```
import numpy as np
import pandas as pd
import matplotlib.pyplot as mp
import seaborn as sb
import re
from datetime import datetime, timedelta
import os

trans = pd.read_csv("QVI_data.csv")

trans.head()
```

	LYLTY_CARD_NBR	DATE	STORE_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	PACK_SIZE	BRAND	LIFESTAGE	PI
0	1000	2018-10-17	1	1	5	Natural Chip Compny SeaSalt175g	2	6.0	175	NATURAL	YOUNG SINGLES/COUPLES	
1	1002	2018-09-16	1	2	58	Red Rock Deli Chikn&Garlic Aioli 150g	1	2.7	150	RRD	YOUNG SINGLES/COUPLES	
2	1003	2019-03-07	1	3	52	Grain Waves Sour Cream&Chives 210G	1	3.6	210	GRNWVES	YOUNG FAMILIES	
						Natural						

Define trial period

```
trans['DATE'] = pd.to_datetime(trans['DATE'])

trial_start = pd.to_datetime('2019-02-01')
trial_end = pd.to_datetime('2019-04-30')

pre_trial_data = trans[trans['DATE']<trial_start]
```

Create month metrics(total sales and customer count)

```
pre_trial_data['MONTH'] = pre_trial_data['DATE'].dt.to_period('M')

monthly_sales = pre_trial_data.groupby(['STORE_NBR','MONTH'])['TOT_SALES'].sum().reset_index()

monthly_customers = pre_trial_data.groupby(['STORE_NBR','MONTH'])['LYLTY_CARD_NBR'].nunique().reset_index()
monthly_customers.rename(columns={'LYLTY_CARD_NBR':'n_customers'}, inplace=True)

monthly_metrics = pd.merge(monthly_sales, monthly_customers, on=['STORE_NBR','MONTH'])
```

```
<ipython-input-10-5c638fb7a503>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  pre_trial_data['MONTH'] = pre_trial_data['DATE'].dt.to_period('M')
```

Define similarity

```
from scipy.stats import pearsonr
import numpy as np

def calculate_similarity(trial_store, metric_df, metric_col):
    # Get trial store metric values
    trial = metric_df[metric_df['STORE_NBR'] == trial_store].sort_values('MONTH')

    similarities = []

    for store in metric_df['STORE_NBR'].unique():
        if store == trial_store:
            continue
```

```
        control = metric_df[metric_df['STORE_NBR'] == store].sort_values('MONTH')
        # Make sure both stores have the same months
        merged = pd.merge(trial, control, on='MONTH', suffixes=('_trial', '_control'))

        # Pearson correlation
        if len(merged) < 2:
            corr = 0
        else:
            corr, _ = pearsonr(merged[f'{metric_col}_trial'], merged[f'{metric_col}_control'])

        # Magnitude distance
        diff = abs(merged[f'{metric_col}_trial'] - merged[f'{metric_col}_control'])
        dist = 1 - (diff.sum() - diff.min()) / (diff.max() - diff.min() + 1e-6)

        similarities.append({
            'control_store': store,
            'correlation': corr,
            'magnitude_distance': dist
        })

    return pd.DataFrame(similarities)
```

Apply to store 77

```
sales_similarity_77 = calculate_similarity(77, monthly_metrics[['STORE_NBR', 'MONTH', 'TOT_SALES']], 'TOT_SALES')
sales_similarity_77.sort_values(by=['correlation', 'magnitude_distance'], ascending=False).head()

customer_similarity_77 = calculate_similarity(77, monthly_metrics[['STORE_NBR', 'MONTH', 'n_customers']], 'n_customers')
customer_similarity_77.sort_values(by=['correlation', 'magnitude_distance'], ascending=False).head()
```

```
<ipython-input-11-a0e17192d281>:21: ConstantInputWarning: An input array is constant; the correlation coefficient is not defined.
  corr, _ = pearsonr(merged[f'{metric_col}_trial'], merged[f'{metric_col}_control'])
<ipython-input-11-a0e17192d281>:21: ConstantInputWarning: An input array is constant; the correlation coefficient is not defined.
  corr, _ = pearsonr(merged[f'{metric_col}_trial'], merged[f'{metric_col}_control'])
```

	control_store	correlation	magnitude_distance
230	233	0.990358	-1.499999
116	119	0.983267	-110.666629
251	254	0.916208	-4.714285
110	113	0.901348	-47.857136
82	84	0.858571	-4.999999

