

Areal Data: Rasters

HES 505 Fall 2023: Session 9

Matt Williamson

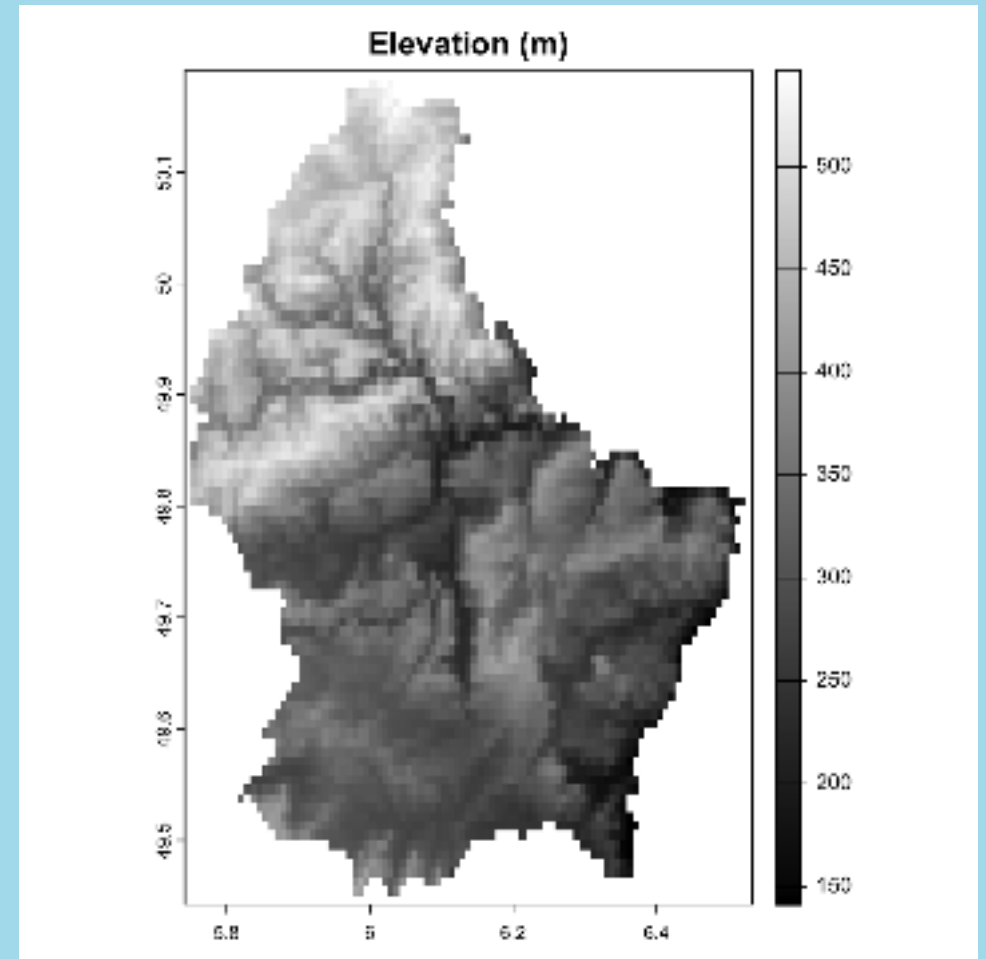
Today's Plan

By the end of today, you should be able to:

- Access the elements that define a raster
- Build rasters from scratch using matrix operations and **terra**
- Evaluate logical conditions with raster data
- Calculate different measures of raster data

Revisiting the Raster Data Model

- **Vector data** describe the “exact” locations of features on a landscape (including a Cartesian landscape)
- **Raster data** represent spatially continuous phenomena (**NA** is possible)
- Depict the alignment of data on a regular lattice (often a square)
 - Operations mimic those for **matrix** objects in **R**
- Geometry is implicit; the spatial extent and number of rows and columns define the cell size



Rasters with **terra**

- syntax is different for **terra** compared to **sf**
- Representation in **Environment** is also different
- Can break pipes, **Be Explicit**

