Click anywhere to navigate to website

Python Summer Party

by **Interview Master** & in partnership with **DataCamp**

15 days of *NumPy & Pandas* coding challenges

The challenge has started!

You can still sign up to join at any time.

Join **3,651** other people on this challenge!

Go to Python Party

Data Cleaning in Python Cheatsheet

Step 0: Import libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
```

Step 1: Understand Data Structure

```
df.info()
df.describe()
df.head (10)
```

Step 2: Explore the Data

For numerical columns

```
df['numeric_column'].min()
df['numeric_column'].max()
df['numeric_column'].mean()
df['numeric_column'].std()
```

For categorical columns

```
df['categorical_column'].value_counts()
```

Plot distributions

```
sns.histplot(df['numeric_column'])
sns.displot(df['numeric_column'], kde=True)
sns.countplot(x='categorical_column', data=df)
```

Step 3: Standardize Data Formats

Convert to lower/upper case

```
df['column_name'] = df['column_name'].str.lower()
df['column_name'] = df['column_name'].str.upper()
```

Format dates

```
df['column_name'] = pd.to_datetime(df['column_name'],
                    format='%Y-%m-%d')
```

Remove leading/trailing spaces

```
df['column_name'] = df['column_name'].str.strip()
```

Convert data types

```
df['string_col'] = df['string_col'].astype('category')
df['num_col'] = df['num_col'].astype('float')
```

Step 4: Remove Duplicates

```
df = df.drop_duplicates()
```

Step 5: Handle Missing Values

Fill with 0 (when 0 is meaningful)

```
df['column_name'] = df['column_name'].fillna(0)
```

Drop rows if missing data is critical

```
df['numeric_column'] = df['numeric_column']
fillna(df['numeric_column'].mean())
```

Fill with average (numerical columns only)

```
df = df[df['column_name'].notna()]
```

Step 6: Standardize String Values

Using replace

```
df['column_name_cleaned'] = df['column_name'].
                        .replace({'val1':'standard_val',
                        'Val 1': 'standard_val',
                        'VAL1': 'standard_val'})
```

Or using numpy:

```
df['column_name_cleaned'] = np.where(
     df['column_name'].isin(['val1','Val1','VAL1']),
     'standard_val', df['column_name'])
```

Step 7: Filter Out Bad Data

Remove rows with clearly bad values (e.g. negative sales)

```
df = df[df['sales'] >= 0]
```

Drop columns with too many nulls (e.g. more than 50%)

```
threshold = len(df) * 0.5
df = df.dropna(axis=1, thresh=threshold)
```

Step 8: Remove Outliers

Use IQR method to filter outliers

```
Q1 = df['numeric_column'].quantile(0.25)
Q3 = df['numeric_column'].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

df = df[(df['numeric_column'] >= lower_bound &
(df['numeric_column'] <= upper_bound)]
```

Step 9: Rename Columns

```
df = df.rename(columns={
'column_name_1': 'clean_column_name_1',
'column_name_2': 'clean_column_name_2'
})
```

Step 10: Save Cleaned Data

Export to csv

