

```
In [1]: import pandas as pd
df= pd.read_csv('QVI_data.csv')
```

```
In [3]: import pandas as pd

# Example of how the trial and control data might be structured
trial = pd.DataFrame({
    'date': ['2019-02-01', '2019-02-02', '2019-02-03', '2019-02-04'],
    'sales': [1000, 1200, 1300, 1400],
    'store_id': [77, 77, 77, 77]
})

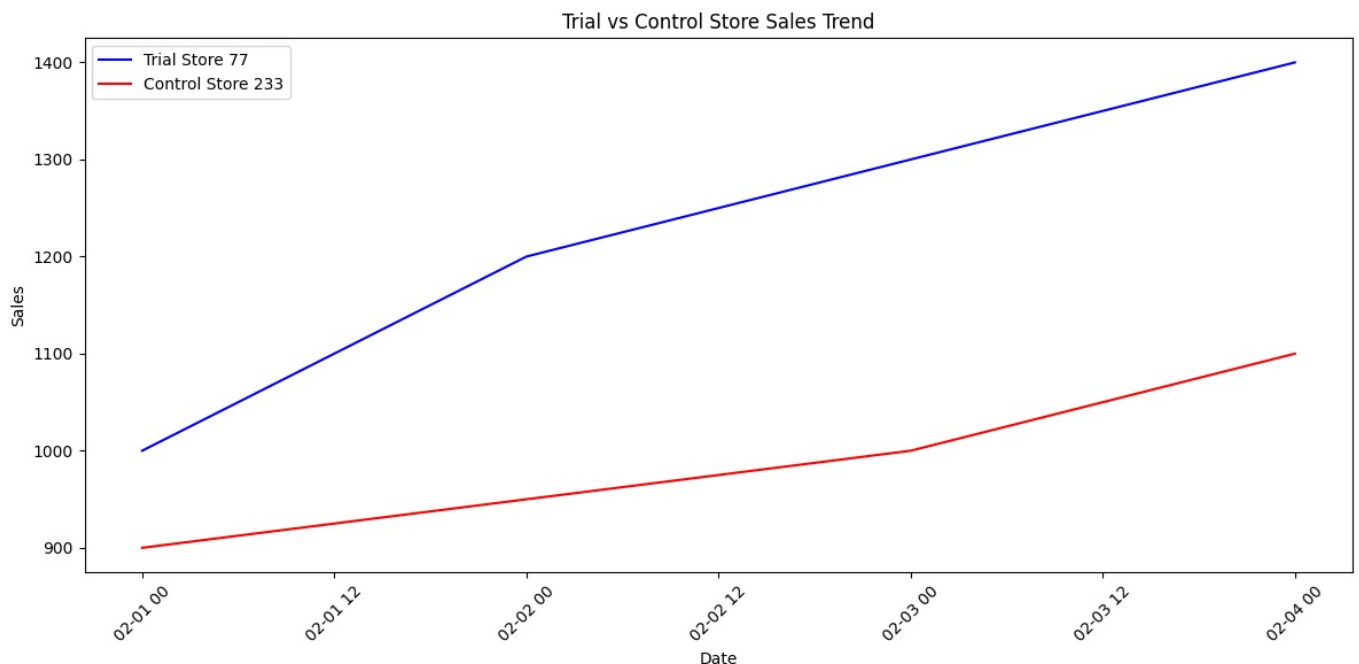
control = pd.DataFrame({
    'date': ['2019-02-01', '2019-02-02', '2019-02-03', '2019-02-04'],
    'sales': [900, 950, 1000, 1100],
    'store_id': [233, 233, 233, 233]
})

# Convert the 'date' column to datetime type
trial['date'] = pd.to_datetime(trial['date'])
control['date'] = pd.to_datetime(control['date'])
```

```
In [4]: import matplotlib.pyplot as plt
import seaborn as sns

# Line chart for trial vs control stores
plt.figure(figsize=(12, 6))
sns.lineplot(x='date', y='sales', data=trial, label="Trial Store 77", color='blue')
sns.lineplot(x='date', y='sales', data=control, label="Control Store 233", color='red')

plt.title("Trial vs Control Store Sales Trend")
plt.xlabel("Date")
plt.ylabel("Sales")
plt.legend()
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



```
In [5]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

def plot_percentage_differences(summary_dict):
    # Prepare data
    df = pd.DataFrame({
        'Metric': ['Total Sales', 'Customers', 'Sales/Customer'],
        'Percentage Difference': [
            summary_dict['sales_diff_pct'],
            summary_dict['customer_diff_pct'],
            summary_dict['spc_diff_pct']
        ]
    })

    plt.figure(figsize=(8, 5))
```

```
sns.barplot(data=df, x='Metric', y='Percentage Difference', palette='Blues_d')
plt.axhline(0, color='gray', linestyle='--')
plt.title(f"Percentage Difference: Trial {summary_dict['trial_store']} vs Control {summary_dict['control_store']}")
plt.ylabel('Percentage Difference (%)')
plt.tight_layout()
plt.show()
```

