



Python: From Beginner to **Intermediate**



Sunglok Choi, Assistant Professor, Ph.D.
Computer Science and Engineering Department, SeoulTech
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 numbers
 - Step #2) Calculate the weight average of each line
 - Step #3) Calculate mean, variance, median, min, and max of all columns

- **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

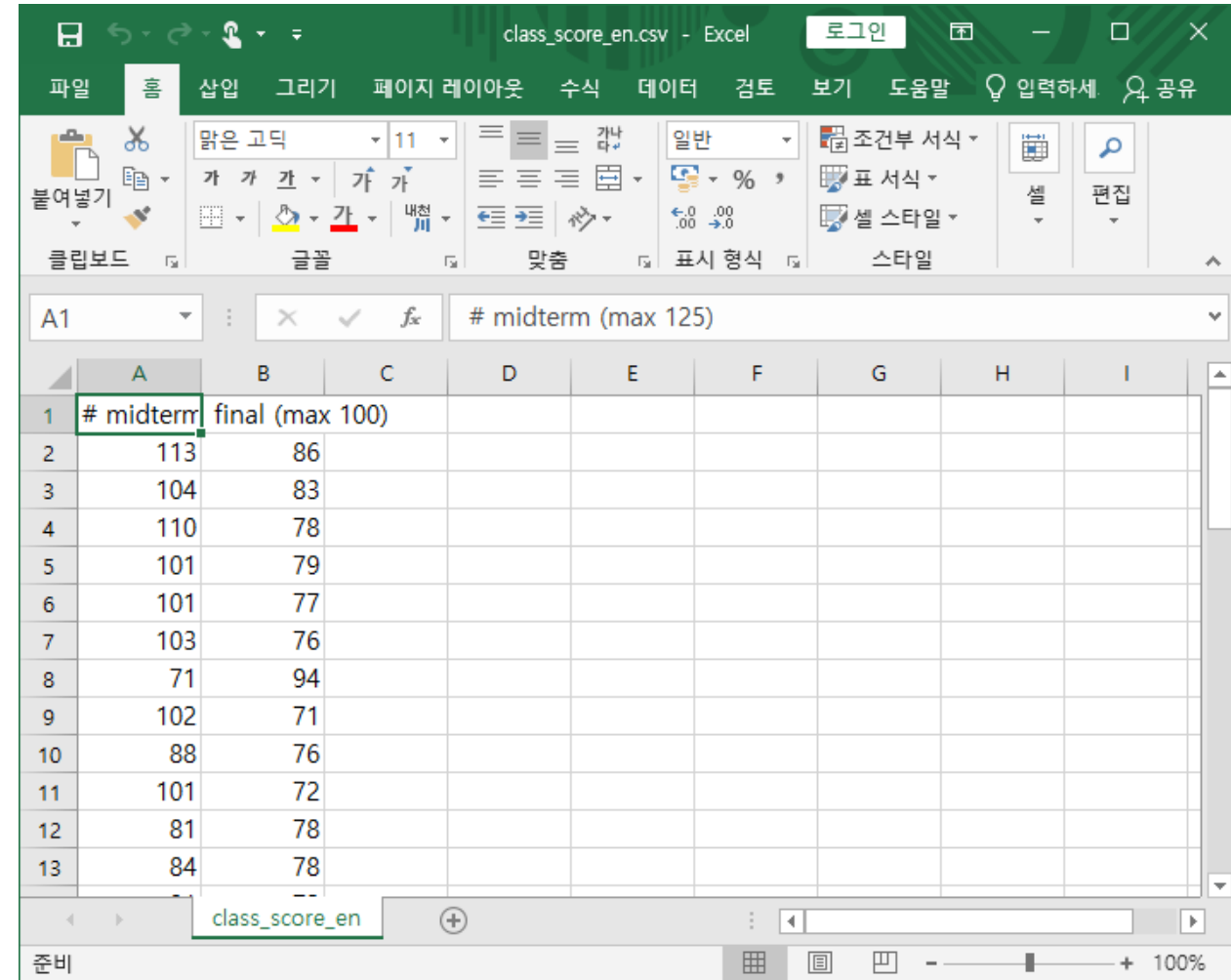
88, 76

101, 72

81, 78

84, 78

...



