Task 2

library(data.table)

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

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

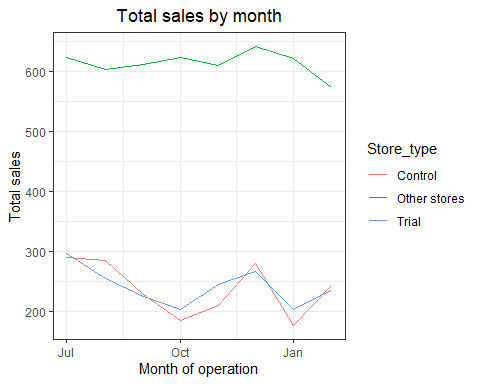
##   
## Attaching package: 'dplyr'

## 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)) +  
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': 70.750000 (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'

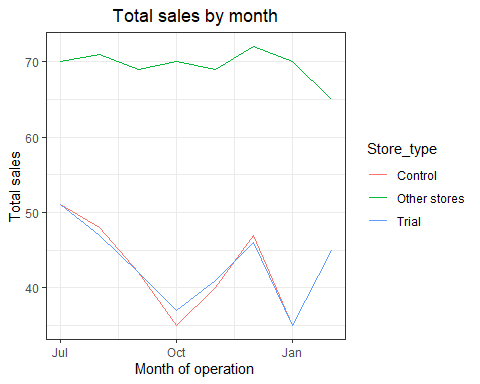
## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store\_type,  
## ifelse(STORE\_NBR == : Group 9 column 'nCustomers': 71.509506 (type  
## 'double') at RHS position 1 truncated (precision lost) when assigning to  
## type 'integer'

## 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'

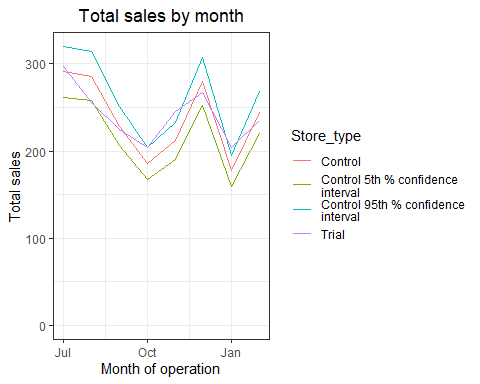
## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store\_type,  
## ifelse(STORE\_NBR == : Group 11 column 'nCustomers': 70.865900 (type  
## 'double') at RHS position 1 truncated (precision lost) when assigning to  
## type 'integer'

## 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'

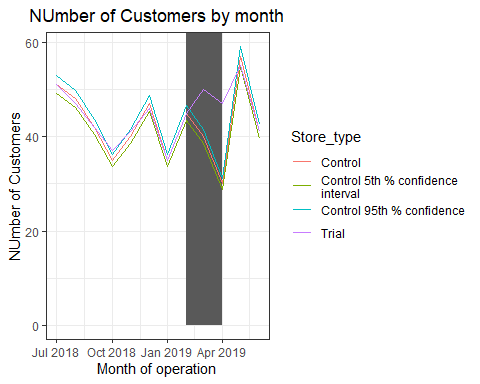
ggplot(pastCustomers, aes(TransactionMonth, nCustomers, color = Store\_type)) +  
 geom\_line() +  
 labs(x = "Month of operation", y = "Total sales", title = "Total sales by month")



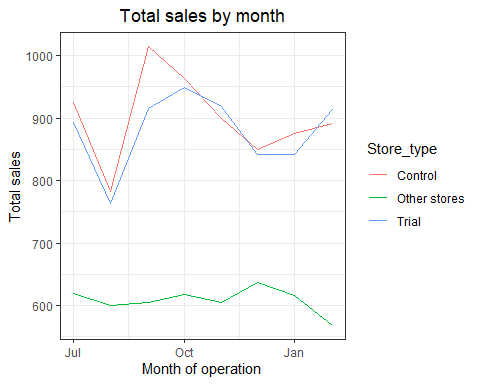
scalingFactorForControlSales <- preTrialMeasures[STORE\_NBR == trial\_store &  
YEARMONTH < 201902, sum(totSales)]/preTrialMeasures[STORE\_NBR == control\_store &  
YEARMONTH < 201902, sum(totSales)]  
  
  
scaledControlSales <- measureOverTimeSales[STORE\_NBR == control\_store, ][ ,  
controlSales := 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")



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'

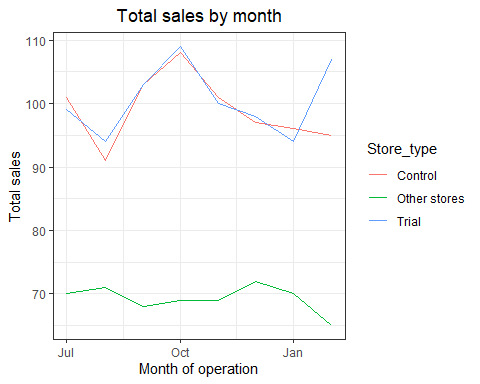
## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store\_type,  
## ifelse(STORE\_NBR == : Group 9 column 'nCustomers': 71.057034 (type  
## 'double') at RHS position 1 truncated (precision lost) when assigning to  
## type 'integer'

