Project 1

Project 1 is all about biodiversity: the variety of different species that exist in a particular habitat. Biodiversity is important because ecosystems with a wider range of species are typically more productive, and have a greater ability to withstand disturbance. A major challenge with biodiversity is how to measure it: we can typically only observe a sample of a large and complex habitat. Birds are considered a good indicator of a healthy ecosystem as they are highly mobile. If they don't like their current surroundings, they will move somewhere else. Even after deciding which species to observe, we still have the problem of working out how to quantify the diversity we observe.

In this project you will write Python functions to explore different questions about biodiversity, in the context of surveying, analysing and predicting the biodiversity of bird species.

Question 1

One of the most straightforward ways to quantify the biodiversity of an environment is simply to count the number of different species that can be found there, irrespective of how many times a particular species is observed. This measure is known as species richness.

Write a function `get_species_richness()` that calculates the species richness of a habitat, based on a series of observations of various bird species. The function takes one argument: `observed_list`, a list of independent observations of bird species. The function should return a tuple consisting of:

- the *species richness*, calculated as the number of different species observed; and
- an alphabetically sorted list of the species that were observed.

Question 2

A limitation of species richness is that it doesn't differentiate between common and rare species. Each species is counted only once, no matter how many birds of that species are observed. Another approach to measuring biodiversity is species evenness, which takes account of similarity in the number of times each bird species is observed. This measure is also known as relative abundance.

One measure of evenness is Simpson's index

Write a function `get_species_evenness()` that calculates the species evenness of a habitats, based on a series of observations of various bird species. As in Question 1, the function takes one argument:

`observed_list`, a list of independent observations of birds.

The function should return a tuple consisting of:

- the *species evenness*, calculated as the inverse of Simpson's index as described in the slide above; and
- a list of tuples, consisting of a bird species and the number of times it was observed, sorted alphabetically by species.

If the list of species is empty, then the value of evenness would be 0 (as an int).

Question 3

Measures of biodiversity are often used in a comparative setting, to analyse data from multiple habitats. By comparing measures of diversity across different sites, we can look for patterns that might help to understand why some areas show higher diversity than others.

Write a function `compare_diversity()` that ranks a set of a habitats by one of the diversity metrics from Questions 1 and 2. The function takes two arguments:

- `observed_list`, a list of independent observations of birds; each observation is now a tuple consisting of the species of the bird, and the habitat is was observed in; and
- `diversity_measure`, a string describing the measure of diversity to be used in ranking the habitats; this string will take one of two values: *richness* or *evenness*.

The function should return a list of tuples, with each tuple consisting of the habitat name, and the diversity of that habitat, according to the specified measure. This list should be sorted from most diverse to least diverse habitat. Where more than one habitat has the same level of diversity, these should be sorted alphabetically.

The first line of default code in your workspace `from hiddenimport get_species_richness, get_species_evenness` allows you to use our implementation of functions from Q1 and Q2: there is no need to copy/paste your solutions to those questions to use them, and you can be confident that these functions will work correctly in all cases. Do not delete this line or the functions will be no longer available.

Optimal Sampling of Habitats

Now we will turn our attention to how data on bird species are collected. It is very rarely possible to record data on every single animal living in a particular habitat. Rather, researchers will visit a habitat on one or more occasions to record a sample of bird species observed there. This sampling takes time and effort, and each additional visit is likely to result in more observations of species that have already been recorded and fewer new species.

Sampling may continue for a particular period of time, or in a particular subset of the habitat, or until some other stopping criterion is met. From this sample, estimates can then be made of bird species diversity in that habitat.

One approach to defining a stopping rule is on the basis of the observations recorded up to that point. For example, we may decide to stop collecting samples once there have been a specified number of *consecutive visits* that are deemed unproductive. An unproductive visit is one in which the number of *previously unseen species* observed is below a specified threshold. If you reach the end of the list and the threshold has not been reached you can assume that the sample is representative of the bird population.

Determining the appropriate consecutive visits and unseen species thresholds for a given habitat is important. If these values are set too low, there is a risk that sampling may stop before sufficient data has been collected on bird species, leading to an underestimate of diversity.

Question 4

Write a function `optimise_study()` that evaluates the effect that consecutive visits and unseen species thresholds have on the accuracy of diversity estimates. The function takes three parameters:

- `sample_data` is a list of lists of data collected over the course of multiple sampling visits; each item in the main list is a list of species that were observed on that visit;
- `unseen_species` (an `int`) is the minimum number of previously unseen species that must be observed before a visit is deemed productive; and
- `consecutive_visits` (an `int`) is the number of consecutive unproductive visits, after the initial visit, that must occur to trigger the stopping rule.

The functon should return a tuple containing:

- the number of visits that will occur before the study has stopped; and
- the proportion of the total bird species observed by the study at that point, compared to if all sampling visits contained in sample_data had been conducted.

Note that the species listed in each sampling visit indicate only the presence of that species; abundance (the number of observations of that species) is not captured. Thus a species will only appear at most once in the list for a particular sampling visit. It may of course appear again in lists for other sampling visits.