```
library(data.table)
library(ggplot2)
library(tidyr)
data <-fread("D:\\Projects\\Proj#3_Quantium_Data_Analysis_Retail_ChipsCategory/Task_1/Task_1_QVI_dat
head(data)
theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = 0.5))

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

# STEP 4: Calculate metrics per store per month
# Aggregate key metrics for each store and each month
measureOverTime <- data[, .(
  totSales = sum(TOT_SALES),                          # Total revenue in that month
  nCustomers = uniqueN(LYLTY_CARD_NBR),               # Number of unique customers
  nTxn = uniqueN(TXN_ID),                             # Number of transactions
  nChips = sum(PROD_QTY),                             # Total number of chip packets sold
  avgPricePerUnit = sum(TOT_SALES)/sum(PROD_QTY)      # Average price per chip packet
), by = .(STORE_NBR, YEARMONTH)]                      # Group by store and month

# Add two more derived metrics
measureOverTime[, nTxnPerCust := nTxn / nCustomers]     # Average number of transactions per custome
measureOverTime[, nChipsPerTxn := nChips / nTxn]

# STEP 5: Filter to pre-trial period
# Only consider months before February 2019 for finding control stores
preTrialMeasures <- measureOverTime[YEARMONTH < 201902]

# Identify stores that have data for all 12 pre-trial months (July 2018 to Jan 2019)
storesWithFullObs <- unique(measureOverTime[, .N, by = STORE_NBR][N == 12]$STORE_NBR)

# Keep only those stores in our filtered pre-trial data
preTrialMeasures <- preTrialMeasures[STORE_NBR %in% storesWithFullObs]

# STEP 6: Correlation function to compare trend similarity
calculateCorrelation <- function(inputTable, metricCol, storeComparison) {
  calcCorrTable = data.table(Store1 = numeric(), Store2 = numeric(), corr_measure = numeric())
  storeNumbers <- unique(inputTable[, STORE_NBR])

  for (i in storeNumbers) {
    if (i != storeComparison) {
      comparisonData <- inputTable[STORE_NBR %in% c(storeComparison, i),
                                   .(STORE_NBR, YEARMONTH, metric = get(as.character(metricCol)))]
      comparisonData <- dcast(comparisonData, YEARMONTH ~ STORE_NBR, value.var = "metric")
      setnames(comparisonData, c("YEARMONTH", "Trial", "Control"))
      correlation <- cor(comparisonData[["Trial"]], comparisonData[["Control"]])
      calcCorrTable <- rbind(calcCorrTable, data.table(Store1 = storeComparison, Store2 = i, corr_me
    }
  }
  return(calcCorrTable)
}

# STEP 7: Magnitude distance function to compare value closeness
calculateMagnitudeDistance <- function(inputTable, metricCol, storeComparison) {
  calcDistTable = data.table(Store1 = numeric(), Store2 = numeric(), YEARMONTH = numeric(), measure
  storeNumbers <- unique(inputTable[, STORE_NBR])

  for (i in storeNumbers) {
    if (i != storeComparison) {
      calculatedMeasure = data.table(
        Store1 = storeComparison,
        Store2 = i,
        YEARMONTH = inputTable[STORE_NBR == storeComparison, YEARMONTH],
        measure = abs(inputTable[STORE_NBR == storeComparison, get(as.character(metricCol))] -
                      inputTable[STORE_NBR == i, get(as.character(metricCol))])
      )
      calcDistTable <- rbind(calcDistTable, calculatedMeasure)
    }
  }
```

```
  minMaxDist <- calcDistTable[, .(minDist = min(measure), maxDist = max(measure)), by = .(Store1, YE
  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)
}



# LOOPED ANALYSIS: Repeat trial analysis for stores 77, 86 and 88

trial_stores <- c(77, 86, 88)

for (trial_store in trial_stores) {