## 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'

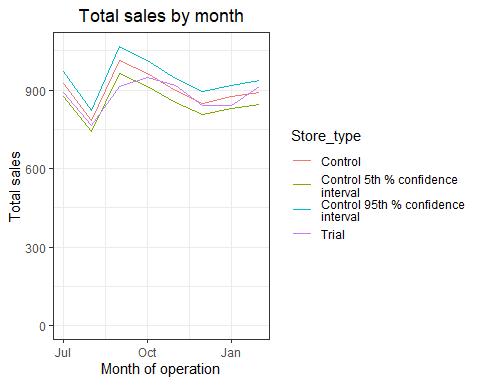
## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store\_type,  
## ifelse(STORE\_NBR == : Group 11 column 'nCustomers': 70.490421 (type  
## 'double') at RHS position 1 truncated (precision lost) when assigning to  
## type 'integer'

## 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'

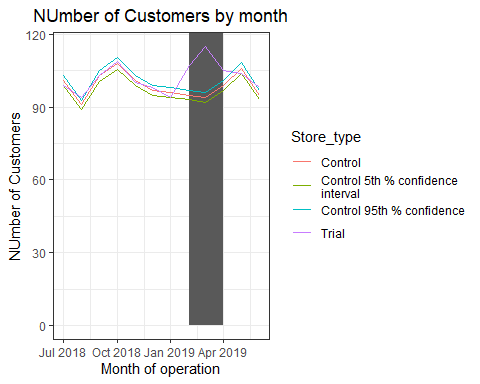
ggplot(pastCustomers, aes(TransactionMonth, nCustomers, color = Store\_type)) +  
 geom\_line() +  
 labs(x = "Month of operation", y = "Total sales", title = "Total sales by month")



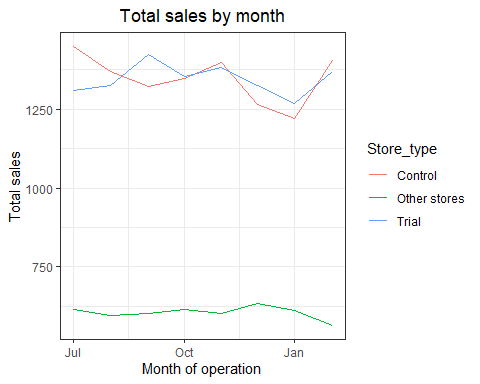
scalingFactorForControlSales <- preTrialMeasures[STORE\_NBR == trial\_store &  
 YEARMONTH < 201902, sum(totSales)]/preTrialMeasures[STORE\_NBR == control\_store &  
 YEARMONTH < 201902, sum(totSales)]  
  
  
scaledControlSales <- measureOverTimeSales[STORE\_NBR == control\_store, ][ ,  
 controlSales := 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'

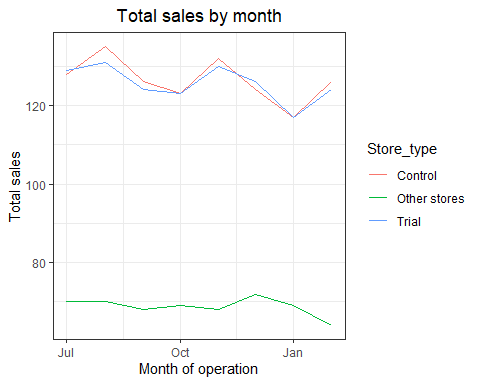
## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store\_type,  
## ifelse(STORE\_NBR == : Group 9 column 'nCustomers': 70.889734 (type  
## 'double') at RHS position 1 truncated (precision lost) when assigning to  
## type 'integer'

## 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'

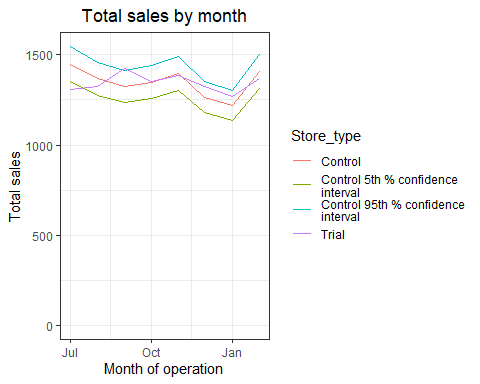
## Warning in `[.data.table`(measureOverTimeSales[, `:=`(Store\_type,  
## ifelse(STORE\_NBR == : Group 11 column 'nCustomers': 70.310345 (type  
## 'double') at RHS position 1 truncated (precision lost) when assigning to  
## type 'integer'

## 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")