Rasters by Construction

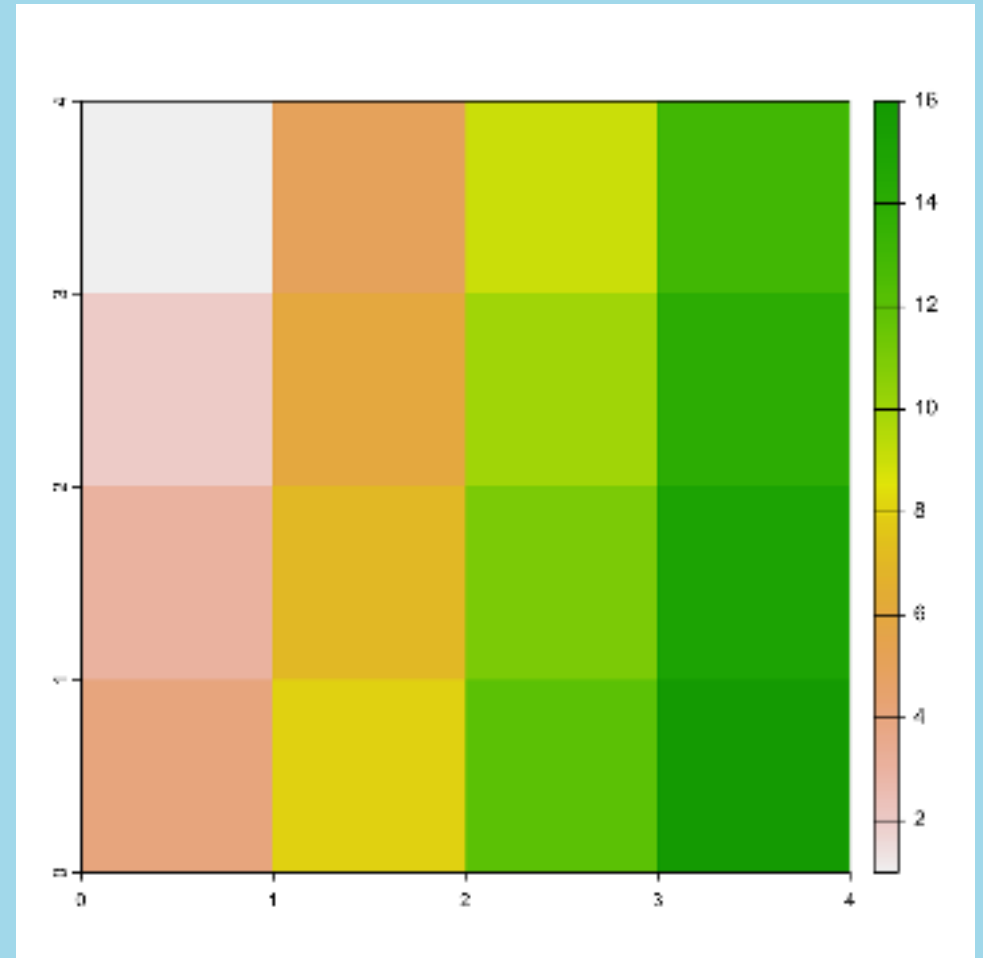
Rasters by Construction

```
1 mtx <- matrix(1:16, nrow=4)
2 mtx
```

```
      [,1] [,2] [,3] [,4]
[1,]     1     5     9    13
[2,]     2     6    10    14
[3,]     3     7    11    15
[4,]     4     8    12    16
```

```
1 rstr <- terra::rast(mtx)
2 rstr
```

```
class           : SpatRaster
dimensions      : 4, 4, 1 (nrow, ncol,
nlyr)
resolution      : 1, 1 (x, y)
extent          : 0, 4, 0, 4 (xmin,
xmax, ymin, ymax)
coord. ref.     :
source(s)       : memory
name            : lyr.1
min value       :      1
max value       :     16
```



