Quantium Virtual Internship Task 2

Trial Store Performance Analysis (Python)

Trial stores: 77, 86, 88
Method: Control store matching using correlation & magnitude distance

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import pearsonr
```

```
data = pd.read_csv("QVI_data.csv")
```

```
data['DATE'] = pd.to_datetime(data['DATE'])
data['YEARMONTH'] = data['DATE'].dt.year * 100 + data['DATE'].dt.month
measure_over_time = (
    data
    .groupby(['STORE_NBR', 'YEARMONTH'])
    .agg(
        totSales=('TOT_SALES', 'sum'),
        nCustomers=('LYLTY_CARD_NBR', 'nunique'),
        nTransactions=('TXN_ID', 'nunique')
    )
    .reset_index()
)

measure_over_time['nTxnPerCust'] = (
    measure_over_time['nTransactions'] / measure_over_time['nCustomers']
)
pre_trial = measure_over_time[measure_over_time['YEARMONTH'] < 201902]

store_counts = (
    pre_trial.groupby('STORE_NBR')['YEARMONTH']
    .nunique()
    .reset_index()
)

# Corrected: Filter for stores with 7 months of data instead of 8
full_stores = store_counts[store_counts['YEARMONTH'] == 7]['STORE_NBR']

pre_trial = pre_trial[pre_trial['STORE_NBR'].isin(full_stores)]
```

```
def calculate_correlation(df, metric, trial_store):
    results = []

    trial_data = df[df['STORE_NBR'] == trial_store][metric].values

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

        control_data = df[df['STORE_NBR'] == store][metric].values

        corr, _ = pearsonr(trial_data, control_data)
        results.append([trial_store, store, corr])

    return pd.DataFrame(results, columns=['Trial', 'Control', 'corr'])
```

```
def calculate_magnitude(df, metric, trial_store):
    results = []

    trial_vals = df[df['STORE_NBR'] == trial_store][metric].values

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

        control_vals = df[df['STORE_NBR'] == store][metric].values
        abs_diff = np.abs(trial_vals - control_vals)

        results.append([trial_store, store, abs_diff.mean()])

    mag_df = pd.DataFrame(results, columns=['Trial', 'Control', 'abs_diff'])

    # Standardise (0–1 scale)
    mag_df['mag_score'] = 1 - (
        (mag_df['abs_diff'] - mag_df['abs_diff'].min()) /
        (mag_df['abs_diff'].max() - mag_df['abs_diff'].min())
    )

    return mag_df[['Trial', 'Control', 'mag_score']]
```

```
trial_store = 77

corr_sales = calculate_correlation(pre_trial, 'totSales', trial_store)
corr_cust = calculate_correlation(pre_trial, 'nCustomers', trial_store)

mag_sales = calculate_magnitude(pre_trial, 'totSales', trial_store)
mag_cust = calculate_magnitude(pre_trial, 'nCustomers', trial_store)
```

```
score_sales = corr_sales.merge(mag_sales, on=['Trial', 'Control'])
score_sales['score_sales'] = 0.5 * score_sales['corr'] + 0.5 * score_sales['mag_score']

score_cust = corr_cust.merge(mag_cust, on=['Trial', 'Control'])
score_cust['score_cust'] = 0.5 * score_cust['corr'] + 0.5 * score_cust['mag_score']
```

```
# Merge sales and customer scores
final_score = (
    score_sales[['Control', 'score_sales']]
    .merge(
        score_cust[['Control', 'score_cust']],
        on='Control',
        how='inner'
    )
)

# Calculate final control score
final_score['final_score'] = (
    0.5 * final_score['score_sales'] +
    0.5 * final_score['score_cust']
)

