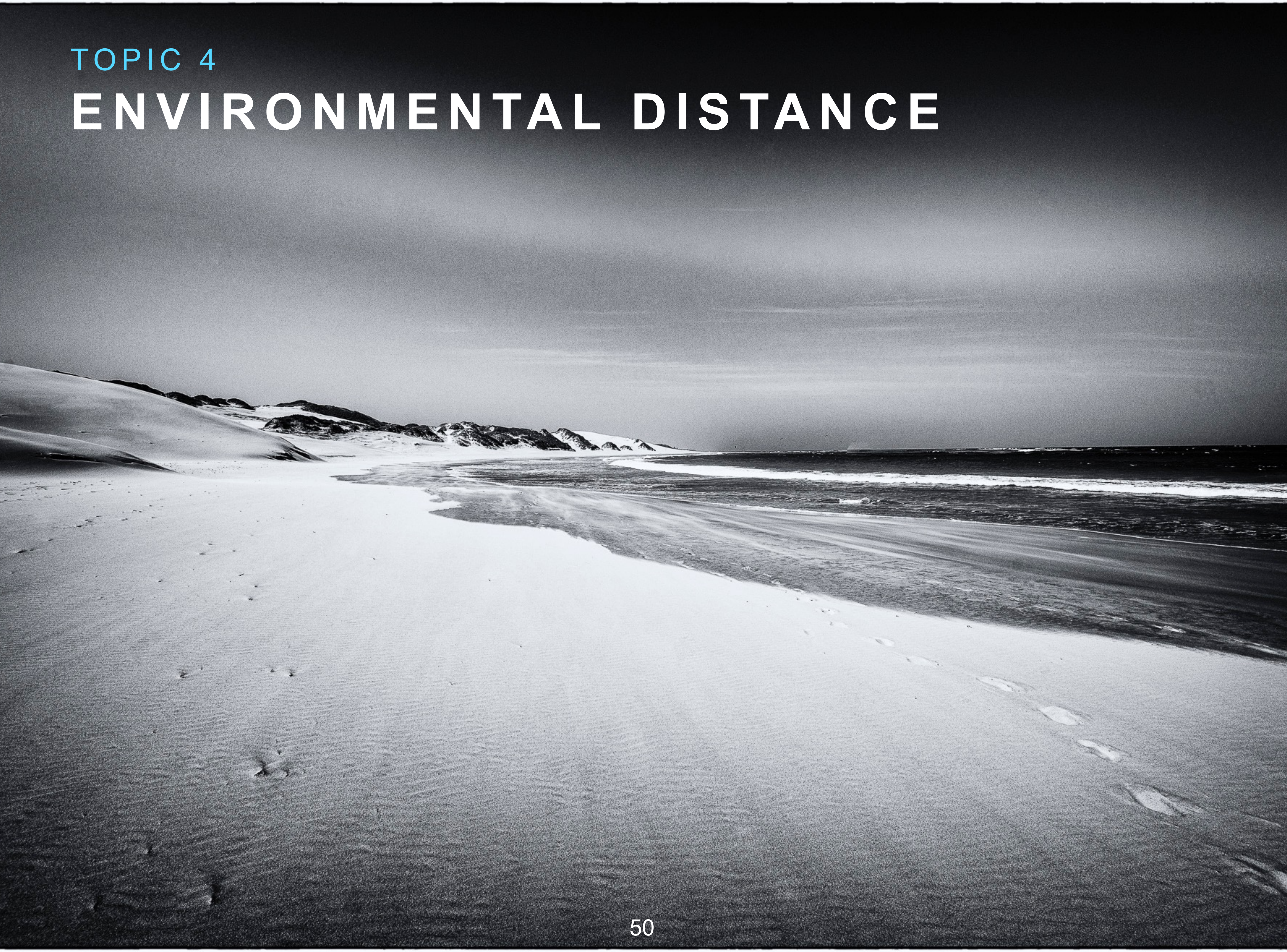
TOPIC 4

ENVIRONMENTAL DISTANCE



Similarity/dissimilarity and distance

- Species:
  - sites sharing a similar species composition are ecologically similar
    - i.e. high similarity / low dissimilarity;
  - we will cover dissimilarities in Topic 5.
- Environmental variables:
  - sites sharing similar environmental conditions have a low environmental distance between them;
  - how similar sites to each other depends on…
    - measurable environmental differences that influence species composition,
    - it can be due to unmeasured influences, or
    - it can also simply be noise.
- It is the ecologist's role to figure out what influences the similarity/dissimilarity/distances between sites.

