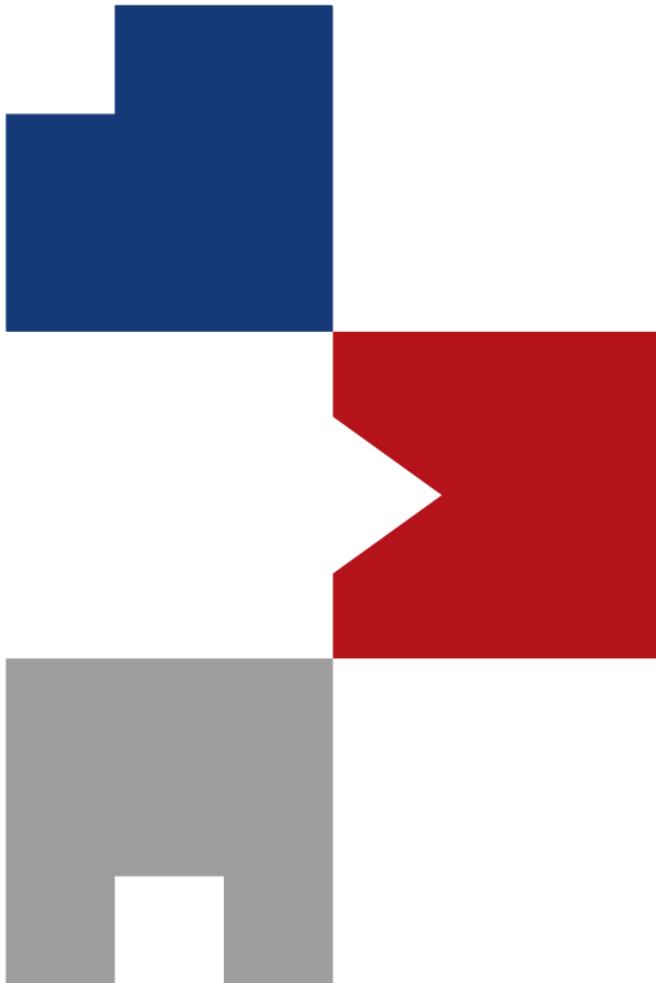


# Python Lab #2:

## Midterm and Final Exam Analysis



Sunglok Choi, Assistant Professor, Ph.D.  
Computer Science and Engineering Department, SEOULTECH  
[sunglok@seoultech.ac.kr](mailto:sunglok@seoultech.ac.kr) | <https://mint-lab.github.io/>

# Overview

- **Prerequisite**

- Anacodna (Individual Edition)

- **Practice) Midterm and Final Exam Analysis**

- The given data
- Expected results
- Practice with the skeleton code
  - Step #1) Read a CSV file as a list of integer numbers
  - Step #2) Calculate the weight average of each row of data
  - Step #3) Calculate mean, variance, median, min, and max of each column of data

- **Assignment**

- Mission: Complete the given skeleton code

## Practice) Midterm and Final Exam Analysis

- The given data (file: data/class\_score\_en.csv)

# midterm (max 125), final (max 100)

113, 86

104, 83

110, 78

101, 79

101, 77

103, 76

71, 94

102, 71

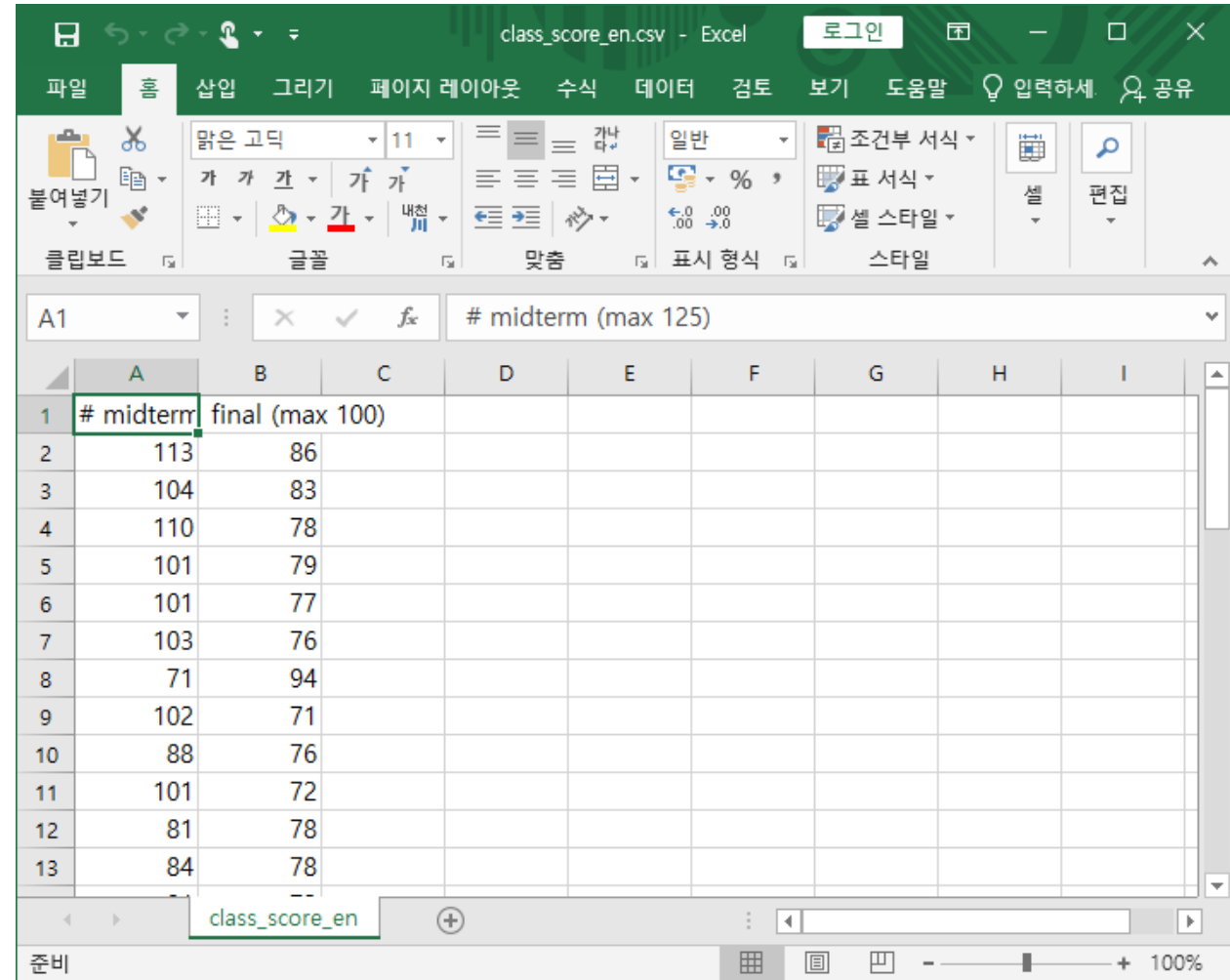
88, 76

101, 72

81, 78

84, 78

...



	# midterm (max 125)	final (max 100)
1	# midterm (max 125)	final (max 100)
2	113	86
3	104	83
4	110	78
5	101	79
6	101	77
7	103	76
8	71	94
9	102	71
10	88	76
11	101	72
12	81	78
13	84	78

# Practice) Midterm and Final Exam Analysis

- Expected results (file: data/class\_score\_analysis.md)
  - Note) The result will be written as a markdown file.
  - **Individual Score**
    - Write each *midterm*, *final*, and its *average* score
      - Weight: Midterm (40%) and final (60%)
      - **Weighted** *average* = **40/125** x *midterm* + **60/100** x *final*
  - **Examination Analysis**
    - Write mean, variance, median, and min/max as summary

## Individual Score

Midterm	Final	Average
113	86	87.760
104	83	83.080
110	78	82.000
...	...	...

## Examination Analysis

- Midterm|
  - Mean: **74.209**
  - Variance: 632.817
  - Median: **72.000**
  - Min/Max: (21.000, 117.000)
- Final
  - Mean: **58.674**
  - Variance: 618.545
  - Median: **66.000**
  - Min/Max: (0.000, 94.000)
- Average
  - Mean: **58.952**
  - Variance: 423.546
  - Median: **65.000**
  - Min/Max: (6.720, 87.760)

# Practice) Midterm and Final Exam Analysis

- The given skeleton code (file: `class_score_analysis_skeleton.py`)

- Step #1) **Read a CSV file** as a list of integer numbers
  - Warning) Please skip the header which starts with #.
  - Implement `read_data()`
- Step #2) **Calculate the weight average**  
**of each row of data**
  - Implement `calc_weighted_average()`
- Step #3) **Calculate mean, variance, median**, and min/max  
**of each column of data**
  - Implement `analyze_data()`

```
def read_data(filename):
    # TODO) Read `filename` as a list of integer numbers
    data = []
    return data

def calc_weighted_average(data_2d, weight):
    # TODO) Calculate the weighted averages of each row of `data_2d`
    average = []
    return average

def analyze_data(data_1d):
    # TODO) Derive summary of the given `data_1d`
    # Note) Please don't use NumPy and other libraries. Do it yourself.
    mean = 0
    var = 0
    median = 0
    return mean, var, median, min(data_1d), max(data_1d)
```

- Note) Please understand 1) how to access multiple data using zip and 2) how to access each column of the 2D list.

# Practice) Midterm and Final Exam Analysis

- The given skeleton code (file: class\_score\_analysis\_skeleton.py)
  - Note) Please understand 1) how to access multiple data using zip and 2) how to access each column of the 2D list.

```
...
if __name__ == '__main__':
    data = read_data('data/class_score_en.csv')
    if data and len(data[0]) == 2: # Check `data` is valid
        average = calc_weighted_average(data, [40/125, 60/100])

    # Write the analysis report as a markdown file
    with open('class_score_analysis.md', 'w') as report:
        report.write('### Individual Score\n\n')
        report.write('| Midterm | Final | Average |\n')
        report.write('| ----- | ----- | ----- |\n')
        for ((m_score, f_score), a_score) in zip(data, average):
            report.write(f'| {m_score} | {f_score} | {a_score:.3f} |\n')
        report.write('\n\n\n')

    report.write('### Examination Analysis\n')
    data_columns = {
        'Midterm': [m_score for m_score, _ in data],
        'Final' : [f_score for _, f_score in data],
        'Average': average }
    for name, column in data_columns.items():
        mean, var, median, min_, max_ = analyze_data(column)
        report.write(f'* {name}\n')
        report.write(f' * Mean: **{mean:.3f}**\n')
        report.write(f' * Variance: {var:.3f}\n')
        report.write(f' * Median: **{median:.3f}**\n')
        report.write(f' * Min/Max: ({min_:.3f}, {max_:.3f})\n')
```

# Assignment

- Mission
  - Complete the given skeleton code (`class_score_analysis_skeleton.py`)
  - Submit your code (`class_score_analysis.py`) and its output (`class_score_analysis.md`)
- Condition
  - Please follow the above filename convention.
  - Please do not use other libraries such as NumPy and others. Do it yourself with only Python built-in functions.
  - You **can** start from scratch (without using the given skeleton code).
    - However, you **should** use the given data.
  - You **can** freely change the given skeleton code if necessary.
- Submission
  - Deadline: **September 25, 2024 23:59** (**firm deadline**; no extension)
  - Where: e-Class > Assignments
  - Score: Max 10 points