

# Python Summer Party Challenge

by Interview Master

Day 3 of 15

## Disney

You are a data analyst working with the Disney Parks revenue team to understand nuanced guest spending patterns across different park experiences. The team wants to develop a comprehensive view of visitor purchasing behaviors. Your goal is to uncover meaningful insights that can drive personalized marketing strategies.

### Challenge Questions

**Q1:**

What is the average spending per guest per visit for each park experience type during July 2024? Ensure that park experience types with no recorded transactions are shown with an average spending of 0.0. This analysis helps establish baseline spending differences essential for later segmentation.

**Q2:**

For guests who visited our parks more than once in August 2024, what is the difference in spending between their first and their last visit? This investigation, using sequential analysis, will reveal any shifts in guest spending behavior over multiple visits.

**Q3:**

In September 2024, how can guests be categorized into distinct spending segments such as Low, Medium, and High based on their total spending? Use the following thresholds for categorization:  
-Low: Includes values from \$0 up to, but not including, \$50.  
-Medium: Includes values from \$50 up to, but not including, \$100.  
-High: Includes values from \$100 and above.  
Exclude guests who did not make any purchases in the period.



# 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](http://www.interviewmaster.ai/python-party)

Or keep scrolling to see my solutions

# My Solution - Q1

Day 3 Python Challenge

```
# Unique park experience types
unique_experience_types = fct_guest_spending[['park_experience_type']].drop_duplicates()

# Get park experience in July 2024 alone
fct_guest_spending['visit_date'] = pd.to_datetime(fct_guest_spending['visit_date'])
july_24_visits = fct_guest_spending[
    (fct_guest_spending['visit_date'].dt.month == 7) &
    (fct_guest_spending['visit_date'].dt.year == 2024)
]

# Calculate the total spend and number of visits for each experience type
experience_spend = (
    july_24_visits.groupby('park_experience_type')
        .agg(total_spent=('amount_spent', 'sum'), total_visits=('guest_id', 'count'))
        .reset_index()
)

# Calculate the average spend for each experience type
experience_spend['avg_spending'] = experience_spend['total_spent'] / experience_spend['total_visits']

# Merge with the unique experience types to include the ones with no recorded transactions
avg_spend = pd.merge(
    unique_experience_types,
    experience_spend[['park_experience_type', 'avg_spending']],
    on='park_experience_type',
    how='left'
)

# Fill missing values with 0.0
avg_spend['avg_spending'] = avg_spend['avg_spending'].fillna(0.0)
print(avg_spend)
```



# My Solution - Q2

Day 3 Python Challenge

```
# Get data for August 2024 visits alone
fct_guest_spending['visit_date'] = pd.to_datetime(fct_guest_spending['visit_date'])
august_24_visits = fct_guest_spending[
    (fct_guest_spending['visit_date'].dt.month == 8) &
    (fct_guest_spending['visit_date'].dt.year == 2024)
]

# Identify guests with multiple visits
august_24_visits['visit_count'] = august_24_visits.groupby('guest_id')['visit_date'].transform('count')
multiple_visits = august_24_visits[august_24_visits['visit_count'] > 1]

# Determine the first and last visit and spending for each guest
first_last_spend = (
    multiple_visits.sort_values('visit_date')
    .groupby('guest_id').agg(
        first_spending=('amount_spent', 'first'),
        last_spending=('amount_spent', 'last')
    )
    .reset_index()
)

# Calculate the spending difference between the first and last visit
first_last_spend['spending_difference'] = first_last_spend['last_spending'] - first_last_spend['first_spending']
print(first_last_spend[['guest_id', 'spending_difference']])
```



# My Solution - Q3

Day 3 Python Challenge

```
# Get 'guests' visits for September 2024 alone
fct_guest_spending['visit_date'] = pd.to_datetime(fct_guest_spending['visit_date'])
sept_24_visits = fct_guest_spending[
    (fct_guest_spending['visit_date'].dt.month == 9) &
    (fct_guest_spending['visit_date'].dt.year == 2024)
]

# Group guests to get their total spend and exclude guests without purchase
guest_spend = sept_24_visits.groupby('guest_id', as_index=False)[['amount_spent']].sum()
guest_spend = guest_spend[guest_spend['amount_spent'] > 0]

# Categorize guest based on their total spend
def categorize_spending(amount):
    if amount < 50:
        return 'Low'
    elif amount < 100:
        return 'Medium'
    else:
        return 'High'

guest_spend['spending_segment'] = guest_spend['amount_spent'].apply(categorize_spending)
print(guest_spend)
```

Ready for your own challenge?

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

[www.interviewmaster.ai/python-party](http://www.interviewmaster.ai/python-party)