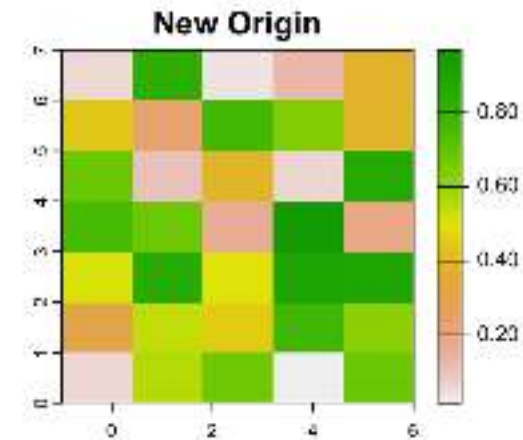
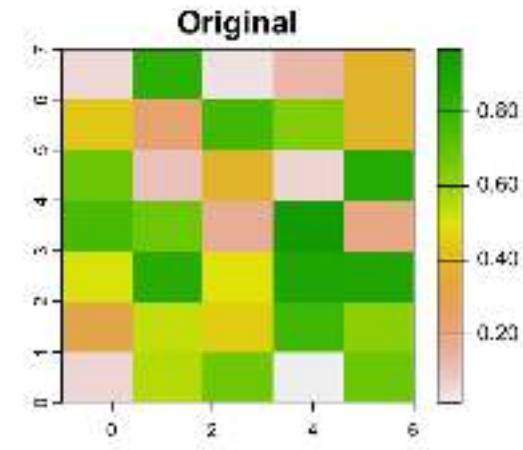
Rasters by Construction: Origin

- Origin defines the location of the intersection of the x and y axes

```
1 r <- rast(xmin=-4, xmax = 9.5, ncol=10, nrow=10)
2 r[] <- runif(ncell(r))
3 origin(r)
```

```
[1] 0.05 0.00
```

```
1 r2 <- r
2 origin(r2) <- c(2,2)
```



Rasters by Construction: Resolution

- Geometry is implicit; the spatial extent and number of rows and columns define the cell size
- **Resolution** (**res**) defines the length and width of an individual pixel

```
1 r <- rast(xmin=-4, xmax = 9.5,  
2          ncols=10)  
3 res(r)
```

```
[1] 1.35 1.00
```

```
1 r2 <- rast(xmin=-4, xmax = 5,  
2          ncols=10)  
3 res(r2)
```

```
[1] 0.9 1.0
```

```
1 r <- rast(xmin=-4, xmax = 9.5,  
2          res=c(0.5,0.5))  
3 ncol(r)
```

```
[1] 27
```

```
1 r2 <- rast(xmin=-4, xmax = 9.5,  
2          res=c(5,5))  
3 ncol(r2)
```

```
[1] 3
```

Predicates and measures in **terra**

Extending predicates

- **Predicates:** evaluate a logical statement asserting that a property is **TRUE**
- **terra** does not follow the same hierarchy as **sf** so a little trickier

Unary predicates in terra

- Can tell us qualities of a raster dataset
- Many similar operations for **SpatVector** class (note use of **.**)

predicate	asks...
is.lonlat	Does the object have a longitude/latitude CRS?
inMemory	is the object stored in memory?
is.factor	Are there categorical layers?
hasValues	Do the cells have values?

Unary predicates in **terra**

- **global**: tests if the raster covers all longitudes (from -180 to 180 degrees) such that the extreme columns are in fact adjacent
- **perhaps**: If TRUE and the crs is unknown, the method returns TRUE if the coordinates are plausible for longitude/latitude

```
1 r <- rast()  
2 is.lonlat(r)
```

```
[1] TRUE
```

```
1 is.lonlat(r, global=TRUE)
```

```
[1] TRUE
```

```
1 crs(r) <- ""  
2 is.lonlat(r)
```

```
[1] NA
```

```
1 is.lonlat(r, perhaps=TRUE, warn=FALSE)
```

```
[1] TRUE
```

