

Python Summer Party Challenge

by Interview Master

Day 14 of 15

Starbucks

You are a Business Analyst on the Starbucks Rewards team investigating customer transaction behavior. Your team wants to understand how loyalty program membership influences purchasing patterns. The goal is to compare transaction metrics between loyalty members and non-members.

Challenge Questions

Q1:

For the month of July 2024, how many transactions did loyalty program members and non-members make? Compare the transaction counts between these two groups.

Q2:

What is the average transaction value for loyalty program members and non-members during July 2024? Use this to identify which group has a higher average transaction value.

Q3:

Determine the percentage difference in average transaction value between loyalty program members and non-members for July 2024.



Want to try this yourself?

Join the Challenge

Sign up for the Python Summer Party Challenge and solve 21 days
of data science problems

www.interviewmaster.ai/python-party

Or keep scrolling to see my solutions

My Solution - Q1

Day 14 Python Challenge

```
# Merging fct_transactions and dim_customers
merged_data = pd.merge(
    fct_transactions, dim_customers,
    on='customer_id',
    how='inner'
)

# Filtering for July 2024 transactions
july_24 = merged_data[
    (merged_data['transaction_date'].dt.year == 2024) &
    (merged_data['transaction_date'].dt.month == 7)
]

# Counting transactions by loyalty membership
transaction_counts = july_24.groupby("is_loyalty_member")["transaction_id"].count().reset_index()
transaction_counts.columns = ["is_loyalty_member", "transaction_count"]

print(transaction_counts)
```



My Solution - Q2

Day 14 Python Challenge

```
# Merging fct_transactions and dim_customers
merged_data = pd.merge(
    fct_transactions, dim_customers,
    on='customer_id',
    how='inner'
)

# Filtering for July 2024 transactions
july_24 = merged_data[
    (merged_data['transaction_date'].dt.year == 2024) &
    (merged_data['transaction_date'].dt.month == 7)
]

# Calculating average transaction value by loyalty membership
transaction_avg = (
    july_24.groupby("is_loyalty_member")
    .agg(avg_transaction_value= ("transaction_value", "mean"))
    .reset_index()
)
print(transaction_avg)
```



My Solution - Q3

Day 14 Python Challenge

```
# Merging fct_transactions and dim_customers
merged_data = pd.merge(
    fct_transactions, dim_customers,
    on='customer_id',
    how='inner'
)

# Filtering for July 2024 transactions
july_24 = merged_data[
    (merged_data['transaction_date'].dt.year == 2024) &
    (merged_data['transaction_date'].dt.month == 7)
]

# Calculating average transaction value by loyalty membership
transaction_avg = (
    july_24.groupby("is_loyalty_member")
    .agg(avg_transaction_value=("transaction_value", "mean"))
    .reset_index()
)
print(transaction_avg)

# Extracting the average values
avg_member = transaction_avg.loc[transaction_avg["is_loyalty_member"] == True, "avg_transaction_value"].values[0]
avg_non_member = transaction_avg.loc[transaction_avg["is_loyalty_member"] == False, "avg_transaction_value"].values[0]

# Calculating percentage difference
percentage_diff = ((avg_member - avg_non_member) / avg_non_member) * 100
print(f"\nPercentage difference in average transaction value\n(Members vs Non-members): {percentage_diff:.2f}%")
```

Ready for your own challenge?

Try this yourself by signing up for the Python Summer Party challenge:

www.interviewmaster.ai/python-party