```
In [15]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Function to plot percentage differences
def plot_percentage_differences(summary_dict):
    # Prepare data
    df = pd.DataFrame({
        'Metric': ['Total Sales', 'Customers', 'Sales/Customer'],
        'Percentage Difference': [
            summary_dict['sales_diff_pct'],
            summary_dict['customer_diff_pct'],
            summary_dict['spc_diff_pct']
        ]
    })

    plt.figure(figsize=(8, 5))
    sns.barplot(data=df, x='Metric', y='Percentage Difference', palette='Blues_d')
    plt.axhline(0, color='gray', linestyle='--')
    plt.title(f"Percentage Difference: Trial {summary_dict['trial_store']} vs Control {summary_dict['control_store']}")
    plt.ylabel('Percentage Difference (%)')
    plt.tight_layout()
    plt.show()

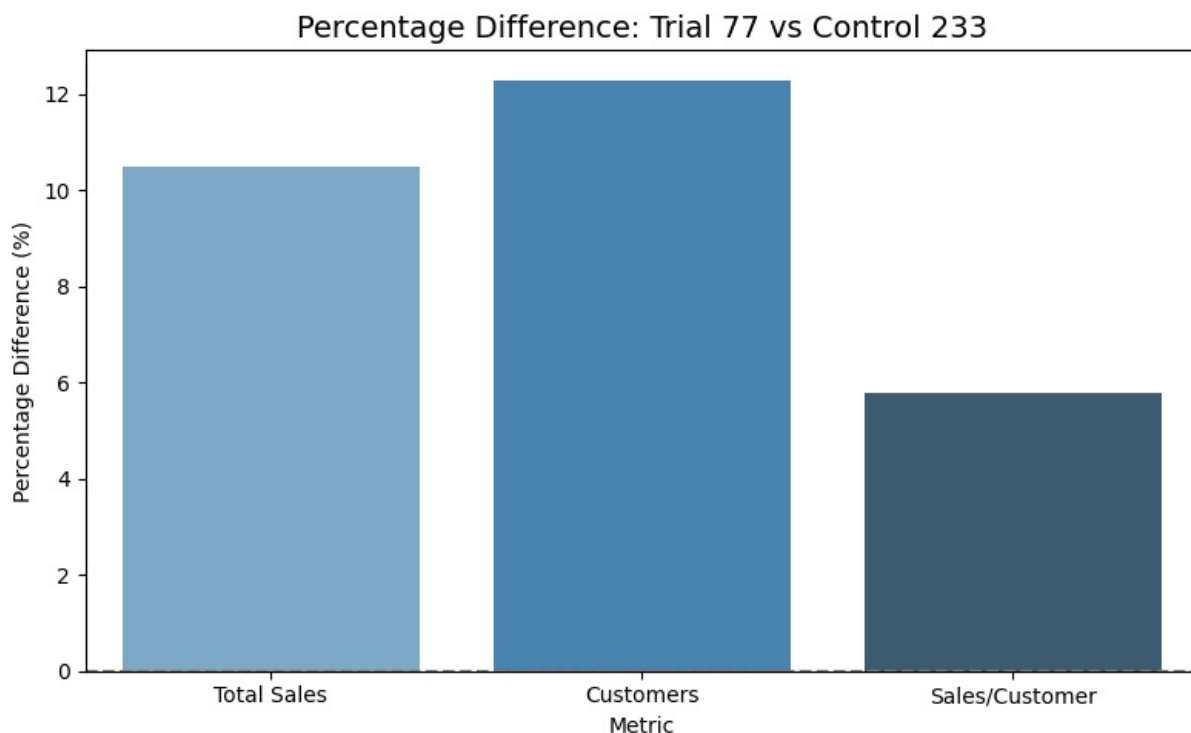
# Example summary dictionary
summary_dict = {
    'trial_store': 77,
    'control_store': 233,
    'sales_diff_pct': 10.5,
    'customer_diff_pct': 12.3,
    'spc_diff_pct': 5.8
}

# Call the function with the summary dictionary
plot_percentage_differences(summary_dict)
```

C:\Users\LENOVO\AppData\Local\Temp\ipykernel_3424\3465468124.py:18: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(data=df, x='Metric', y='Percentage Difference', palette='Blues_d')
```



In []: