SAT 4650 – Applied Computing with Python

Lab 4: Mathematics with Python

Objective

Apply python functions and data analysis techniques to explore numerical patterns in the Fibonacci sequence by generating sequences, computing ratios, analyzing summary statistics, and organizing results using appropriate data structures and file formats. The focus is on programming logic and function design, and you are encouraged to look up (e.g., using online resources) unfamiliar concepts or library functions as needed to understand how to apply them correctly. Although this exercise involves mathematical sequences and statistical terms, no advanced mathematics or statistics background is required.

Background

The Fibonacci sequence is a well-known numerical sequence in mathematics where each term is the sum of the two preceding terms. Starting from two initial values, the sequence grows as new terms are generated iteratively. For example, if the starting values are `1` and `1`, the sequence begins as: `1, 1, 2, 3, 5, 8, 13, 21, 34, 55` and so on. Fibonacci-like sequences appear in many real-world contexts, including population growth models, financial trends, natural patterns, and algorithmic analysis. One interesting property of the Fibonacci sequence is that the ratio of consecutive terms often approaches a stable value as the sequence grows, which makes it useful for numerical exploration and pattern analysis. In this lab, you will explore the statistics of this golden ratio.

Problem 1 – Fibonacci Ratio Analysis – 3.5 marks

1. Write a function named `generate_fibonacci(n, start1, start2)` that generates a Fibonacci sequence of `n` terms, starting from two user-defined initial values (`start1` and `start2`). The function should include the starting values as the first two terms and return the full sequence as a list.
2. Write a function called `analyze_ratios(ratios)` that takes the list of ratios and computes the summary statistics for that list. The function must return the summary stats (TO 9 DECIMAL DIGITS) as a dictionary object and should include:
    a. minimum value
    b. maximum value
    c. mean (average)
    d. median (requires `import statistics`)
    e. mode (requires `import statistics`)
    f. final ratio value aka the last value of your ratio list
    g. standard deviation (requires `import statistics`)

3. Once the above functions are created, use them in your script to call each function in sequence. Your Fibonacci sequence should be of `25` terms and starting values of `1` and `1`. Display the generated Fibonacci sequence and computed ratio statistics using appropriate string formatting, and save the final summary statistics to a JSON file in the specified format.
4. The JSON file should include sequence length and starting values in addition to summary statistics. Find out the use of `indent` argument in json's `dump` method. Use an indent of `4` when writing your results to JSON. An example of what your output JSON should look like is shown below:

```
{
  "sequence_length": 12,
  "starting_values": [1, 1],
  "ratio_statistics": {
    "minimum": 1.5,
    "maximum": 1.6667,
    "mean": 1.618,
    "median": 1.618,
    "mode": 1.618,
    "final_ratio": 1.618,
    "standard_deviation": 0.05
  }
}
```

Problem 2 – Why though? – 0.5 marks

Write a two-paragraph viewpoint on why you think this lab was important. You should google 'Golden Ratio' before that.

Submission Requirements

Please submit the following on Canvas:

1. Your Python source code (preferably in `.py` format). *[Note: You can convert your `.ipynb` notebook to an executable python script using the conversion command: `jupyter nbconvert --to script notebook_name.ipynb`. If it does not work, activate the correct conda environment you are working with and install nbconvert using `conda install conda-forge::nbconvert`. Then try the conversion command again.]*
2. Resulting JSON file.
3. A single screenshot of printed content on console resulting from program run.
4. A .doc/.txt/.pdf solution of problem 2.