```
df.to_csv('cleaned_data.csv', index=False)
```

Save as a new dataframe

```
cleaned_df = df.copy()
```

Dawn Choo

SQL to Python

Quickstart Guide

	SQL	Python
Filtering	`SELECT * FROM table WHERE column = 'value'`	`df[df['column'] == 'value']`
Ordering	`SELECT * FROM table ORDER BY column ASC`	`df.sort_values(by='column', ascending=True)`
Removing duplicates	`SELECT DISTINCT col1, col2 FROM table`	`df.drop_duplicates (subset=['col1','col2'])`
Filling missing values	`SELECT COALESCE(col, 'xxx') FROM table`	`df['column'].fillna('xxx')`
Changing data types	`SELECT CAST(col AS INTEGER) FROM table`	`df['column'].astype(int)`
Renaming columns	`SELECT col AS new_col FROM table`	`df['column'].rename. (columns={'col':'new_col'})`
Summing	`SELECT SUM(column) FROM table`	`df['column'].sum()`
Averaging	`SELECT AVG(column) FROM table`	`df['column'].mean()`
Minimum / maximums	`SELECT MIN(column) FROM table`	`df['column'].min()`
Counting	`SELECT COUNT(column) FROM table`	`df['column'].count()`
Percentiles	`SELECT PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY col) FROM table`	`df['column'].quantile(0.5)`
Aggregating by groups	`SELECT group_column, AVG(col) FROM table GROUP BY group_column`	`df.groupby('group_column'). ['col'].mean()`
Merging datasets	`SELECT * FROM table1 JOIN table2 ON table1.key = table2.key`	`pd.merge(table1, table2, on='key')`
Appending datasets	`SELECT * FROM table1 UNION ALL SELECT * FROM table2`	`pd.concat([table1, table2])`

Dawn Choo

Get 200+ SQL & Python practice questions at www.InterviewMaster.ai

Frequently Asked Questions 

What is the Python Summer Party?

It's a 15-day coding challenge designed to level up your Python data analysis skills! Each day, you'll get a new coding challenge focusing on NumPy and Pandas. Every question is based on a real company so that you're learning real-world skills.

How much Python experience is needed?

