```
> # Load the new dataset
> QVI_data <- fread("QVI_data.csv")  # assuming CSV; adjust if Excel
> head(QVI_data)
   LYLTY_CARD_NBR       DATE STORE_NBR TXN_ID PROD_NBR
            <int>     <IDat>     <int>  <int>    <int>
1:           1000 2018-10-17         1      1        5
2:           1002 2018-09-16         1      2       58
3:           1003 2019-03-07         1      3       52
4:           1003 2019-03-08         1      4      106
5:           1004 2018-11-02         1      5       96
6:           1005 2018-12-28         1      6       86
                                   PROD_NAME PROD_QTY TOT_SALES PACK_SIZE
                                      <char>    <int>     <num>     <int>
1: Natural Chip        Compny SeaSalt175g        2       6.0       175
2:  Red Rock Deli Chikn&Garlic Aioli 150g        1       2.7       150
3:  Grain Waves Sour    Cream&Chives 210G        1       3.6       210
4: Natural ChipCo      Hony Soy Chckn175g        1       3.0       175
5:         WW Original Stacked Chips 160g        1       1.9       160
6:                    Cheetos Puffs 165g        1       2.8       165
        BRAND              LIFESTAGE PREMIUM_CUSTOMER
       <char>                 <char>           <char>
1:    NATURAL  YOUNG SINGLES/COUPLES          Premium
2:        RRD  YOUNG SINGLES/COUPLES       Mainstream
3:    GRNWVES         YOUNG FAMILIES           Budget
4:    NATURAL         YOUNG FAMILIES           Budget
5: WOOLWORTHS  OLDER SINGLES/COUPLES       Mainstream
6:    CHEETOS MIDAGE SINGLES/COUPLES       Mainstream
> # Convert DATE to Date if not already
> QVI_data[, DATE := as.Date(DATE)]
>
> # Extract year-month for aggregation
> QVI_data[, MONTH := format(DATE, "%Y-%m")]
>
> # Aggregate per store per month
> monthly_store_metrics <- QVI_data[, .(
+   Total_Sales = sum(TOT_SALES, na.rm = TRUE),
+   Num_Customers = uniqueN(LYLTY_CARD_NBR),
+   Avg_Transactions_Per_Customer = .N / uniqueN(LYLTY_CARD_NBR)
+ ), by = .(STORE_NBR, MONTH)]
>
> head(monthly_store_metrics)
   STORE_NBR   MONTH Total_Sales Num_Customers Avg_Transactions_Per_Customer
       <int>  <char>       <num>         <int>                         <num>
1:         1 2018-10       188.1            44                      1.022727
2:         1 2018-09       278.8            59                      1.050847
3:         1 2019-03       192.9            45                      1.088889
4:         1 2018-11       192.6            46                      1.021739
5:         1 2018-12       189.6            42                      1.119048
6:         1 2018-07       206.9            49                      1.061224
>
> # Function to calculate similarity between trial and potential control store
> compare_stores <- function(trial_store, candidate_store, metric = "Total_Sales") {
+
+   trial_data <- monthly_store_metrics[STORE_NBR == trial_store, ..metric]
+   control_data <- monthly_store_metrics[STORE_NBR == candidate_store, ..metric]
+
+   # Make sure same months are compared
+   trial_data <- trial_data[!is.na(trial_data[[1]])]
+   control_data <- control_data[!is.na(control_data[[1]])]
+
+   # Calculate Pearson correlation
+   cor_value <- cor(trial_data[[1]], control_data[[1]])
```

```
+
+   # Optionally: magnitude distance
+   mag_distance <- 1 - (abs(mean(trial_data[[1]]) - mean(control_data[[1]])) /
+                        (max(trial_data[[1]], control_data[[1]]) - min(trial_data[[1]], control_data[[1]])))
+
+   return(list(Pearson_Correlation = cor_value,
+               Magnitude_Distance = mag_distance))
+ }
> trial_stores <- c(77, 86, 88)
> candidate_controls <- setdiff(unique(QVI_data$STORE_NBR), trial_stores)
>
> # Example: Compare trial store 77 to all candidate controls
> results <- lapply(candidate_controls, function(c) compare_stores(77, c))
Error in cor(trial_data[[1]], control_data[[1]]) :
  incompatible dimensions
> names(results) <- candidate_controls
Error: object 'results' not found
> results_df <- data.table(Store = candidate_controls, do.call(rbind, lapply(results, as.data.frame)))
Error: object 'results' not found
> head(results_df[order(-Pearson_Correlation)])  # top matches
Error: object 'results_df' not found
