CIS 9 - Sample questions for midterm 1

Numpy

1. What is the output?

arr = np.zeros((4,2))

print(arr[-1])

Answer: [0. 0.]

2. Write code to print the first, middle, and last element of a numpy array named arr.

print(arr[[0, len(arr)//2, -1]])

3. Explain what this prints: print(arr[arr%2==0])

Answer: all elements of arr that are even numbers

4. You have an array of one year or 365 average daily temperatures, in degree Celsius.

Convert them to Fahrenheit. The formula is: F = (C x 9/5) + 32

Answer: arr\*9/5+32

5. In the temperature array above, find the index of the highest temperature.

Answer: highest = arr.max()

np.where(arr==highest)

Someone in class asked if this answer is okay: np.argsort(arr)[-1]

This will give the same answer if there is exactly one highest temperature.

If there are 2 or 3 elements with the same highest temperature, then the first answer with np.where will give the locations of all the highest values, but the second answer with np.argsort will only give one location.

6. What is the output?

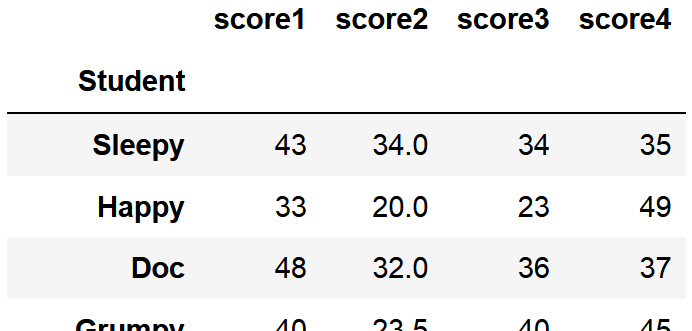
arr = np.array([[1,4], [7,-1],[2,5]])

print(np.median(arr, axis=0))

Answer: [2. 4.]

Pandas

Given the following DataFrame named gb, with the first few rows shown but there are many other rows:



1. Print Happy's scores

Answer: gb.loc["Happy"]

Someone in class asked if it's possible to select a slice of rows, such as from rows 2 to 5, and the answer is no. It's not possible to do a slice with strings.

It is possible to select multiple rows: gb.loc[["Happy", "Doc"]] but not select a block of consecutive rows when the index values are strings.

If you have an application where you might want to select a block of rows with a slice, then it's best to let pandas use the default indexing (0,1, 2, 3...)

For the gradebook example, you'd read it in with: gb = pd.read\_excel("excel\_filename.xlsx")

and you'll get the default numeric indexing.

2. Find the mean of score4

Answer: gb.score4.mean()

3. Explain what this prints: print(gb.max())

Answer: print the max of all 4 scores

4. Change Doc's score4 to 50

Answer: gb.at["Bashful", "score4"] = 50

5. Explain what this prints: print(gb[gb.score1 > 45].score1)

Answer: print all score1 values that are greater than 45

Matplotlib

1. Given the gb DataFrame above, you want to know whether score1 is generally better or worse than score 2.

Write code to create a plot that compares the 2 scores.

plt.hist(gb.score1, alpha=0.2, label="score1") # compare distribution of score1 and score2

plt.hist(gb.score2, alpha=0.2, label="score2")

plt.legend()