```
import pandas as pd
import numpy as np

# Load the transaction data
transaction_data = pd.read_csv(r"D:\Sayali\QuantiQuantium -Internship\QVI_data.csv")

# Convert date columns to datetime type if necessary
transaction_data['DATE'] = pd.to_datetime(transaction_data['DATE'])

# Filter data for the pre-trial period
pre_trial_data = transaction_data[transaction_data['DATE'] < '2019-02-01']

# Calculate summary statistics for each store
store_metrics = pre_trial_data.groupby('STORE_NBR').agg({
    'TOT_SALES': 'sum',
    'LYLTY_CARD_NBR': 'nunique',
    'TXN_ID': 'count'
}).reset_index()

# Rename columns for clarity
store_metrics.columns = ['store_number', 'total_sales', 'num_customers', 'num_transactions']

# Calculate transactions per customer
store_metrics['transactions_per_customer'] = store_metrics['num_transactions'] /
store_metrics['num_customers']

# Print the summary statistics
print(store_metrics)

# Extract metrics for trial stores
trial_stores = store_metrics[store_metrics['store_number'].isin([77, 86, 88])]
```

```
print(trial_stores)


# Define a function to calculate similarity score (e.g., Euclidean distance)
def calculate_similarity(row, trial_store):
    return np.sqrt(
        (row['total_sales'] - trial_store['total_sales'])**2 +
        (row['num_customers'] - trial_store['num_customers'])**2 +
        (row['transactions_per_customer'] - trial_store['transactions_per_customer'])**2
    )


# Calculate similarity for each trial store and find control stores
control_stores = {}
for index, trial_store in trial_stores.iterrows():
    store_metrics['similarity_score'] = store_metrics.apply(lambda row: calculate_similarity(row, trial_store), axis=1)
    control_store = store_metrics[(store_metrics['store_number'] != trial_store['store_number']) & (store_metrics['store_number'].isin([77, 86, 88]) == False)].sort_values('similarity_score').iloc[0]
    control_stores[trial_store['store_number']] = control_store['store_number']


print(control_stores)


# Filter data for the trial period
trial_data = transaction_data[(transaction_data['DATE'] >= '2019-02-01') & (transaction_data['DATE'] <= '2019-04-30')]


# Aggregate sales data for trial stores
trial_performance = trial_data[trial_data['STORE_NBR'].isin([77, 86, 88])].groupby('STORE_NBR')['TOT_SALES'].sum().reset_index()
trial_performance.columns = ['STORE_NBR', 'TOT_SALES_trial']


# Aggregate sales data for control stores
```

```
control_performance =
trial_data[trial_data['STORE_NBR'].isin(control_stores.values())].groupby('STORE_NBR')['TOT_SALES']
.sum().reset_index()

control_performance.columns = ['STORE_NBR', 'TOT_SALES_control']


# Print the aggregated sales data for verification

print(trial_performance)

print(control_performance)


# Merge trial and control performance

performance_comparison = pd.merge(trial_performance, control_performance, left_index=True,
right_index=True, suffixes=('_trial', '_control'))

print(performance_comparison)


import seaborn as sns

import matplotlib.pyplot as plt


# Prepare data for visualization

performance_melted = performance_comparison.melt(

    id_vars='STORE_NBR_trial',

    value_vars=['TOT_SALES_trial', 'TOT_SALES_control'],

    var_name='type',

    value_name='TOT_SALES'

)


# Ensure performance_melted is correctly populated

print(performance_melted)


# Visualize sales performance

sns.barplot(x='STORE_NBR_trial', y='TOT_SALES', hue='type', data=performance_melted)

plt.title('Sales Performance Comparison')

plt.xlabel('Store Number')
```

```
plt.ylabel('Sales')

plt.show()
```

Summary Statistics Calculation:

The provided code begins by loading transaction data from a CSV file and converting date columns to the datetime data type. It then filters the data to include only transactions before February 1, 2019, representing the pre-trial period. Summary statistics are calculated for each store during this period, including total sales, the number of unique customers (based on loyalty card numbers), and the total number of transactions. These metrics provide an overview of each store's performance leading up to the trial period.

Trial Stores Identification:

Among the stores, three trial stores (Store 77, Store 86, and Store 88) are identified based on criteria that are not explicitly mentioned in the provided code. This likely involves specific attributes or characteristics that make these stores suitable for the trial.

Control Stores Selection:

Control stores are chosen for each trial store using a similarity analysis approach. The code defines a function to calculate a similarity score between each store and its corresponding trial store based on factors such as total sales, number of customers, and transactions per customer. Control stores are then selected based on their similarity to the trial stores in terms of these metrics. This ensures that the selected control stores closely resemble the trial stores in terms of their pre-trial performance.

Trial and Control Stores Sales Comparison:

Next, the code filters transaction data to include only transactions during the trial period (February 1, 2019, to April 30, 2019). It aggregates sales data for both trial and control stores during this period, providing a comparison of their sales performance. This comparison helps evaluate the impact of any changes or interventions introduced during the trial period.

Visualization of Sales Performance:

Finally, the aggregated sales data for trial and control stores are visualized using a bar plot. This visualization allows for a clear comparison of total sales between trial and control stores, making it easier to identify any differences or patterns in sales performance during the trial period.

Conclusion:

In conclusion, the analysis provides valuable insights into the sales performance of trial stores compared to their respective control stores before and during the trial period. By identifying suitable control stores and comparing their sales performance, the effectiveness of any interventions or changes introduced during the trial period can be evaluated. This information can inform decision-making and strategy development to optimize sales and performance in retail settings.