Student Analysis Documentation

By: Saihaj Law

Conversion

At first glance of the assignment I decided I wanted to convert my code into *.csv* files so that I could read from them easier; this proved to be unnecessary as I was able to use a **multi-dimensional array** instead of any alternative methods I was originally considering to use. I have retained this method as it does not have any major effect on our code, but I prefer to use *.csv* files for manipulation of strings anyways.

```
import pandas as pd
import csv
read_file = pd.read_csv (r'student-records-1.txt')
read_file.to_csv (r'student-records-1.csv', index=None)

read_file = pd.read_csv (r'student-records-2.txt')
read_file.to_csv (r'student-records-2.csv', index=None)

read_file = pd.read_csv (r'student-records-3.txt')
read_file.to_csv (r'student-records-3.csv', index=None)
```

All this is doing is reading from the files using the *pandas* library and then converting all of our text files into .csv files

Reading Records

In order to read from the records (*since they are in .csv format*) we have to use a method which is built-in to pandas which is the

```
reader = csv.reader(file)
```

Other than that, the code is simply just initializing an array for the records, reading from the records, and storing the rows of the records into that given array.

```
def read_student_records(filename):
  student_records = []
  for file in filename:
    with open(file, 'r') as file:
      reader = csv.reader(file)
      for row in reader:
```

```
      student_records.append(row)
  return student_records
```

Average Age Calculation

This is also some simple code which takes the length (the number) of different students in our student records, and adds those values into the age variable. After this the average_age is calculated simply based upon the number of students and the added age value (which is then rounded to 2 decimal places).

```
def avg_age_calculation(student_records):
  age = 0
  num_students = len(student_records)
  for student in student_records:
    age += int(student[3])
  average_age = age/num_students
  return round(average_age, 2)
```

Average Grade

Next we have the average grade calculation which is incredibly similar to our above calculation except it is caluculating it based on the index of *[6]* as that is assigned to the student average value.

```
def avg_grade(student_records):
  grade = 0
  num_students = len(student_records)
  for student in student_records:
    grade += float(student[6])
  average_grade = grade / num_students
  return round (average_grade, 2)
```

Highest and Lowest Averages

These share similar code in the sense that they are calculating the max average from obtaining the max value in the "list" of the student records. Then appending that value on to a given variable, rounding that value, and then return that value for later use.

```
def highest_average(student_records):
  max_average = max(float(student[6]) for student in student_records)
  student_max_average = [student for student in student_records if float(student[6]) == max_
  return student_max_average

def lowest_average(student_records):
  lowest_average = min(float(student[6]) for student in student_records)
  student_lowest_average = [student for student in student_records if float(student[6]) == l
  return student_lowest_average
```

Gender Count

This is some more primitive code which is counting the value based on whether or not a given value is “male”. The females are calculated by simply removing the males from the number of total students.

```
def gender_count(student_records):
  males = sum(student[4].lower().strip() == 'male' for student in student_records)
  female = len(student_records) - males
  return males, female
```

Main Loop

In here all we are doing is initializing our file names, and then printing out all the given values which we discussed earlier! (*The empty prints are to create line breaks*)

```
file_names = ["student-records-1.csv", "student-records-2.csv", "student-records-3.csv"]

student_records = read_student_records(file_names)
average_age = avg_age_calculation(student_records)
print(f"The average age of the students is {average_age}")

average_grade = avg_grade(student_records)
print(f"The average grade of all the students is {average_grade}")
print("")

top_average = highest_average(student_records)
for student in top_average:
  print(f"The students with the name {student[1]} {student[2]} has the highest average score

print("")

low_average = lowest_average(student_records)
for student in low_average:
  print(f"The student with the name {student[1]} {student[2]} has the lowest average score o

print("")

male_num, female_num = gender_count(student_records)
print(f"The number of male students is {male_num}")
```

```python
print(f"The number of female students is {female_num}")
print("")
```