You just need to know Python basics to get started! The challenges range from beginner to intermediate level. Plus, you have the AI to help you out if you get stuck. We encourage you to join no matter your current skill level!

Frequently Asked Questions 

When do new challenges become available?

A new challenge becomes available each day at 8AM ET starting August 1st. You can complete each challenge as many times as you want and take your time to really understand the concepts.

Do I have to pay to join?

No! This is a free event. We want you to get the most out of it by learning as much as possible.

Frequently Asked Questions 

What happens if I miss a day of the challenge?

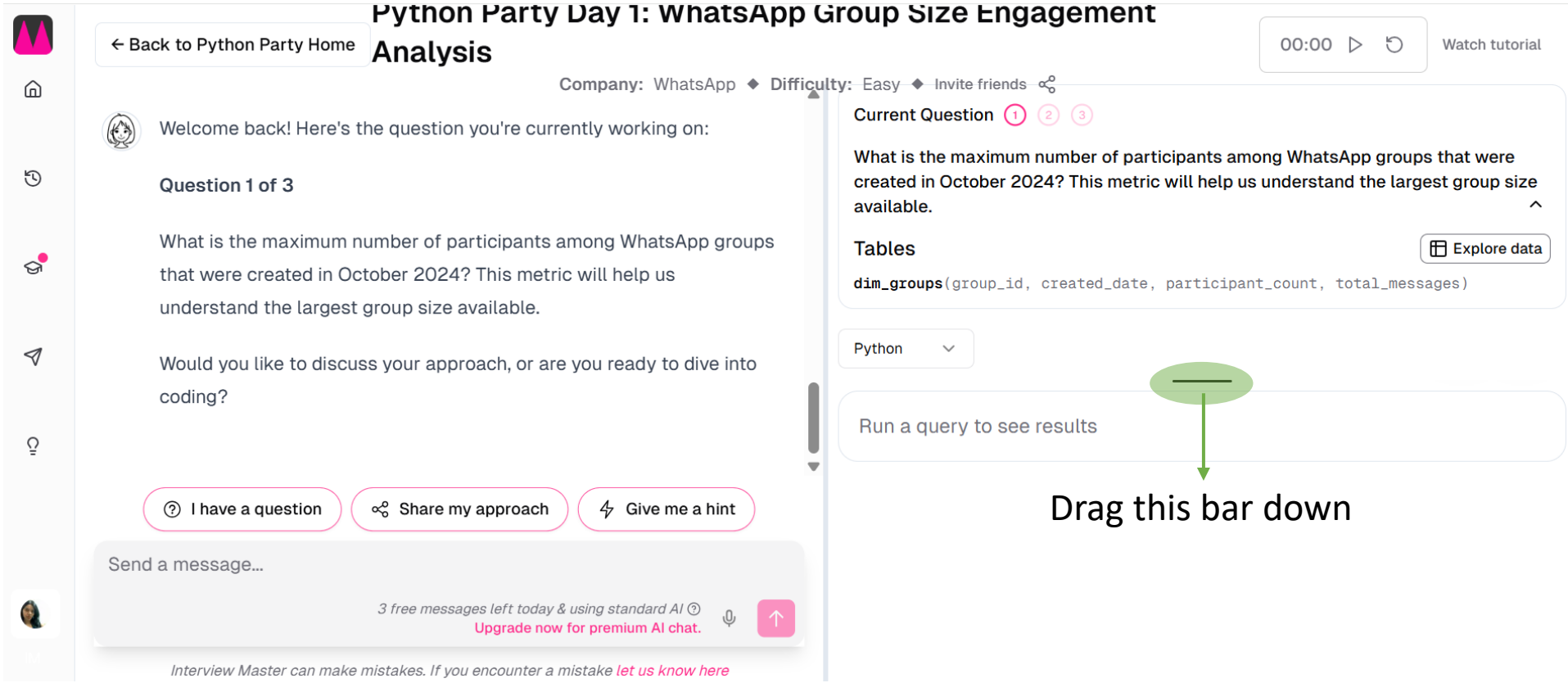
No worries! We understand that life happens and we all have busy schedules. You can always catch up on the challenges you missed. The challenge will be available until August 31st.

Will I get a certificate of completion?

Yes! You'll get a certificate of completion if you complete all 15 challenges within the challenge period (August 1st - August 31st).

Frequently Asked Questions

Cannot see Code Cell?



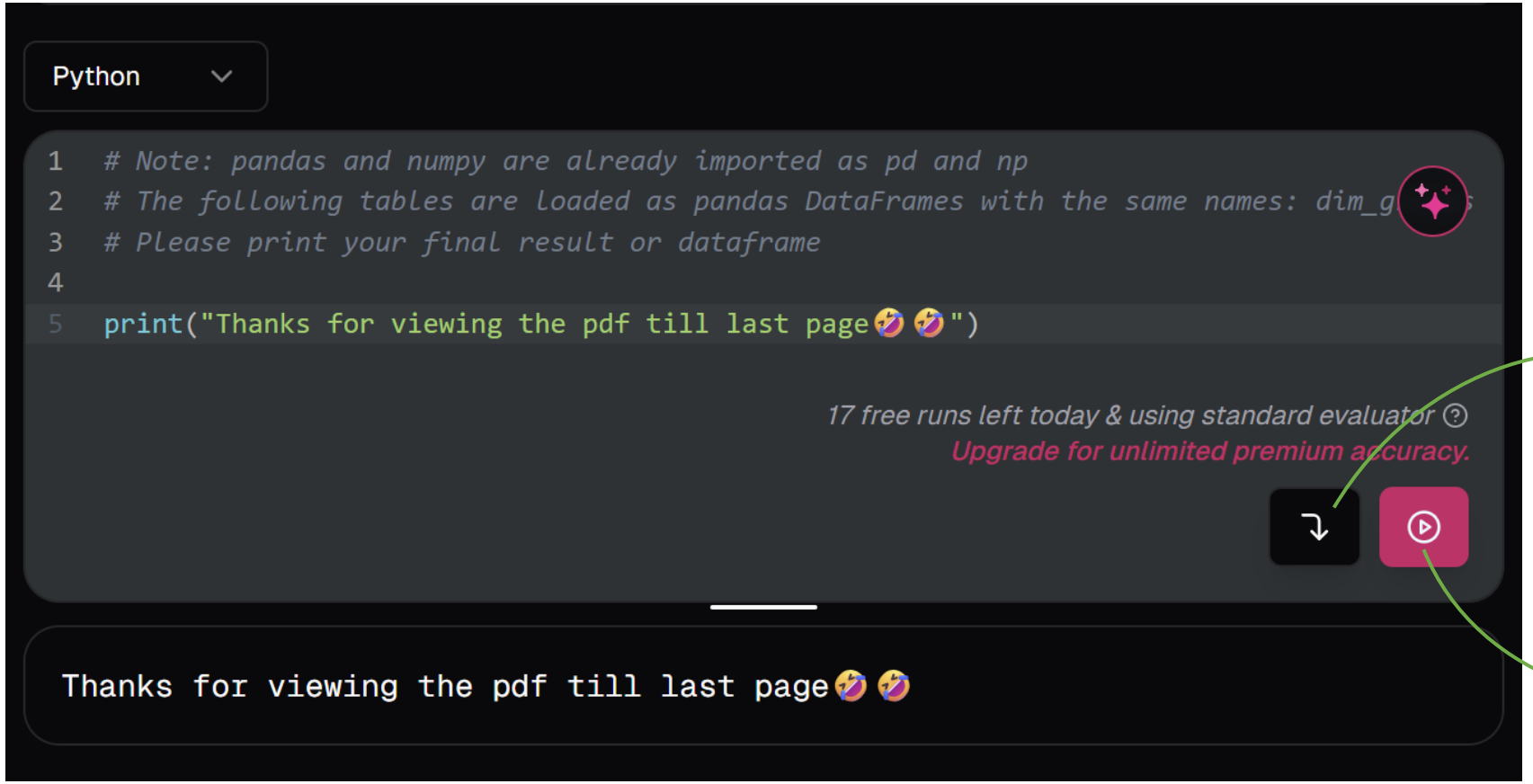
Frequently Asked Questions 

What questions we can ask AI assistant to help us?

1. Can you please help me understand question in better?
2. Can you help me structure a solution for this question?
3. How joins / merge / groupby / aggregate function work in Pandas?
4. How to merge these two DataFrames using (product_id) column?
5. How do I calculate new column based on existing ones?
6. What columns we should have in output?
7. How to calculate (CTR, Percentage Difference)?
8. Why am I getting this error in my pandas code?
9. Can you help me identify error in my code?
10. Can you please explain me this code?

Frequently Asked Questions

20 Submit Limit? Run Is Your Friend!



💡 Break down the problem into small parts and validate your results after each step by running the code.

Run query without submission (unlimited)

Submit + Evaluate (limited to 20)