Imagine a fantasy world of dragons. We are going to store the dragons data in a numpy array (call it D). Create 100 dragons with a unique numerical ID.

Each dragon can be one of 4 elements: Earth, Water, Fire, and Air. Code these elements as 0, 1, 2, and 3, respectively. Assign the elements to the dragons using the following probability distribution. A dragon has 50% chance of being Earth, 30% - Water, 10% - Fire, and 10% - Air. Using np.sum() calculate the total number of dragons of each element and print this information. Notice that the numbers may slightly differ from the 50-30-10-10 distribution (given the random draw).

Imagine that each dragon has a cave where it stores its wealth. Assume that each Earth dragon has 100 bags of gold; Water – 120, Fire – 200, and Air – 20. Try to use numpy functionality to model this (avoid for loops if possible).

Imagine that the dragon world is under attack by a powerful enemy. The enemy can be either Alpha or Beta (with equal probability). If the enemy is Alpha: all Earth and Water dragons have to give up 10 bags of gold (each), Fire dragons have to give all of their gold, Air dragons keep all of their gold. However, if the enemy is Beta: all Dragons double their wealth.

Model the enemy attack on the dragon world. Save the total wealth and average wealth for each type of dragons in another array (call it data). (Hint: you will need at least 8 columns in data since you have 4 types of dragons and you are saving total and average values; you may also want to store the total number of dragons of each type – in which case you will need 12 columns). Try to avoid for loops for this task (unless you are really stuck).

Imagine that the dragon world experienced 10 such attacks. Create a figure with two subplots: (1) total wealth for each type of dragons over time, and (2) average wealth for each of dragons over time. The figure may look something like this (the format; actual values will of course be different):