  # Recalculate similarity scores for the current trial store
  corr_nSales <- calculateCorrelation(preTrialMeasures, "totSales", trial_store)
  corr_nCustomers <- calculateCorrelation(preTrialMeasures, "nCustomers", trial_store)
  magnitude_nSales <- calculateMagnitudeDistance(preTrialMeasures, "totSales", trial_store)
  magnitude_nCustomers <- calculateMagnitudeDistance(preTrialMeasures, "nCustomers", trial_store)

  # Combine scores
  score_nSales <- merge(corr_nSales, magnitude_nSales, by = c("Store1", "Store2"))
  score_nSales[, scoreNSales := (corr_measure * 0.5 + mag_measure * 0.5)]

  score_nCustomers <- merge(corr_nCustomers, magnitude_nCustomers, by = c("Store1", "Store2"))
  score_nCustomers[, scoreNCust := (corr_measure * 0.5 + mag_measure * 0.5)]

  score_Control <- merge(score_nSales[, .(Store1, Store2, scoreNSales)],
                         score_nCustomers[, .(Store1, Store2, scoreNCust)],
                         by = c("Store1", "Store2"))
  score_Control[, finalControlScore := (scoreNSales + scoreNCust)/2]

  # Select best matching control store
  control_store <- score_Control[Store1 == trial_store & Store2 != trial_store][order(-finalControlS
  print(paste("Trial Store:", trial_store, "- Control Store:", control_store))

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

  # Apply scaled sales
  measureOverTimeSales <- measureOverTime
  measureOverTimeSales[STORE_NBR == control_store, controlSales := totSales * scalingFactorForContro

  trialSales <- measureOverTimeSales[STORE_NBR == trial_store, .(YEARMONTH, trialSales = totSales)]
  controlSales <- measureOverTimeSales[STORE_NBR == control_store, .(YEARMONTH, controlSales)]

  percentageDiff <- merge(trialSales, controlSales, by = "YEARMONTH")
  percentageDiff[, percentageDiff := abs(trialSales - controlSales) / controlSales]

  stdDev <- sd(percentageDiff[YEARMONTH < 201902, percentageDiff])
  percentageDiff[, tValue := (percentageDiff - 0) / stdDev]
  percentageDiff[, TransactionMonth := as.Date(paste0(YEARMONTH, "01"), format = "%Y%m%d")]

  # Create plots with confidence bands
  pastSales <- measureOverTimeSales[STORE_NBR %in% c(trial_store, control_store),
    .(totSales = mean(totSales)), by = .(YEARMONTH, STORE_NBR)]
  pastSales[, Store_type := ifelse(STORE_NBR == trial_store, "Trial", "Control")]
  pastSales[, TransactionMonth := as.Date(paste0(YEARMONTH, "01"), format = "%Y%m%d")]

  pastSales_Controls95 <- pastSales[Store_type == "Control"]
  pastSales_Controls95[, totSales := totSales * (1 + stdDev * 2)]
  pastSales_Controls95[, Store_type := "Control 95% Confidence"]

  pastSales_Controls5 <- pastSales[Store_type == "Control"]
  pastSales_Controls5[, totSales := totSales * (1 - stdDev * 2)]
  pastSales_Controls5[, Store_type := "Control 5% Confidence"]

  trialAssessment <- rbind(pastSales, pastSales_Controls95, pastSales_Controls5)
```

```
  trialHighlight <- data.table(
    xmin = as.Date("2019-02-01"),
    xmax = as.Date("2019-04-30"),
    ymin = 0,
    ymax = Inf
  )

  p <- ggplot(trialAssessment, aes(x = TransactionMonth, y = totSales, color = Store_type)) +
    geom_rect(data = trialHighlight, aes(xmin = xmin, xmax = xmax, ymin = ymin, ymax = ymax),
              fill = "grey90", alpha = 0.3, inherit.aes = FALSE) +
    geom_line(size = 1.1) +
    labs(x = "Month", y = "Total Sales", title = paste("Trial vs Control Sales: Store", trial_store)
    theme_minimal()

  ggsave(filename = paste0("Trial_Store_", trial_store, "_sales_plot.png"), plot = p, width = 8, hei
}
```