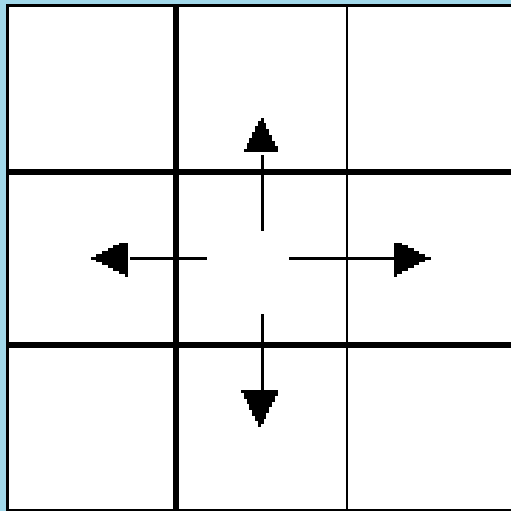
```
1 crs(r) <- "+proj=lcc +lat_1=48 +lat_2=33 +lon_0=-1  
2 is.lonlat(r)
```

```
[1] FALSE
```

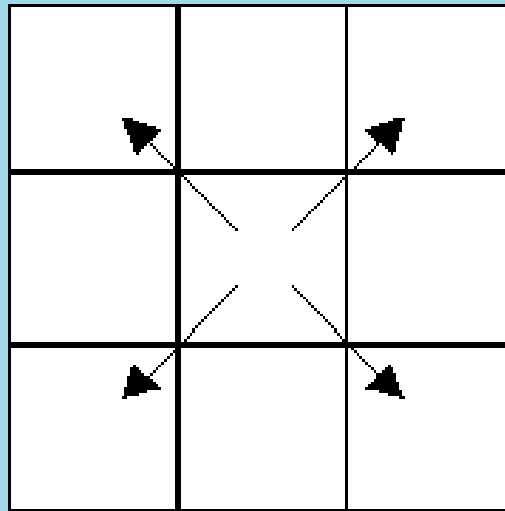
Binary predicates in **terra**

- Take exactly 2 inputs, return 1 matrix of cell locs where value is **TRUE**
- **adjacent**: identifies cells adjacent to a set of raster cells

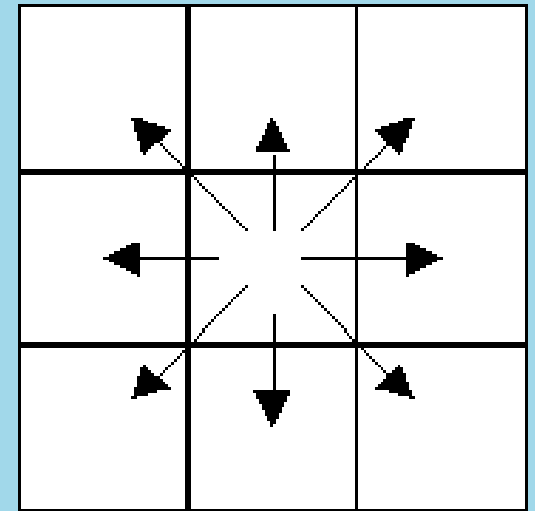
Rooks Case



Bishops Case



Queen's (Kings) Case



Unary measures in terra

- Slightly more flexible than sf
- One result for each layer in a stack

measure	returns
cellSize	area of individual cells
expanse	summed area of all cells
values	returns all cell values
ncol	number of columns
nrow	number of rows
ncell	number of cells
res	resolution
ext	minimum and maximum of x and y coords
origin	the origin of a SpatRaster
crs	the coordinate reference system
cats	categories of a categorical raster

Binary measures in **terra**

- Returns a matrix or **SpatRaster** describing the measure

measure	returns
distance	shortest distance to non-NA or vector object
gridDistance	shortest distance through adjacent grid cells
costDistance	Shortest distance considering cell-varying friction
direction	azimuth to cells that are not NA