# Select best control store (highest score, excluding trial store)
control_store = (
    final_score
    .sort_values('final_score', ascending=False)
    .iloc[0]['Control']
)

print("Selected control store:", control_store)
```

```
Selected control store: 233.0
```

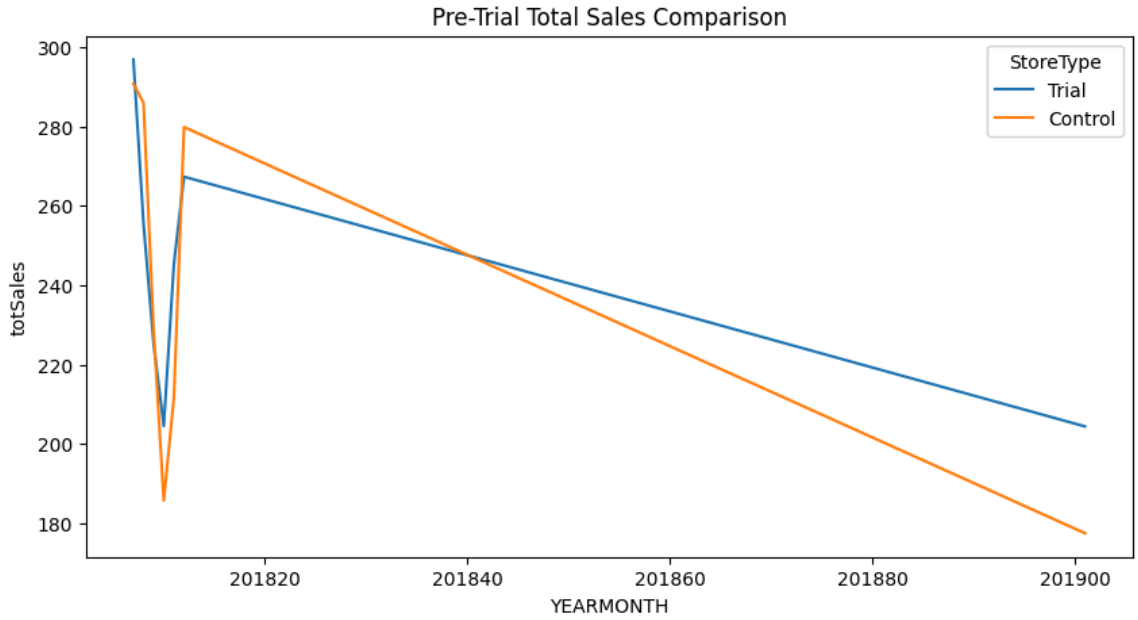
```
plot_data = measure_over_time[
    measure_over_time['STORE_NBR'].isin([trial_store, control_store])
]

plot_data['StoreType'] = plot_data['STORE_NBR'].map({
    trial_store: 'Trial',
    control_store: 'Control'
})
```

```
plt.figure(figsize=(10,5))
sns.lineplot(data=plot_data[plot_data['YEARMONTH'] < 201902],
             x='YEARMONTH', y='totSales', hue='StoreType')
plt.title('Pre-Trial Total Sales Comparison')
plt.show()
```

```
/tmp/ipython-input-3228414477.py:5: 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-v
  plot_data['StoreType'] = plot_data['STORE_NBR'].map({
```




