{raster} vs {terra}

{raster} vs {terra}

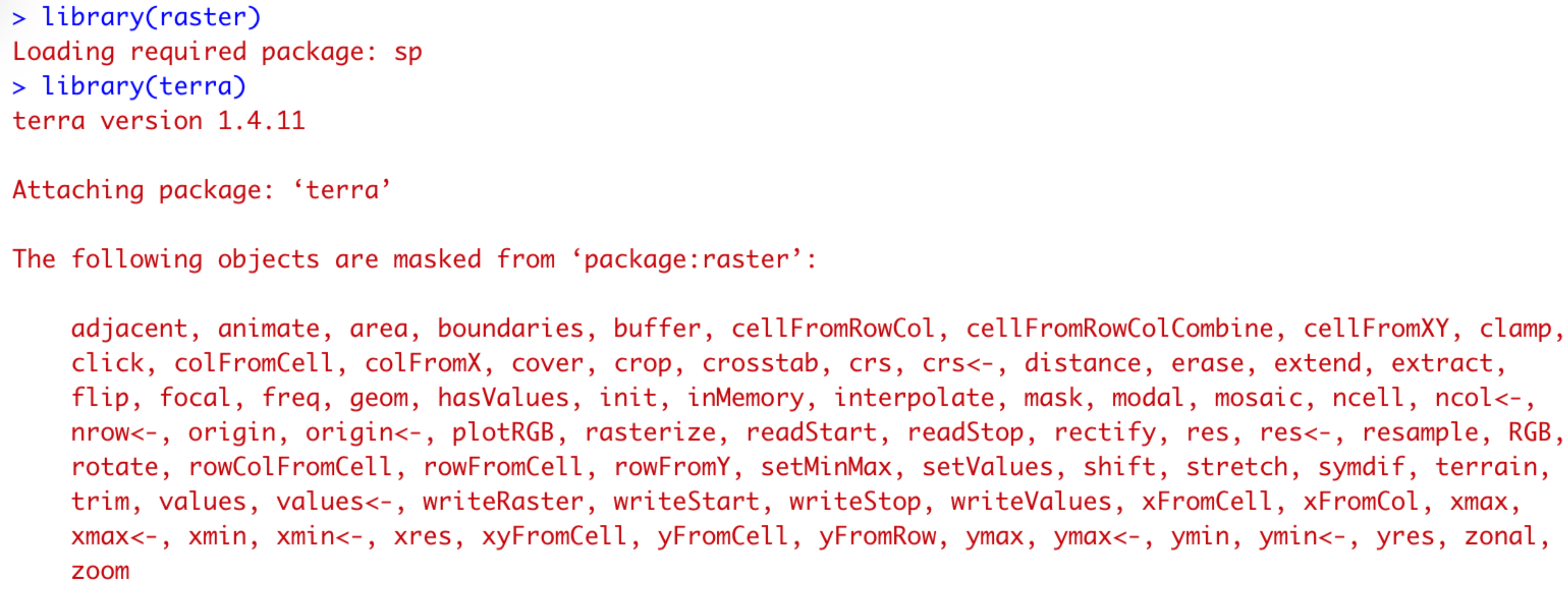
{raster} vs {terra}

Both of these packages are designed for working with single layer raster GIS datasets.

- {raster} has existed since 2010 and is extremely widely used.
- {terra} has only existed since 2020 and (as of 2022) is very novel
    - {mapview} is not yet capable of visualising {terra} objects

{raster} vs {terra}

It's difficult to combine both packages into the same script due to function masking.



```
> library(raster)
Loading required package: sp
> library(terra)
terra version 1.4.11

Attaching package: ‘terra’

The following objects are masked from ‘package:raster’:

    adjacent, animate, area, boundaries, buffer, cellFromRowCol, cellFromRowColCombine, cellFromXY, clamp,
    click, colFromCell, colFromX, cover, crop, crosstab, crs, crs<-, distance, erase, extend, extract,
    flip, focal, freq, geom, hasValues, init, inMemory, interpolate, mask, modal, mosaic, ncell, ncol<-,
    nrow<-, origin, origin<-, plotRGB, rasterize, readStart, readStop, rectify, res, res<-, resample, RGB,
    rotate, rowColFromCell, rowFromCell, rowFromY, setMinMax, setValues, shift, stretch, symdif, terrain,
    trim, values, values<-, writeRaster, writeStart, writeStop, writeValues, xFromCell, xFromCol, xmax,
    xmax<-, xmin, xmin<-, xres, xyFromCell, yFromCell, yFromRow, ymax, ymax<-, ymin, ymin<-, yres, zonal,
    zoom
```

{raster}, {terra} and the tidyverse

Neither of these packages are designed for a tidyverse workflow.

- We can't use `select()` and `filter()` to extract parts of these objects
- We can't modify/add parameters with `mutate()`

{raster} and {terra} examples

In the visualisation sections of the course I'll provide code samples in the `data/` folder for manipulating both `{raster}` and `{terra}` objects.