Quantium Retail Analytics - Task 2: Trial Store Performance Evaluation

```
# Load necessary libraries
library(dplyr)
library(ggplot2)
library(lubridate)
library(tidyr)
library(scales)

# Load the data
qvi <- read.csv("QVI_data.csv", stringsAsFactors = FALSE)
qvi$DATE <- as.Date(qvi$DATE)
qvi$month <- floor_date(qvi$DATE, "month")

# Monthly KPI aggregation
monthly_metrics <- qvi %>%
  group_by(STORE_NBR, month) %>%
  summarise(
    total_sales = sum(TOT_SALES),
    total_customers = n_distinct(LYLTY_CARD_NBR),
    avg_txn_per_customer = n() / n_distinct(LYLTY_CARD_NBR),
    .groups = "drop"
  )

# Similarity function using magnitude distance
get_similarity_score <- function(trial_store, metrics_df) {
  trial_data <- metrics_df %>% filter(STORE_NBR == trial_store)
  other_stores <- metrics_df %>% filter(STORE_NBR != trial_store)

  scores <- other_stores %>%
    group_by(STORE_NBR) %>%
    summarise(
           score_sales = 1 - (sum(abs(total_sales - trial_data$total_sales)) - min(sum(abs(total_sales -
trial_data$total_sales)))) /
                      (max(sum(abs(total_sales - trial_data$total_sales))) - min(sum(abs(total_sales -
trial_data$total_sales)))),
                  score_customers = 1 - (sum(abs(total_customers - trial_data$total_customers)) -
min(sum(abs(total_customers - trial_data$total_customers)))) /
               (max(sum(abs(total_customers - trial_data$total_customers))) - min(sum(abs(total_customers -
trial_data$total_customers)))),
                score_txn = 1 - (sum(abs(avg_txn_per_customer - trial_data$avg_txn_per_customer)) -
min(sum(abs(avg_txn_per_customer - trial_data$avg_txn_per_customer)))) /
                               (max(sum(abs(avg_txn_per_customer - trial_data$avg_txn_per_customer))) -
min(sum(abs(avg_txn_per_customer - trial_data$avg_txn_per_customer)))),
      total_score = (score_sales + score_customers + score_txn) / 3,
      .groups = "drop"
    ) %>% arrange(desc(total_score))

  return(scores)
}

# Trial stores and control store selection
trial_stores <- c(77, 86, 88)
control_stores <- lapply(trial_stores, function(trial) {
  similarity_scores <- get_similarity_score(trial, monthly_metrics)
  top_match <- similarity_scores$STORE_NBR[1]
  return(data.frame(trial_store = trial, control_store = top_match))
})
control_stores_df <- do.call(rbind, control_stores)

# Trial period range
trial_period <- as.Date(c("2019-02-01", "2019-04-30"))

# Comparison function
compare_stores <- function(trial_store, control_store, metrics_df) {
  data <- metrics_df %>% filter(STORE_NBR %in% c(trial_store, control_store)) %>%
    mutate(Store_Type = ifelse(STORE_NBR == trial_store, "Trial", "Control"))

  # Plot total sales
  p1 <- ggplot(data, aes(x = month, y = total_sales, color = Store_Type)) +
    geom_line(linewidth = 1.2) +
    labs(title = paste("Total Sales: Trial Store", trial_store, "vs Control Store", control_store),
         x = "Month", y = "Total Sales") +
    scale_y_continuous(labels = dollar) +
    theme_minimal()
  print(p1)

  # Statistical test
  trial_data <- data %>% filter(month >= trial_period[1] & month <= trial_period[2])
  sales_test <- t.test(
    trial_data$total_sales[trial_data$Store_Type == "Trial"],
    trial_data$total_sales[trial_data$Store_Type == "Control"]
  )
  cat(paste0("
Trial Store ", trial_store, " vs Control Store ", control_store, "
"))
  cat("p-value for sales difference during trial: ", round(sales_test$p.value, 4), "
")
}

# Run comparisons
for (i in 1:nrow(control_stores_df)) {
  compare_stores(control_stores_df$trial_store[i], control_stores_df$control_store[i], monthly_metrics)
}
```