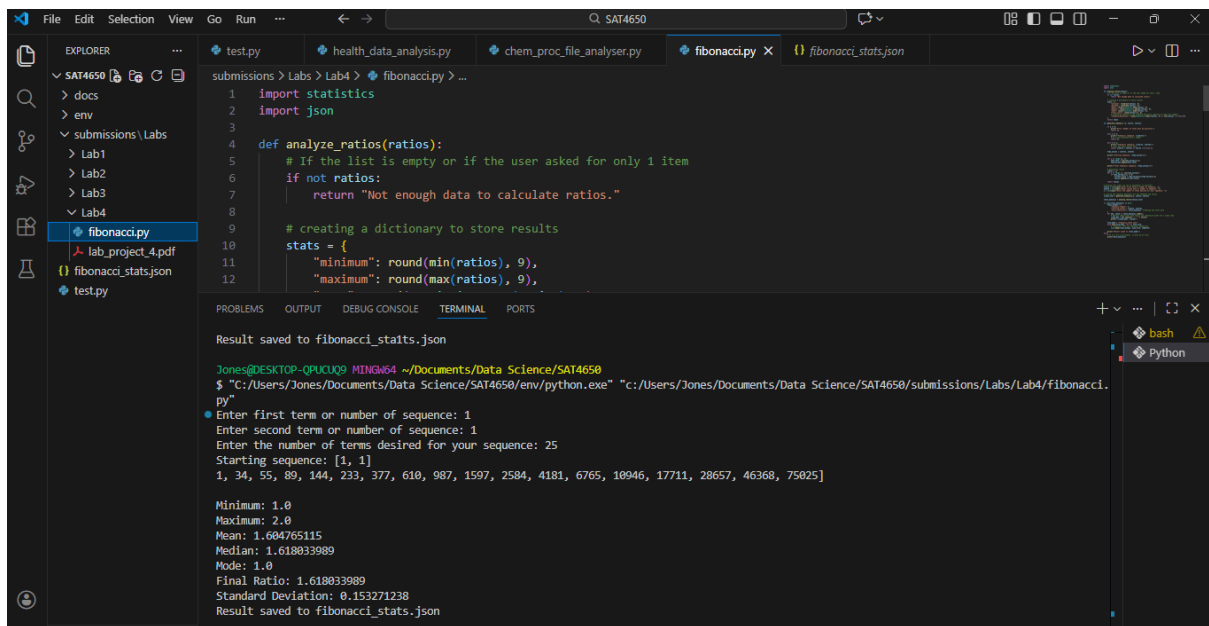


## Console screenshot:



The screenshot shows a Visual Studio Code editor window with the following components:

- EXPLORER:** Displays the file structure of a project named 'SAT4650'. The 'submissions' folder is expanded, showing 'Labs' and 'Lab4'. The 'fibonacci.py' file is selected.
- EDITOR:** Shows the code for 'fibonacci.py'. The code defines an 'analyze\_ratios' function that takes a list of ratios and returns a dictionary with 'minimum', 'maximum', 'mean', 'median', 'mode', 'final\_ratio', and 'standard\_deviation'.
- TERMINAL:** Shows the output of running the script. It prompts the user to enter the first term, second term, and number of terms. The output displays the resulting Fibonacci sequence and its statistical analysis.

```
1 import statistics
2 import json
3
4 def analyze_ratios(ratios):
5     # If the list is empty or if the user asked for only 1 item
6     if not ratios:
7         return "Not enough data to calculate ratios."
8
9     # creating a dictionary to store results
10    stats = {
11        "minimum": round(min(ratios), 9),
12        "maximum": round(max(ratios), 9),
13        "mean": round(statistics.mean(ratios), 9),
14        "median": round(statistics.median(ratios), 9),
15        "mode": round(statistics.mode(ratios), 9),
16        "final_ratio": round(statistics.fmean(ratios), 9),
17        "standard_deviation": round(statistics.stdev(ratios), 9)
18    }
19    return stats
20
21 # Main function
22 def main():
23     # Get user input
24     first_term = int(input("Enter first term or number of sequence: "))
25     second_term = int(input("Enter second term or number of sequence: "))
26     num_terms = int(input("Enter the number of terms desired for your sequence: "))
27
28     # Generate the sequence
29     sequence = generate_fibonacci(first_term, second_term, num_terms)
30
31     # Calculate ratios
32     ratios = calculate_ratios(sequence)
33
34     # Analyze ratios
35     stats = analyze_ratios(ratios)
36
37     # Save results to JSON
38     save_stats(stats)
39
40 if __name__ == "__main__":
41     main()
```

Result saved to fibonacci\_stats.json

Jones@DESKTOP-QPUCU99 MINGW64 ~/Documents/Data Science/SAT4650  
\$ "C:/Users/Jones/Documents/Data Science/SAT4650/env/python.exe" "c:/Users/Jones/Documents/Data Science/SAT4650/submissions/Labs/Lab4/fibonacci.py"

Enter first term or number of sequence: 1  
Enter second term or number of sequence: 1  
Enter the number of terms desired for your sequence: 25  
Starting sequence: [1, 1]  
1, 34, 55, 89, 144, 233, 377, 618, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025]

Minimum: 1.0  
Maximum: 2.0  
Mean: 1.604765115  
Median: 1.618033989  
Mode: 1.0  
Final Ratio: 1.618033989  
Standard Deviation: 0.153271238  
Result saved to fibonacci\_stats.json