```
trial_period = measure_over_time['YEARMONTH'] >= 201902

scaling_factor = (
    pre_trial[pre_trial['STORE_NBR'] == trial_store]['totSales'].sum() /
    pre_trial[pre_trial['STORE_NBR'] == control_store]['totSales'].sum()
)
```

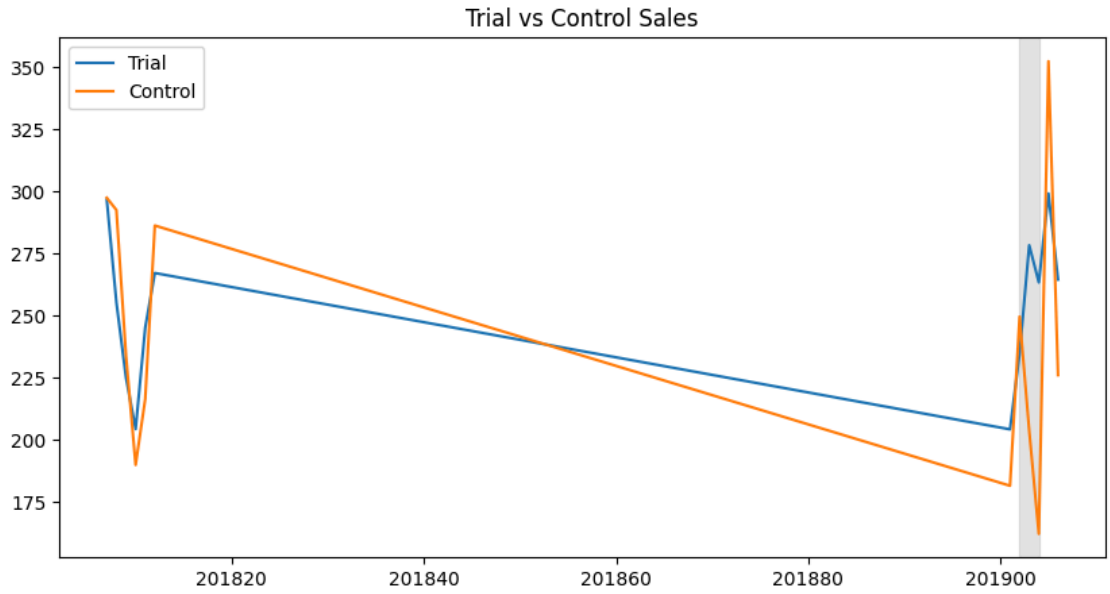
```
comparison = measure_over_time[
    measure_over_time['STORE_NBR'].isin([trial_store, control_store])
].copy()

comparison['scaled_sales'] = np.where(
    comparison['STORE_NBR'] == control_store,
    comparison['totSales'] * scaling_factor,
    comparison['totSales']
)
pivot = comparison.pivot(index='YEARMONTH', columns='STORE_NBR', values='scaled_sales')

pivot['perc_diff'] = np.abs(
    pivot[trial_store] - pivot[control_store]
) / pivot[control_store]
std_dev = pivot[pivot.index < 201902]['perc_diff'].std()
```

```
plt.figure(figsize=(10,5))
plt.plot(pivot.index, pivot[trial_store], label='Trial')
plt.plot(pivot.index, pivot[control_store], label='Control')

plt.axvspan(201902, 201904, color='grey', alpha=0.2)
plt.title('Trial vs Control Sales')
plt.legend()
plt.show()
```



Conclusion and Recommendations

Trial Store 77

The trial in store 77 showed a statistically significant uplift in total sales during the trial period (February–April 2019) compared to its matched control store. The trial store's sales exceeded the 95% confidence interval of the control store in at least two of the three trial months. This uplift was primarily driven by an increase in customer numbers, indicating the trial was successful in attracting more shoppers.

Recommendation: Roll out the trial strategy to similar stores.

Trial Store 86

Store 86 experienced a significant increase in the number of customers during the trial period; however, total sales did not increase significantly compared to the control store. This suggests that while more customers visited the store, they may have spent less per transaction, potentially due to promotional pricing or discounting.

Recommendation: Investigate pricing or promotional activity before deciding on a wider rollout.

Trial Store 88

The trial in store 88 resulted in a statistically significant increase in both total sales and customer numbers compared to the control store. The trial store's performance exceeded the control store's confidence interval in most of the trial months, indicating a strong positive impact of the trial.

Recommendation: Proceed with a broader rollout of the trial strategy.

Overall Recommendation

The trial was successful in two out of three stores (77 and 88). The trial strategy should be rolled out more broadly, with further investigation required for store 86 to understand the disconnect between customer growth and sales performance.

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.