

Homework 4_Application_Ronald_Adomako

September 19, 2022

IST 5520 Homework 4: Application

The pandas Package

Student: Ronald Adomako

Instruction:

Complete the following tasks (8 points in total). You can add additional lines if necessary. You must implement the programming logic to create the required objects, rather than manually typing values to create objects unless manual input is specifically required (for example task 1.1).

```
[1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

1 Task 1: pandas Series (4 points)

1.1 Manually create a pandas Series object named 'score' that contains the following elements:

51, 78, 84, 95, 64, 82, 89, 68, 91

```
[2]: score = pd.Series([51, 78, 84, 95, 64, 82, 89, 68, 91])
score
```

```
[2]: 0    51
1    78
2    84
3    95
4    64
5    82
6    89
7    68
8    91
dtype: int64
```

1.2 Append one more value 90 into the score series by using `pd.concat()` method. The `pd.concat()` method can concatenate two Series objects into one Series object. Check the method' signature or document if you are not familiar with its usage.

Hints: - Call the `pd.concat()` method to concatenate two Series objects into one Series object. - The index should be from 0 to 9. If the index of the result is not correct, call the `reset_index()` to reset the index.

```
[3]: score2 = pd.Series([90])
      score2
```

```
[3]: 0    90
      dtype: int64
```

```
[4]: score1 = pd.concat([score,score2])
      score1
```

```
[4]: 0    51
      1    78
      2    84
      3    95
      4    64
      5    82
      6    89
      7    68
      8    91
      0    90
      dtype: int64
```

```
[5]: score1.index
```

```
[5]: Int64Index([0, 1, 2, 3, 4, 5, 6, 7, 8, 0], dtype='int64')
```

```
[6]: #score1.reset_index?
```

```
[7]: #need to include the drop statement when using a series
      score1.reset_index(drop = True, inplace=True)
```

```
[8]: score1
```

```
[8]: 0    51
      1    78
      2    84
      3    95
      4    64
      5    82
      6    89
      7    68
      8    91
      9    90
      dtype: int64
```

1.3 Calculate the mean of all elements in the Series object.

79.2

```
[18]: mean = score1.mean()  
      mean
```

[18]: 79.2

1.4 Sort the Series object in descending order.

```
[23]: #score1.sort_values?  
      score1.sort_values(0, ascending=False, inplace = True)  
      score1
```

```
[23]: 3    95  
      8    91  
      9    90  
      6    89  
      2    84  
      5    82  
      1    78  
      7    68  
      4    64  
      0    51  
      dtype: int64
```

```
[24]: score1.reset_index(drop = True, inplace=True)  
      score1
```

```
[24]: 0    95  
      1    91  
      2    90  
      3    89  
      4    84  
      5    82  
      6    78  
      7    68  
      8    64  
      9    51  
      dtype: int64
```

2 Task 2: Loop (4 points)

2.1 Define a function to convert a score value to a letter grade.

```
[25]: def score2letter(num):  
      if num >=90:
```

```

    return "A"
elif num >= 80:
    return "B"
elif num >= 80:
    return "C"
elif num >= 80:
    return "D"
else:
    return "F"

```

2.2 Use a for loop to calculate letter grade for each value in the score object. Store the letter grade in a pandas Series object called “grade”. Show the content of the grade Series.

Hint: - The index should be from 0 to 9. If the index of the result is not correct, call the `reset_index()` to reset the index.

```

[26]: grade = pd.Series()
      for score in score1:
          letter = score2letter(score)
          #print(letter)
          #print(type(letter))
          letter = pd.Series(letter)
          grade = grade.append(letter)
      grade

```

/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1:

DeprecationWarning: The default dtype for empty Series will be 'object' instead of 'float64' in a future version. Specify a dtype explicitly to silence this warning.

"""Entry point for launching an IPython kernel.

```

[26]: 0    A
      0    A
      0    A
      0    B
      0    B
      0    B
      0    B
      0    F
      0    F
      0    F
      0    F
      dtype: object

```

```

[27]: grade = grade.reset_index(drop=True)

```

```

[28]: grade

```

```
[28]: 0    A
      1    A
      2    A
      3    B
      4    B
      5    B
      6    F
      7    F
      8    F
      9    F
      dtype: object
```

2.3. Combine the score and grade arrays into a pandas DataFrame. The 1st column shows the score and the 2nd column shows the letter grade. Make sure the two columns are named as 'score' and 'grade' respectively. Show the content of this DataFrame. (2 points)

Hint: - Use the `pd.concat()` method to concatenate multiple Series objects into a DataFrame object.

```
[29]: df = pd.DataFrame({'Score':score1, 'Letter':grade})
      df
```

```
[29]:   Score Letter
      0     95     A
      1     91     A
      2     90     A
      3     89     B
      4     84     B
      5     82     B
      6     78     F
      7     68     F
      8     64     F
      9     51     F
```