

# Python Summer Party Challenge

by Interview Master

Day 15 of 15

## Uber

You are a Business Analyst on the Uber Pool Product Team working to optimize driver compensation. The team aims to understand how trip characteristics impact driver earnings. Your goal is to develop data-driven recommendations that maximize driver earnings potential.

### Challenge Questions

**Q1:**

What is the average driver earnings per completed UberPool ride with more than two riders between July 1st and September 30th, 2024? This analysis will help isolate trips that meet specific rider thresholds to understand their impact on driver earnings.

**Q2:**

For completed UberPool rides between July 1st and September 30th, 2024, derive a new column calculating earnings per mile (total\_earnings divided by total\_distance) and then compute the average earnings per mile for rides with more than two riders. This calculation will reveal efficiency metrics for driver compensation.

**Q3:**

Identify the combination of rider count and total distance that results in the highest average driver earnings per UberPool ride between July 1st and September 30th, 2024. This analysis directly recommends optimal trip combination strategies to maximize driver earnings.



# 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 15 Python Challenge

```
# Filtering the data to meet the conditions
filtered_trips = fct_trips[
    (fct_trips['ride_type'] == 'UberPool') &
    (fct_trips['rider_count'] > 2) &
    (fct_trips['trip_date'].dt.year == 2024) &
    (fct_trips['trip_date'].dt.month.isin([7,8,9]))]
]

# Calculating driver average earnings
avg_earnings = filtered_trips['total_earnings'].mean()
print(f"The average driver earnings per completed UberPool ride with more than two riders between July 1st and September 30th, 2024 is: ${avg_earnings}")
```



# My Solution - Q2

Day 15 Python Challenge

```
# Filtering the data to meet the conditions
filtered_trips = fct_trips[
    (fct_trips['ride_type'] == 'UberPool') &
    (fct_trips['trip_date'].dt.year == 2024) &
    (fct_trips['trip_date'].dt.month.isin([7,8,9]))]
]

# Adding the earnings per mile column
filtered_trips['earnings_per_mile'] = filtered_trips['total_earnings'] / filtered_trips['total_distance']

# Filtering for rides with more than two riders
filtered_trips = filtered_trips[filtered_trips['rider_count'] > 2]

# Calculating average earnings per mile
avg_earnings_per_mile = filtered_trips['earnings_per_mile'].mean()

print(f"The average earnings per mile for completed UberPool ride with more than two riders between July 1st and September 30th, 2024 is: ${avg_earnings_per_mile:.2f}")
```



# My Solution - Q3

Day 15 Python Challenge

```
# Filtering the data to meet the conditions
filtered_trips = fct_trips[
    (fct_trips['ride_type'] == 'UberPool') &
    (fct_trips['trip_date'].dt.year == 2024) &
    (fct_trips['trip_date'].dt.month.isin([7,8,9]))]
]

# Create a unique combination of rider count and trip distance
# and calculate average total_earnings
unique_combinations_earnings = (
    filtered_trips.groupby(['rider_count','total_distance'])
    .agg(avg_earnings=('total_earnings','mean'))
    .reset_index()
)

# Find the top combination
top_combination = unique_combinations_earnings.nlargest(1, 'av
g_earnings')

print("Optimal rider count and distance combination for maximi
zing driver earnings:")
print(top_combination)
```

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)



InterviewMaster.AI