Distance matrices

- A **distance matrix** is produced from a data table (**species table** or **environment table**) by calculating one of several dissimilarity indices.
- Also called **association** or **resemblance** matrices.
- See `vegdist()` **vegan** for a list of dissimilarity indices.
- The result is a matrix of **pairwise differences** in community composition (as synthesised by the chosen index) or ecological distance between all sites.

Distance matrix for environmental data

- **Euclidian distance** is “the ‘ordinary’ straight-line distance between two points in Euclidean space” (i.e. in its simplest form a planar area, which you know of as a graph with x- and y-axes)
- So, in 2D and 3D, gives distance in **cartesian units** between points on a plane (x, y) or in a volume (x, y, z).
- Conforms to our physical concept of distance
    - e.g. short geographic distances between points on a map, and
    - (loses accuracy over large distances, as Earth’s surface is not on a plane but on a sphere… correct by using great circle distances, e.g. use the Haversine formula).
- Calculated using the Pythagorean theorem
    - differences are squared, so single large differences become very important, and
    - this is not useful for species data.

Distance matrices

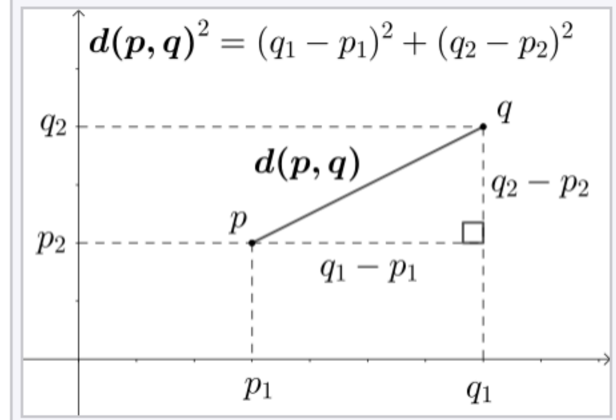
- The matrices are **square** and **symmetrical**, and they will have as many rows and columns as the number of sites present in the original species or environment table.
- The **diagonals are zero** (a site is the same as itself, so it has zero dissimilarity).
- The table can be read directly, and each cell represents the species or ecological difference between a pair of sites.
- All information of the species ID (and maybe also abundance) of a site is lost, as this info is condensed into one metric.

Two dimensions [edit]

In the Euclidean plane, if **p** = (p_1, p_2) and **q** = (q_1, q_2) then the distance is given by

$$d(\mathbf{p},\mathbf{q}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}.$$

This is equivalent to the Pythagorean theorem.



Euclidean distance in R^2

Alternatively, it follows from (*2*) that if the polar coordinates of the point **p** are (r_1, θ_1) and those of **q** are (r_2, θ_2), then the distance between the points is

$$\sqrt{r_1^2 + r_2^2 - 2r_1r_2\cos(\theta_1 - \theta_2)}.$$

Three dimensions [edit]

In three-dimensional Euclidean space, the distance is

$$d(\mathbf{p},\mathbf{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2}.$$

n dimensions [edit]

In general, for an *n*-dimensional space, the distance is

$$d(\mathbf{p},\mathbf{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_i - q_i)^2 + \cdots + (p_n - q_n)^2}.$$

https://en.wikipedia.org/wiki/Euclidean_distance

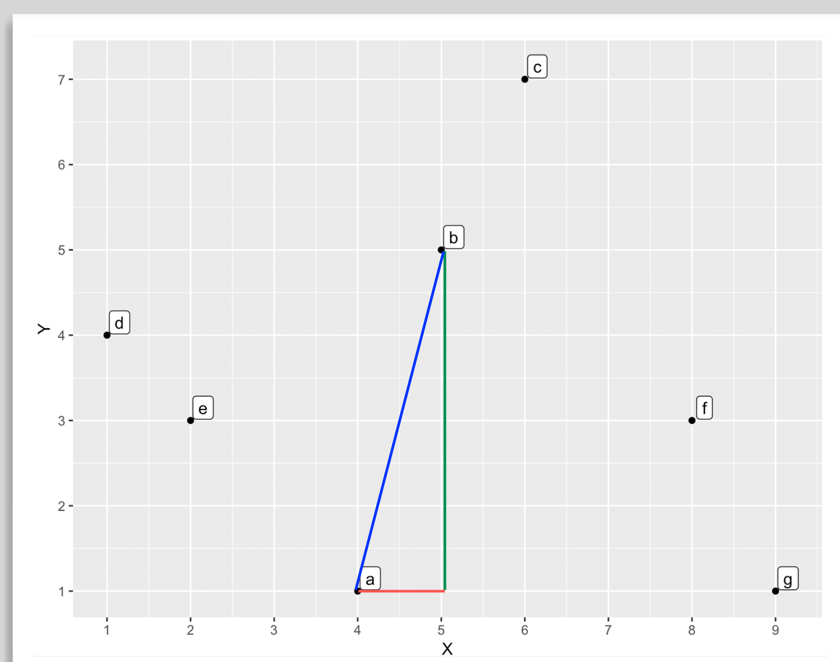
e.g. example with position (such as geographic) coordinates…
use **vegan**'s `vegdist()` function