	# midterm (max 125)	final (max 100)
1	# midterm (max 125)	
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
 - Individual Score
 - Print *midterm*, *final*, and its *total* score
 - $total = 40/125 \times midterm + 60/100 \times final$
 - Examination Analysis
 - Print mean, variance, median, and min/max

class_score_analysis.md - Typora

파일(F) 편집(E) 본문(P) 서식(O) 보기(V) 테마(T) 도움말(H)

Individual Score

Midterm	Final	Total
113	86	87.760
104	83	83.080
110	78	82.000
101	79	79.720
101	77	78.520
103	76	78.560
71	94	79.120
102	71	75.240
88	76	73.760
101	72	75.520
81	78	72.720
84	78	73.680
...		

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)
- Total
 - Mean: **58.952**
 - Variance: 423.546
 - Median: **65.000**
 - Min/Max: (6.720, 87.760)

73 단어

Practice: Midterm and Final Exam Analysis

- The given skeleton code
 - Step #1) **Read a CSV file** as a list of numbers
 - The header starts with #.
 - Implement `read_data()`
 - Step #2) **Calculate the weight average** of each line
 - Implement `add_weighted_average()`
 - Step #3) **Calculate mean, variance, median, min, and max of all columns**
 - How to get **a list of each examination scores**
 - Implement `analyze_data()`

Ind.	Type	Size	Value
0	list	3	[113, 86, 87.76]
1	list	3	[104, 83, 83.08]
2	list	3	[110, 78, 82.0]
3	list	3	[101, 79, 79.72]
4	list	3	[101, 77, 78.52]
5	list	3	[103, 76, 78.56]
6	list	3	[71, 94, 79.12]
7	list	3	[102, 71, 75.24000000000001]
8	list	3	[88, 76, 73.76]
9	list	3	[101, 72, 75.52]
10	list	3	[81, 78, 72.72]
11	list	3	[84, 78, 73.67999999999999]

Ind.	Type	Size	Value
0	list	43	[113, 104, 110, 101, 101, 103, 71, 102, 88, 101, ...]
1	list	43	[86, 83, 78, 79, 77, 76, 94, 71, 76, 72, ...]
2	list	43	[87.76, 83.08, 82.0, 79.72, 78.52, 78.56, 79.12, 75.24, ...]

colwise_data

data

```
def read_data(filename):
    data = []
    # TODO

def add_weighted_average(data, weight):
    for row in data:
        row.append(0)    # TODO

def analyze_data(data):
    mean = 0            # TODO
    var = 0              # TODO
    median = 0          # TODO
    return mean, var, median, min(data), max(data)

if __name__ == '__main__':
    data = read_data('data/class_score_en.csv')
    if data and len(data[0]) == 2: # Check the data is valid
        add_weighted_average(data, [40/125, 60/100])
    if len(data[0]) == 3:         # Check the data is valid
        print('### Individual Score')
        print()
        print('| Midterm | Final | Total |')
        print('| ----- | ---- | ---- |')
        for row in data:
            print(f'| {row[0]} | {row[1]} | {row[2]:.3f} |')
        print()

        print('### Examination Analysis')
        col_n = len(data[0])
        col_name = ['Midterm', 'Final', 'Total']
        colwise_data = [ [row[c] for row in data] for c in range(col_n) ]
        for c, score in enumerate(colwise_data):
            mean, var, median, min_, max_ = analyze_data(score)
            print(f'* {col_name[c]}')
            print(f' * Mean: **{mean:.3f}**')
            ...
```

Assignment

- Mission
 - Complete the given skeleton code (`class_score_analysis_skeleton.py`)
 - Submit your code (`class_score_analysis.py`) and its output (`README.md`)

- Condition
 - Please follow the above filename convention.
 - You **can** start from scratch (without using the given skeleton code).
 - However, you **should** use the same data shown in the slide 5.
 - You **can** freely change the given skeleton code if necessary.

- Submission
 - Deadline: **October 6, 2021 23:59** (**firm deadline**; no extension)
 - Where: e-Class > Assignments
 - Score: Max 10 points