> compare_stores <- function(trial_store, candidate_store, metric = "Total_Sales") {
+
+   trial_data <- monthly_store_metrics[STORE_NBR == trial_store, .(MONTH, value = get(metric))]
+   control_data <- monthly_store_metrics[STORE_NBR == candidate_store, .(MONTH, value = get(metric))]
+
+   # Merge on MONTH to align
+   merged_data <- merge(trial_data, control_data, by = "MONTH", suffixes = c("_trial", "_control"))
+
+   # If after merging we have no overlapping months, return NA
+   if(nrow(merged_data) == 0) return(list(Pearson_Correlation = NA, Magnitude_Distance = NA))

+

+   # Pearson correlation

+   cor_value <- cor(merged_data$value_trial, merged_data$value_control)

+

+   # Magnitude distance

+   mag_distance <- 1 - (abs(mean(merged_data$value_trial) - mean(merged_data$value_control)) /

+                        (max(c(merged_data$value_trial, merged_data$value_control)) - 
min(c(merged_data$value_trial, merged_data$value_control))))

+

+   return(list(Pearson_Correlation = cor_value,

+               Magnitude_Distance = mag_distance))

+ }

> results <- lapply(candidate_controls, function(c) compare_stores(77, c))

Warning messages:

1: In cor(merged_data$value_trial, merged_data$value_control) :

  the standard deviation is zero

2: In cor(merged_data$value_trial, merged_data$value_control) :

  the standard deviation is zero

> names(results) <- candidate_controls

> results_df <- data.table(Store = candidate_controls, do.call(rbind, lapply(results, as.data.frame)))

> head(results_df[order(-Pearson_Correlation)])

   Store Pearson_Correlation Magnitude_Distance

   <int>               <num>              <num>

1:    11           1.0000000         0.12046142

2:    31           1.0000000         0.04784789

3:    41           0.7622919         0.72584034

4:    35           0.6997078         0.40311144

5:   167           0.6960754         0.64280216

6:   184           0.6451178         0.15559512

> trial_months <- c("2019-01", "2019-02", "2019-03")  # adjust based on your data
> # Replace with the selected control stores
> control_for_77 <- 12   # example, replace with actual selected store
> control_for_86 <- 34
> control_for_88 <- 56
>
> # Filter metrics
> trial_data <- monthly_store_metrics[STORE_NBR %in% c(77, 86, 88) & MONTH %in% trial_months]
> control_data <- monthly_store_metrics[STORE_NBR %in% c(control_for_77, control_for_86,
control_for_88) & MONTH %in% trial_months]
> t.test(trial_data$Total_Sales, control_data$Total_Sales)

	Welch Two Sample t-test

data:  trial_data$Total_Sales and control_data$Total_Sales
t = 2.1454, df = 9.5192, p-value = 0.05886
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -17.08392 764.61725
sample estimates:
mean of x mean of y
 845.9000  472.1333

> t.test(trial_data$Num_Customers, control_data$Num_Customers)

	Welch Two Sample t-test

data:  trial_data$Num_Customers and control_data$Num_Customers
t = 2.742, df = 11.669, p-value = 0.01826
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  7.800377 69.088512
```

```
sample estimates:
mean of x mean of y 
 91.22222  52.77778 

> t.test(trial_data$Avg_Transactions_Per_Customer, control_data$Avg_Transactions_Per_Customer)

	Welch Two Sample t-test

data:  trial_data$Avg_Transactions_Per_Customer and control_data$Avg_Transactions_Per_Customer
t = 1.8662, df = 15.716, p-value = 0.08078
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.01330963  0.20678016
sample estimates:
mean of x mean of y 
 1.209952  1.113217 

> library(ggplot2)
> 
> # Combine for plotting
> plot_data <- rbind(trial_data, control_data)
> plot_data$Type <- ifelse(plot_data$STORE_NBR %in% c(77, 86, 88), "Trial", "Control")
> 
> # Total Sales over months
> ggplot(plot_data, aes(x = MONTH, y = Total_Sales, color = Type, group = STORE_NBR)) +
+   geom_line(size = 1.2) +
+   labs(title = "Trial vs Control Stores: Total Sales", x = "Month", y = "Total Sales") +
+   theme_minimal()
Warning message:
Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
```

```
ℹ Please use `linewidth` instead.

This warning is displayed once every 8 hours.

Call `lifecycle::last_lifecycle_warnings()` to see where this warning was

generated.

> Repeat similar plots for Nu

Error: unexpected symbol in "Repeat similar"

>
```