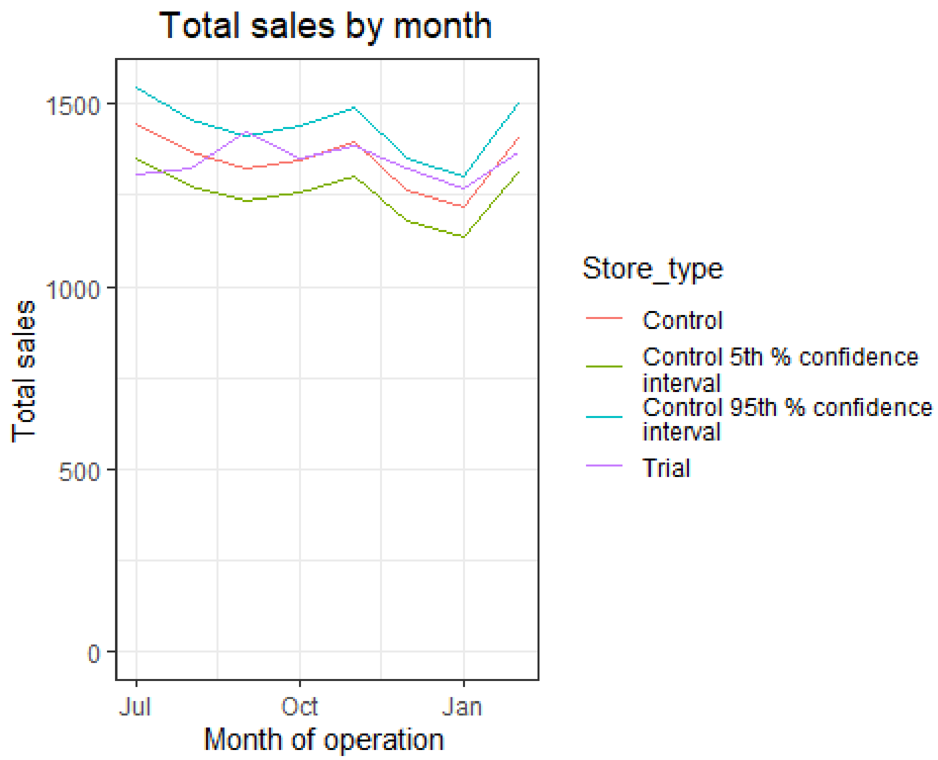
```
pastCustomers <- measureOverTimeSales[, Store_type := ifelse(STORE_NBR ==
trial_store,
                                                             "Trial",
                                                             ifelse(STORE_NBR
== control_store,

"Control", "Other stores"))
                                      ][, nCustomers := mean(nCustomers), by
= c("YEARMONTH",

"Store_type")
                                        ][, TransactionMonth :=
as.Date(paste(YEARMONTH %/% 100, YEARMONTH %% 100, 1, sep = "-"))
                                          ][YEARMONTH < 201903 , ]

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 1 column 'nCustomers': 70.162879 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 2 column 'nCustomers': 70.697318 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 3 column 'nCustomers': 68.477099 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 4 column 'nCustomers': 69.673004 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 5 column 'nCustomers': 68.843511 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 6 column 'nCustomers': 72.130268 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 7 column 'nCustomers': 69.842912 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 8 column 'nCustomers': 64.881679 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 9 column 'nCustomers': 70.889734 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 10 column 'nCustomers': 68.121673 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 11 column 'nCustomers': 70.310345 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'

## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store_type,
## ifelse(STORE_NBR == : Group 12 column 'nCustomers': 68.793893 (type
## 'double') at RHS position 1 truncated (precision lost) when assigning to
## type 'integer'
```

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

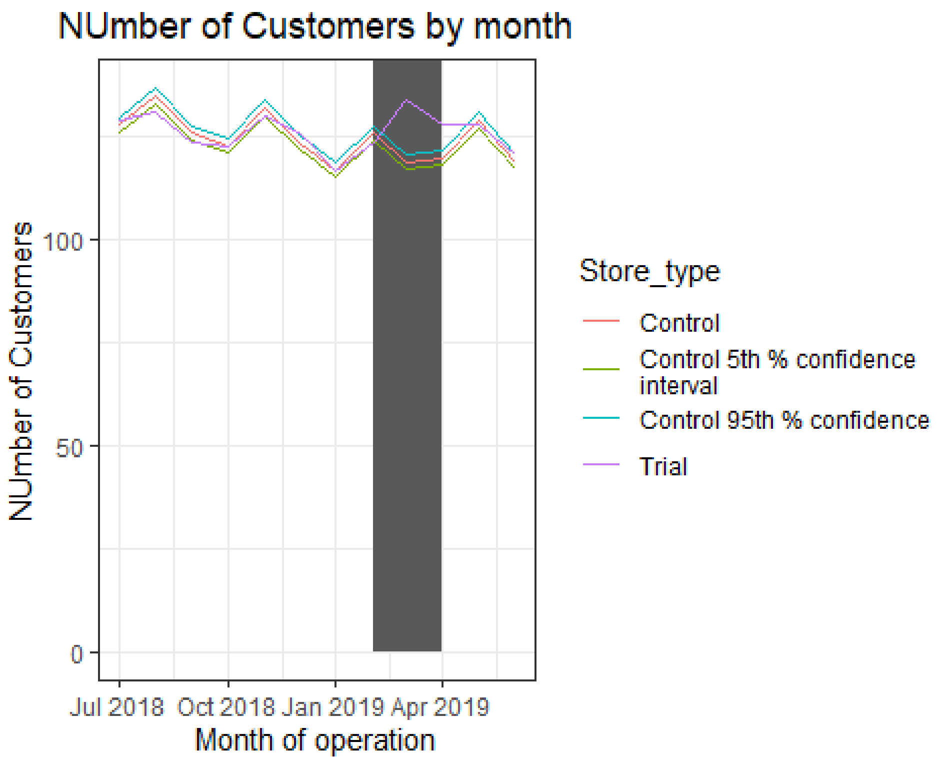


```
scalingFactorForControlSales <- preTrialMeasures[STORE_NBR == trial_store &
                                                   YEARMONTH < 201902,
sum(totSales)]/preTrialMeasures[STORE_NBR == control_store &

YEARMONTH < 201902, sum(totSales)]


scaledControlSales <- measureOverTimeSales[STORE_NBR == control_store, ][ ,

controlSales := totSales * scalingFactorForControlSales]

percentageDiff <- merge(scaledControlSales,pastSales[Store_type == "Trial"],
by = "YEARMONTH")[,percentageDiff := abs(controlSales -
totSales.y)/controlSales ]
stdDev <- sd(percentageDiff[YEARMONTH < 201902 , percentageDiff])

degreesOfFreedom <- 7
percentageDiff[, tValue := abs(totSales.x - mean(totSales.y))/stdDev
               ][, TransactionMonth := TransactionMonth.x
                 ]
pastSales <- pastSales[Store_type %in% c("Trial", "Control"), ]
```

```
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() +
  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 <- measureOverTimeSales
scaledControlCustomers <- measureOverTimeCusts[STORE_NBR == control_store, ][
, controlCustomer := nCustomers * scalingFactorForControlCust]

percentageDiff <- merge(scaledControlCustomers,pastCustomers[Store_type ==
"Trial"],
                        by = "YEARMONTH")[,percentageDiff :=
abs(controlCustomer - nCustomers.y)/controlCustomer ]
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"), ]


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

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 = "NUmber of Customers", title = "NUmber
of Customers by month")
```



NUmber of Customers by month