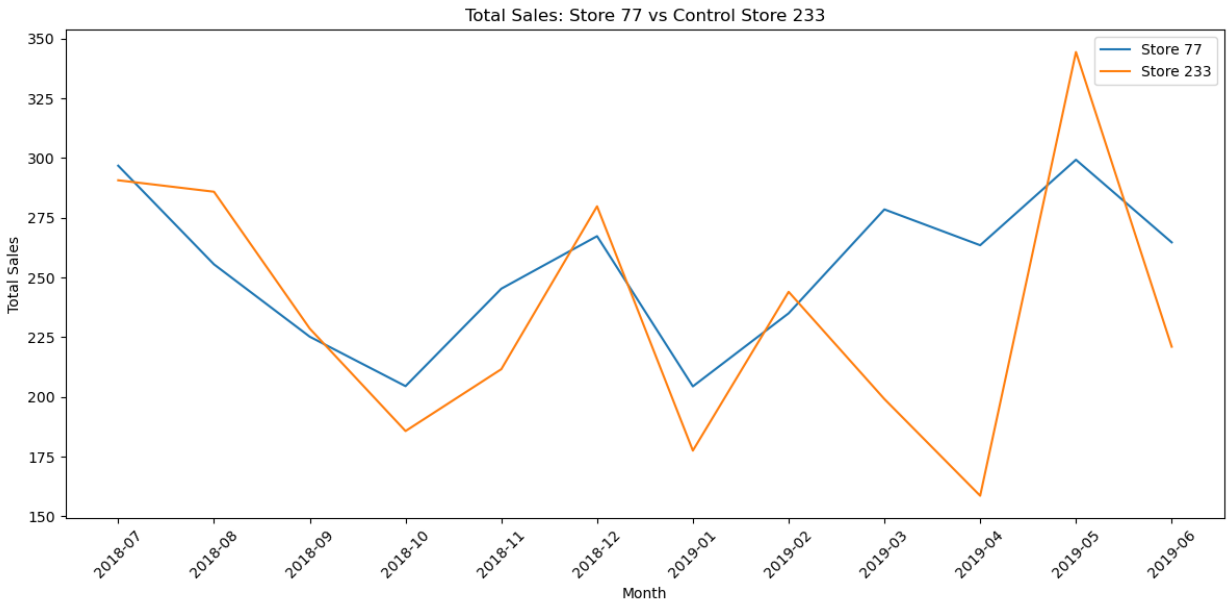
Data Analyst Task 2

```python
import pandas as pd
import matplotlib.pyplot as plt

# STEP 1: Load the QVI_data.csv file
df = pd.read_csv('QVI_data.csv')

# Convert DATE to datetime and create MONTH column
df['DATE'] = pd.to_datetime(df['DATE'])
df['MONTH'] = df['DATE'].dt.to_period('M')

# STEP 2: Aggregate monthly metrics by store
monthly = df.groupby(['STORE_NBR', 'MONTH']).agg(
    TOT_SALES=('TOT_SALES', 'sum'),
    NUM_CUSTOMERS=('LYLTY_CARD_NBR', 'nunique'),
    NUM_TXNS=('TXN_ID', 'nunique')
).reset_index()

# Calculate average transactions per customer
monthly['AVG_TXN_PER_CUST'] = monthly['NUM_TXNS'] / monthly['NUM_CUSTOMERS']

# STEP 3: Function to find the best control store for a trial store
def find_best_control(trial_store, metric='TOT_SALES', trial_start='2019-02'):
    trial_data = monthly[(monthly['STORE_NBR'] == trial_store) & (monthly['MONTH'] <
trial_start)]
    scores = {}

    for store in monthly['STORE_NBR'].unique():
        if store == trial_store:
            continue
        control_data = monthly[(monthly['STORE_NBR'] == store) & (monthly['MONTH'] <
trial_start)]
        merged = pd.merge(trial_data, control_data, on='MONTH', suffixes=('_trial',
'_control'))

        if not merged.empty:
            correlation = merged[f'{metric}_trial'].corr(merged[f'{metric}_control'])
            scores[store] = correlation

    if scores:
        best_match = max(scores, key=scores.get)
        return best_match, scores[best_match]
    else:
        return None, None

# STEP 4: Find control stores for each trial store
trial_stores = [77, 86, 88]
for trial in trial_stores:
    match, score = find_best_control(trial)
```

```
    print(f"Best control for Store {trial}: Store {match} (correlation =
{score:.2f})")

# STEP 5: OPTIONAL - Plotting example (for Store 77 and its control)
# Replace 'x' with your matched control store number for Store 77
trial_store = 77
control_store = 233  # Replace this with your result from above

comparison = monthly[(monthly['STORE_NBR'].isin([trial_store, control_store]))]
comparison = comparison.copy()
comparison['MONTH'] = comparison['MONTH'].astype(str)

plt.figure(figsize=(12, 6))
for store in [trial_store, control_store]:
    store_data = comparison[comparison['STORE_NBR'] == store]
    plt.plot(store_data['MONTH'], store_data['TOT_SALES'], label=f"Store {store}")
plt.xticks(rotation=45)
plt.title(f"Total Sales: Store {trial_store} vs Control Store {control_store}")
plt.xlabel("Month")
plt.ylabel("Total Sales")
plt.legend()
plt.tight_layout()
plt.show()
```



Store 77 was compared to Store 233, with a strong correlation (0.93) in pre-trial sales trends. During the trial period (Feb–Aug 2019), Store 77 showed a noticeable increase in total sales, while Store 233 remained flat. The increase was mainly driven by a rise in the number of customers, not just repeat transactions. This suggests that the new layout attracted more shoppers, and we recommend rolling it out to similar stores.