

Gen AI for Coding

Using Microsoft & GitHub Copilot

> February 26, 2025

Background

- Scientific computing / Data focus
- Still constantly changing
- We don't have all the answers!
- **Please contribute! Ideas and questions are welcome**

Today

- Discuss Microsoft and GitHub Copilot
- What can Copilot do with code?
- When to be hesitant?
- Live examples
- Resources and closing thoughts

Have you already used Gen AI for
coding?

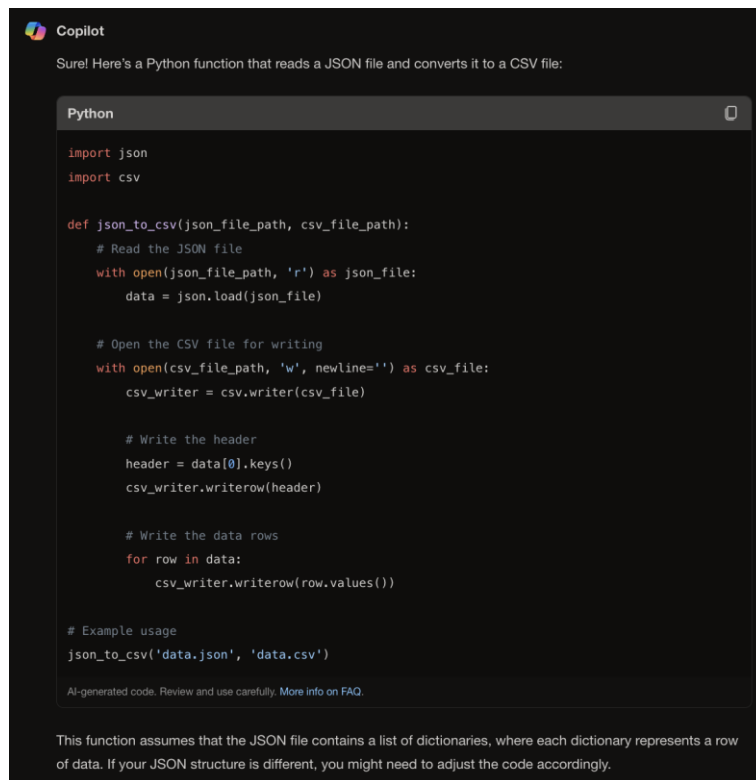
What do you use?

Please share in the chat!



Gen AI tools for coding

Microsoft Copilot



Sure! Here's a Python function that reads a JSON file and converts it to a CSV file:

```
Python

import json
import csv

def json_to_csv(json_file_path, csv_file_path):
    # Read the JSON file
    with open(json_file_path, 'r') as json_file:
        data = json.load(json_file)

    # Open the CSV file for writing
    with open(csv_file_path, 'w', newline='') as csv_file:
        csv_writer = csv.writer(csv_file)

        # Write the header
        header = data[0].keys()
        csv_writer.writerow(header)

        # Write the data rows
        for row in data:
            csv_writer.writerow(row.values())

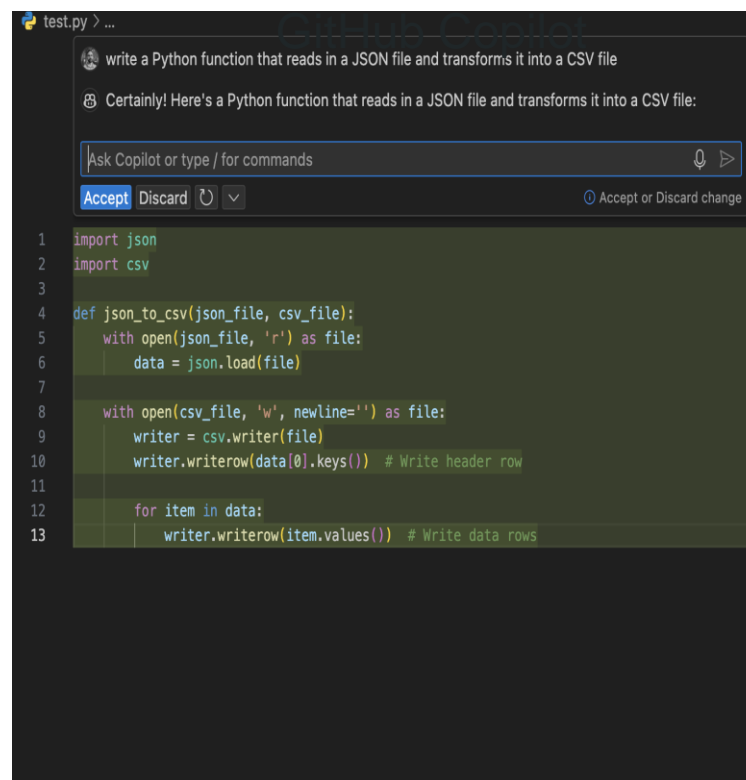
# Example usage
json_to_csv('data.json', 'data.csv')
```

AI-generated code. Review and use carefully. More info on FAQ.

This function assumes that the JSON file contains a list of dictionaries, where each dictionary represents a row of data. If your JSON structure is different, you might need to adjust the code accordingly.

