Transform the percentage cover values into ordinal scale

David Zelený

Introduction

In vegetation ecology, researchers often visually estimate the cover of plant species in the sampled communities. One option is to directly use percentage values for individual plant species, but more common is to use values on an ordinal scale (ordinal scale = values are sorted, but distances between the values are not the same; an example of ordinal scale is encoding of verbal temperature expression cold < warm < hot into numerical ordinal scale 1 < 2 < 3). Vegetation ecologists often use the so-called *Braun-Blanquet cover scale* for estimating the cover of species in the plot, since it is faster and more efficient than using the percentage scale.

Table of Braun-Blanquet cover scale used to estimate the species cover (zero should stay zero):

Percentage cover value	Braun-Blanquet code	Ordinal value
= 0	0	0
> 0 & <= 1	r	1
> 1 & <= 2	+	2
> 2 & <= 3	1	3
> 3 & <= 25	2	4
> 25 & <= 50	3	5
> 50 & <= 75	4	6
> 75 & <= 100	5	7

What to do

1. Write a function `transform2brbl (data)`, which will transform the percentage values of species covers in the argument `data` into ordinal values (0-7) using the Braun-Blanquet scale above. Do not transform the values into the Braun-Blanquet code (0, r, +, 1, 2, 3, 4, 5), but into the numerical ordinal values (0, 1, 2, 3, 4, 5, 6, 7), so that it can be used in further numerical analyses.
2. When you finish the function definition, include also the following two tests to see its functionality:
    1. As a data, use the vector with predefined values:

    ```
    transform2brbl (0:100)
    ```

    and you should get:

    ```
     [1] 0 1 2 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5
    [33] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6
    [65] 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
    [97] 7 7 7 7 7
    ```

    2. As a data, use species data from `vltava` dataset:

```
vltava.spe <- read.delim
('https://raw.githubusercontent.com/zdealveindy/recol/main/data/vltava-spe.txt',
row.names = 1)
transform2brbl (vltava.spe)[1:10, 1:7]
```

and you should get the following result:

```
   Abiealb23 Acerpla23 Acerpse23 Alnuglu23 Alnuinc23 Berbvul23 Betupen23
1          0         0         0         0         0         0         0
2          0         0         0         0         0         0         0
3          0         0         0         0         0         0         0
4          0         0         0         0         0         0         0
5          0         0         0         5         0         0         0
6          0         5         0         0         0         0         0
7          0         0         2         0         0         0         0
8          0         0         2         0         0         0         0
9          0         2         0         0         0         0         3
10         0         0         0         0         0         0         2
```

3. The whole code which you should upload to NTU COOL will include definition of the function, and both tests of it, including the output values.

Hints

- Perhaps the easiest way is to use the sequence of `ifelse` functions, nested within each other. Alternatively, you may use `for` loop, and evaluate each element of the matrix independently using `if` and `else` construct. Do not forget that zero cover should stay zero!