Problem Definition

Let’s say we have 8 users, and they have rated 8 different albums on a scale of 1 to 5. Note that not all users have rated all albums.

```
songData3 = {"Angelica": {"Blues Traveler": 3.5, "Broken Bells": 2.0, "Norah Jones": 4.5, "Phoenix": 5.0,
"Slightly Stoopid": 1.5, "The Strokes": 2.5, "Vampire Weekend": 2.0},
            "Bill":{"Blues Traveler": 2.0, "Broken Bells": 3.5, "Deadmau5": 4.0, "Phoenix": 2.0, "Slightly
        Stoopid": 3.5, "Vampire Weekend": 3.0},
            "Chan": {"Blues Traveler": 5.0, "Broken Bells": 1.0, "Deadmau5": 1.0, "Norah Jones": 3.0,
        "Phoenix": 5, "Slightly Stoopid": 1.0},
            "Dan": {"Blues Traveler": 3.0, "Broken Bells": 4.0, "Deadmau5": 4.5, "Phoenix": 3.0, "Slightly
        Stoopid": 4.5, "The Strokes": 4.0, "Vampire Weekend": 2.0},
            "Hailey": {"Broken Bells": 4.0, "Deadmau5": 1.0, "Norah Jones": 4.0, "The Strokes": 4.0,
        "Vampire Weekend": 1.0},
            "Jordyn":  {"Broken Bells": 4.5, "Deadmau5": 4.0, "Norah Jones": 5.0, "Phoenix": 5.0,
        "Slightly Stoopid": 4.5, "The Strokes": 4.0, "Vampire Weekend": 4.0},
            "Sam": {"Blues Traveler": 5.0, "Broken Bells": 2.0, "Norah Jones": 3.0, "Phoenix": 5.0,
        "Slightly Stoopid": 4.0, "The Strokes": 5.0},
            "Veronica": {"Blues Traveler": 3.0, "Norah Jones": 5.0, "Phoenix": 4.0, "Slightly Stoopid":
        2.5, "The Strokes": 3.0}
           }
```

Requirement for this Assignment

You have been provided with a framework for this assignment: “Assignment 1 – Framework.py”.

The Framework defines a Class called ***similarity***. The class includes a class initialization method which takes two rating dictionaries ***ratingP*** and ***ratingQ*** as parameters. The class has two methods:

- ***minkowski*** which takes a single parameter ***r***, and returns the Minkowski Distance between the two dictionaries (that the Class object is instantiated with). You will not be using this method in this assignment.
- ***pearson*** which takes no parameters, and returns the Pearson Correlation between the two dictionaries (that the Class object is instantiated with). You will be coding, and then using this methods in this assignment.

You **must not** change the class framework in any way. This means:

- You must not introduce any new methods in the class
- You must not pass any additional inputs to the existing methods. That is, the minkowski method can only take one parameter r, and the pearson method can take no parameters.

Part 1 of this assignment is to write code for the pearson method using the computationally efficient form. Pseudo-code has been provided to you in the framework. So, you essentially need to plug in appropriate code for Steps 1.1 through 1.7 in the framework.

$$r=\frac{\sum_{i=1}^{n} p_i q_i - \frac{\sum_{i=1}^{n} p_i \sum_{i=1}^{n} q_i}{n}}{\sqrt{\sum_{i=1}^{n} p_i^{2} - \frac{\left(\sum_{i=1}^{n} p_i\right)^{2}}{n}}\sqrt{\sum_{i=1}^{n} q_i^{2} - \frac{\left(\sum_{i=1}^{n} q_i\right)^{2}}{n}}}$$

Part 2 of this assignment is to find the NN (k=1) recommendations for a given userX, based on the <u>Pearson Correlation similarity measure</u> you coded in Part 1. Pseudo-code has been provided to you in the framework. So, you essentially need to plug in appropriate code for Steps 2.1 through 2.5 in the framework.

Assignment Submission

Some things to keep in mind as you code:

- Make your code readable – for instance, use meaningful variable names and comments.
- Make your code elegant – for instance, balance the number of variables you introduce – too many or too few make your code difficult to debug, read, and maintain.
- Make your output readable and user-friendly

Once you have written up the script, save it as follows. Submit the script by uploading your python script. Note: upload the actual script – DO NOT attach a screenshot of the script!

<CohortX><FirstName><LastName>Assignment1.py.

[Example: CohortAHinaAroraAssignment1.py]

The submitted script will be run <u>**as-is**</u> for grading. **I will be plugging in different users for userX to see if your code is giving me the correct recommendations.**

Points will be deducted for scripts that:

- are difficult to read/follow
- don't compile/run
- don't have all the various pieces of code required
- have hard-code values instead of using variables
- have logical errors
- don't result in the expected output
- don't have user-friendly output