Raw data

site	x	y
a	4	1
b	5	5
c	6	6
d	1	4
e	2	3
f	8	3
g	9	1

Euclidian distances

```
> ex.xy.euc <- vegdist(ex.xy, method = "euclidian")
> ex.xy.euc
         a        b        c        d        e        f
b 4.123106
c 6.324555 2.236068
d 4.242641 4.123106 5.830952
e 2.828427 3.605551 5.656854 1.414214
f 4.472136 3.605551 4.472136 7.071068 6.000000
g 5.000000 5.656854 6.708204 8.544004 7.280110 2.236068
```



$$d(a, b) = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$$

e.g. example with 3D position coordinates (a.k.a. dimensions)…

Raw data

site	x	y	z
a	4	1	3
b	5	5	5
c	6	6	4
d	1	4	9
e	2	3	8
f	8	3	1
g	9	1	5

Euclidian distances

```
R> ex.xyz.euc ← vegdist(ex.xyz[,2:4], method = "euclidian")
R> ex.xyz.euc
         1        2        3         4        5        6
2 4.582576
3 5.477226 1.732051
4 7.348469 5.744563 7.348469
5 5.744563 4.690416 6.403124  1.732051
6 4.898979 5.385165 4.690416 10.677078 9.219544
7 5.385165 5.656854 5.916080  9.433981 7.874008 4.582576
```

$$d(a, b) = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2}$$

e.g. example with environmental 'dimensions'…

Raw data

site	temperature	depth	light
a	4	1	3
b	5	5	5
c	6	6	4
d	1	4	9
e	2	3	8
f	8	3	1
g	9	1	5

Euclidian distances

```
R> ex.xyz.euc ← vegdist(ex.xyz[,2:4], method = "euclidian")
R> ex.xyz.euc
          1         2         3         4         5         6
2  4.582576
3  5.477226  1.732051
4  7.348469  5.744563  7.348469
5  5.744563  4.690416  6.403124  1.732051
6  4.898979  5.385165  4.690416 10.677078  9.219544
7  5.385165  5.656854  5.916080  9.433981  7.874008  4.582576
```

$$d(a, b) = \sqrt{(a_{temp} - b_{temp})^2 + (a_{depth} - b_{depth})^2 + (a_{light} - b_{light})^2}$$

e.g. example with higher dimension environmental data…

Raw data

	pH	O2	temp	depth
a	7.1	6.5	12.1	1.1
b	7.5	5.5	12.3	1.3
c	7.6	5.7	11.9	1.5
d	7.0	5.4	11.8	1.6
e	7.1	6.3	12.0	1.8
f	7.2	6.3	12.1	1.9
g	6.9	6.1	12.2	2.2

(transformation)

Standarised data

```
> ex.env.std ← decostand(xy.env, method = "standardize")
> ex.env.std
          pH         O2       temp       depth
a -0.3872983  1.2156767  0.2494233 -1.41749621
b  1.1618950 -1.0842522  1.4133987 -0.88114629
c  1.5491933 -0.6242664 -0.9145521 -0.34479637
d -0.7745967 -1.3142450 -1.4965398 -0.07662142
e -0.3872983  0.7556909 -0.3325644  0.45972850
f  0.0000000  0.7556909  0.2494233  0.72790346
g -1.1618950  0.2957051  0.8314110  1.53242833
```

Euclidian distances

```
                       ex.env.std
> ex.env.euc ← vegdist(xy.dat, method = "euclidian")
> ex.env.euc
         a        b        c        d        e        f
b 4.123106
c 6.324555 2.236068
d 4.242641 4.123106 5.830952
e 2.828427 3.605551 5.656854 1.414214
f 4.472136 3.605551 4.472136 7.071068 6.000000
g 5.000000 5.656854 6.708204 8.544004 7.280110 2.236068
```