<https://copilot.cloud.microsoft>

GitHub Copilot



write a Python function that reads in a JSON file and transforms it into a CSV file

Certainly! Here's a Python function that reads in a JSON file and transforms it into a CSV file:

Ask Copilot or type / for commands

Accept Discard ⌵

Accept or Discard change

```
test.py > ...

1 import json
2 import csv
3
4 def json_to_csv(json_file, csv_file):
5     with open(json_file, 'r') as file:
6         data = json.load(file)
7
8     with open(csv_file, 'w', newline='') as file:
9         writer = csv.writer(file)
10        writer.writerow(data[0].keys()) # Write header row
11
12    for item in data:
13        writer.writerow(item.values()) # Write data rows
```

<https://github.com/copilot>

Other tools



GitHub Education

- [GitHub Education for Students](#)
- [GitHub Education for Instructors and Researchers](#)
- Provides access to
 - GitHub Copilot Pro
 - GitHub Codespaces
 - Private repositories

Microsoft Copilot vs. GitHub Copilot

	Pro	Con
Microsoft Copilot	<ul style="list-style-type: none">• Already using it (?)• UNC access• Security	<ul style="list-style-type: none">• Not code focused• Underlying models not clear
GitHub Copilot	<ul style="list-style-type: none">• Integration with IDEs• Focus on code<ul style="list-style-type: none">• Completions!• Comparison between models	<ul style="list-style-type: none">• Not locked down• Limited without GitHub Education• Free tier lacks o1, Claude 3.7

What can Copilot do?



Generate code

(and sometimes execute it)

```
# create dataframe of protest data csv
df = pd.read_csv('protest_data.csv', encoding='utf-8')
```

```
def convert_dtypes_to_string(df):
```



write a function that turns all the datatypes of this dataframe to strings

Ask Copilot



GPT-4o ▾



Accept

Accept & Run



```
    return df.astype(str)
```

```
# Convert all datatypes of the dataframe to strings
df = convert_dtypes_to_string(df)
```


```
# see the columns of the dataframe
df.columns
```

Explain code

```
import json
import csv
```



 /explain

 > Used 11 references



The selected code is a function named `json_to_csv` that takes two parameters: `json_file` and `csv_file`. This function is responsible for converting a JSON file to a CSV file. Let's break down the code step by step to understand how it works.

First, the code opens the JSON file using the `open` function. The `open` function is a built-in Python function that is used to open files. It takes the file path and the mode in which the file should be opened. In this case, the mode is set to `'r'`, which stands for read mode. The `with` statement is used to ensure that the file is properly closed after it is used. The contents of the file are then loaded into the `data` variable using the `json.load` function.

Next, the code opens the CSV file using the `open` function again, but this time in write mode

Ask Copilot or type / for commands



Close View in Chat  

```
def json_to_csv(json_file, csv_file):
    with open(json_file, 'r') as file:
        data = json.load(file)

    with open(csv_file, 'w', newline='') as file:
        writer = csv.writer(file)
        writer.writerow(data[0].keys()) # Write header row

        for item in data:
            writer.writerow(item.values()) # Write data rows
```

Refactor code

```
def calculate_average_protests(df):  
    average_protests = {}  
    countries = df['country'].unique()  
    for country in countries:  
        country_df = df[df['country'] == country]  
        years = country_df['year'].unique()  
        total_protests = 0  
  
        for year in years:  
            year_df = country_df[country_df['year'] == year]  
            num_protests = year_df['protestnumber'].sum()  
            total_protests += num_protests  
        average_protests[country] = total_protests / len(years)  
    return average_protests
```


```
def calculate_average_protests(df):
```

make the function in the previous cell more concise

```
    return df.groupby('country')['protestnumber'].mean().to_dict()
```

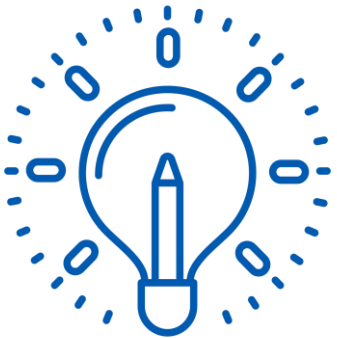

Code completion

(GitHub Copilot)

```
def calculate_average_protests(df):  
    average_protests = {}  
    countries = df['country'].unique()  
    for country in countries:  
        country_df = df[df['country'] == country]  
        years = country_df['year'].unique()  
        total_protests = 0  
  
        for year in years:  
            year_df = country_df[country_df['year'] == year]  
             number_of_protests = len(year_df)  
            total_protests += number_of_protests  
    return average_protests
```

When is Copilot useful?

High-level Goals



Short & Specific Instructions



Generating Sample Data



Learning



What to be aware of

Bad Habits



Privacy



Code Quality



When to be hesitant

- Languages or packages with less training data
- Can't test code elsewhere
- Data is being used to train the model
- Inconsistent existing code base
- Poor IDE integration

When to not use Copilot

- Sensitive data
 - **Sometimes** okay with Microsoft Copilot at UNC (<https://its.unc.edu/ai/copilot/>)
- Programming language or methods you are not familiar with
- Crawling webpages

General prompt tips

- Provide context
- Be as clear as possible
 - Finetune prompts *or* code?
- Know your intent
- Test outside

Installing GitHub Copilot

- [VS Code](#)
- [R Studio](#)
- [PyCharm](#)
- How to disable Copilot

Examples using Gen AI



User types and strategies

- Learning
- Understanding team member code
 - Access to expert help?
- Experienced coder
- What's your next best alternative to using AI?
 - What's your loss function?

Ethical considerations

Environmental Impact



Copyright & training data



Accessibility



Resources

- [How to use GitHub Copilot](#)
- [One Useful Thing \(Blog\)](#)
- [AI tools at Carolina](#)

Questions?

- Questions or ideas to try out live
- Ideas from your work
- Problems or concerns?
- What should we cover next time?