Apply to store 86

```
sales_similarity_86 = calculate_similarity(86, monthly_metrics[['STORE_NBR', 'MONTH', 'TOT_SALES']], 'TOT_SALES')
sales_similarity_86.sort_values(by=['correlation', 'magnitude_distance'], ascending=False).head()

customer_similarity_86 = calculate_similarity(86, monthly_metrics[['STORE_NBR', 'MONTH', 'n_customers']], 'n_customers')
customer_similarity_86.sort_values(by=['correlation', 'magnitude_distance'], ascending=False).head()
```

```
<ipython-input-11-a0e17192d281>:21: ConstantInputWarning: An input array is constant; the correlation coefficient is not defined.
  corr, _ = pearsonr(merged[f'{metric_col}_trial'], merged[f'{metric_col}_control'])
<ipython-input-11-a0e17192d281>:21: ConstantInputWarning: An input array is constant; the correlation coefficient is not defined.
  corr, _ = pearsonr(merged[f'{metric_col}_trial'], merged[f'{metric_col}_control'])
```

	control_store	correlation	magnitude_distance
152	155	0.942876	-2.333332
111	114	0.855339	-3.300000
257	260	0.846502	-25.499998
173	176	0.796380	-22.866665
106	109	0.770778	-1.750000

Apply to store 88

```
sales_similarity_88 = calculate_similarity(88, monthly_metrics[['STORE_NBR', 'MONTH', 'TOT_SALES']], 'TOT_SALES')
sales_similarity_88.sort_values(by=['correlation', 'magnitude_distance'], ascending=False).head()

customer_similarity_88 = calculate_similarity(88, monthly_metrics[['STORE_NBR', 'MONTH', 'n_customers']], 'n_customers')
customer_similarity_88.sort_values(by=['correlation', 'magnitude_distance'], ascending=False).head()
```

```
<ipython-input-11-a0e17192d281>:21: ConstantInputWarning: An input array is constant; the correlation coefficient is not defined.
  corr, _ = pearsonr(merged[f'{metric_col}_trial'], merged[f'{metric_col}_control'])
<ipython-input-11-a0e17192d281>:21: ConstantInputWarning: An input array is constant; the correlation coefficient is not defined.
  corr, _ = pearsonr(merged[f'{metric_col}_trial'], merged[f'{metric_col}_control'])
```

	control_store	correlation	magnitude_distance
234	237	0.947326	-1.749999
13	14	0.942976	-78.999991
175	178	0.939466	-21.666663
34	35	0.899594	-91.333318
110	113	0.862632	-15.599998

```
control_stores = {
    77: 233,
    86: 155,
    88: 178
}
```

Create monthly metrics for trial period

```
# Filter trial period data
trial_data = trans[(trans['DATE'] >= trial_start) & (trans['DATE'] <= trial_end)]
trial_data['MONTH'] = trial_data['DATE'].dt.to_period('M')

# Total sales
monthly_sales_trial = trial_data.groupby(['STORE_NBR', 'MONTH'])['TOT_SALES'].sum().reset_index()

# Unique customers
monthly_customers_trial = trial_data.groupby(['STORE_NBR', 'MONTH'])['LYLTY_CARD_NBR'].nunique().reset_index()
monthly_customers_trial.rename(columns={'LYLTY_CARD_NBR': 'n_customers'}, inplace=True)

# Transactions per customer
transactions_per_customer = (
    trial_data.groupby(['STORE_NBR', 'MONTH'])['TXN_ID'].nunique().reset_index()
    .merge(monthly_customers_trial, on=['STORE_NBR', 'MONTH'])
)
transactions_per_customer['txn_per_customer'] = (
    transactions_per_customer['TXN_ID'] / transactions_per_customer['n_customers']
)
```

```
<ipython-input-20-a54c3f7079ad>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  trial_data['MONTH'] = trial_data['DATE'].dt.to_period('M')
```

Control trial vs control metrics

```
def compare_trial_control(trial_store, control_store, monthly_sales, monthly_customers, txn_per_cust):
    # Filter for stores
    sales = monthly_sales[monthly_sales['STORE_NBR'].isin([trial_store, control_store])]
    customers = monthly_customers[monthly_customers['STORE_NBR'].isin([trial_store, control_store])]
    txns = txn_per_cust[txn_per_cust['STORE_NBR'].isin([trial_store, control_store])]

    # Add store labels
    sales['store_type'] = sales['STORE_NBR'].map({trial_store: 'Trial', control_store: 'Control'})
    customers['store_type'] = customers['STORE_NBR'].map({trial_store: 'Trial', control_store: 'Control'})
    txns['store_type'] = txns['STORE_NBR'].map({trial_store: 'Trial', control_store: 'Control'})

    # Group and compare means
    sales_summary = sales.groupby('store_type')['TOT_SALES'].mean().reset_index()
```

```
    customer_summary = customers.groupby('store_type')['n_customers'].mean().reset_index()
    txn_summary = txns.groupby('store_type')['txn_per_customer'].mean().reset_index()

    return sales_summary, customer_summary, txn_summary
```

Apply for store 77

```
sales_sum_77, cust_sum_77, txn_sum_77 = compare_trial_control(
    77, control_stores[77],
    monthly_sales_trial, monthly_customers_trial, transactions_per_customer
)

print("Total Sales Comparison:\n", sales_sum_77)
print("\nCustomer Count Comparison:\n", cust_sum_77)
print("\nTransactions per Customer Comparison:\n", txn_sum_77)
```

```
Total Sales Comparison:
   store_type   TOT_SALES
0    Control  200.566667
1      Trial  259.000000

Customer Count Comparison:
   store_type  n_customers
0    Control    38.333333
1      Trial    47.333333

Transactions per Customer Comparison:
   store_type  txn_per_customer
0    Control          1.045370
1      Trial          1.040426
<ipython-input-21-19606c82b18e>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  sales['store_type'] = sales['STORE_NBR'].map({trial_store: 'Trial', control_store: 'Control'})
<ipython-input-21-19606c82b18e>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  customers['store_type'] = customers['STORE_NBR'].map({trial_store: 'Trial', control_store: 'Control'})
<ipython-input-21-19606c82b18e>:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  txns['store_type'] = txns['STORE_NBR'].map({trial_store: 'Trial', control_store: 'Control'})
```

Apply for store 86

```
sales_sum_86, cust_sum_86, txn_sum_86 = compare_trial_control(
    86, control_stores[86],
    monthly_sales_trial, monthly_customers_trial, transactions_per_customer
)

print("Total Sales Comparison:\n", sales_sum_86)
print("\nCustomer Count Comparison:\n", cust_sum_86)
print("\nTransactions per Customer Comparison:\n", txn_sum_86)
```

```
Total Sales Comparison:
   store_type   TOT_SALES
0    Control  846.733333
1      Trial  929.400000

Customer Count Comparison:
   store_type  n_customers
0    Control         96.0
1      Trial        109.0

Transactions per Customer Comparison:
   store_type  txn_per_customer
0    Control          1.261077
1      Trial          1.235704
```

```
<ipython-input-21-19606c82b18e>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  sales['store_type'] = sales['STORE_NBR'].map({trial_store: 'Trial', control_store: 'Control'})
<ipython-input-21-19606c82b18e>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  customers['store_type'] = customers['STORE_NBR'].map({trial_store: 'Trial', control_store: 'Control'})
<ipython-input-21-19606c82b18e>:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  txns['store_type'] = txns['STORE_NBR'].map({trial_store: 'Trial', control_store: 'Control'})
```

Apply for store 88

```
sales_sum_88, cust_sum_88, txn_sum_88 = compare_trial_control(
    88, control_stores[88],
    monthly_sales_trial, monthly_customers_trial, transactions_per_customer
)

print("Total Sales Comparison:\n", sales_sum_88)
print("\nCustomer Count Comparison:\n", cust_sum_88)
print("\nTransactions per Customer Comparison:\n", txn_sum_88)
```

```
Total Sales Comparison:
   store_type    TOT_SALES
0    Control  1049.133333
1      Trial  1428.933333

Customer Count Comparison:
   store_type  n_customers
0    Control   112.666667
1      Trial   128.666667

Transactions per Customer Comparison:
   store_type  txn_per_customer
0    Control          1.291494
1      Trial          1.253563
<ipython-input-21-19606c82b18e>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  sales['store_type'] = sales['STORE_NBR'].map({trial_store: 'Trial', control_store: 'Control'})
<ipython-input-21-19606c82b18e>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  customers['store_type'] = customers['STORE_NBR'].map({trial_store: 'Trial', control_store: 'Control'})
<ipython-input-21-19606c82b18e>:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  txns['store_type'] = txns['STORE_NBR'].map({trial_store: 'Trial', control_store: 'Control'})
```

Add statistical test

```
from scipy.stats import ttest_ind

def t_test_metric(trial_store, control_store, df, metric):
    trial_vals = df[(df['STORE_NBR'] == trial_store)][metric]
    control_vals = df[(df['STORE_NBR'] == control_store)][metric]

    t_stat, p_val = ttest_ind(trial_vals, control_vals, equal_var=False)
    return t_stat, p_val
```

statistical test for store 77

```
t_sales, p_sales = t_test_metric(77, control_stores[77], monthly_sales_trial, 'TOT_SALES')
t_customers, p_customers = t_test_metric(77, control_stores[77], monthly_customers_trial, 'n_customers')

print(f"Sales p-value: {p_sales}")
print(f"Customers p-value: {p_customers}")
```

```
Sales p-value: 0.12608087519542935
Customers p-value: 0.16914521460203255
```

statistical test for store 86

```
t_sales, p_sales = t_test_metric(86, control_stores[86], monthly_sales_trial, 'TOT_SALES')
t_customers, p_customers = t_test_metric(86, control_stores[86], monthly_customers_trial, 'n_customers')

print(f"Sales p-value: {p_sales}")
print(f"Customers p-value: {p_customers}")
```

```
Sales p-value: 0.25235000204902347
Customers p-value: 0.03299600009564686
```

statistical test for store 88

```
t_sales, p_sales = t_test_metric(88, control_stores[88], monthly_sales_trial, 'TOT_SALES')
t_customers, p_customers = t_test_metric(88, control_stores[88], monthly_customers_trial, 'n_customers')

print(f"Sales p-value: {p_sales}")
print(f"Customers p-value: {p_customers}")
```

```
Sales p-value: 0.0008605187858916971
Customers p-value: 0.018230469844410273
```

Start coding or generate with AI.