Welcome to

instats

The Session Will Begin Shortly

START

Spatial Data Analysis and Visualization in R

Session 14: Working with Raster Data in R Using the terra package

instats

Content

- Learn how to read, explore, and analyze raster data using **terra**
- Work with elevation and boundary data for Zion National Park
- Practice cropping, masking, extracting, and plotting
- Read and write raster data formats
- Resample and reproject raster data

Required Packages

```r
library(tmap)
library(terra)
library(sf)
library(spDataLarge)
```

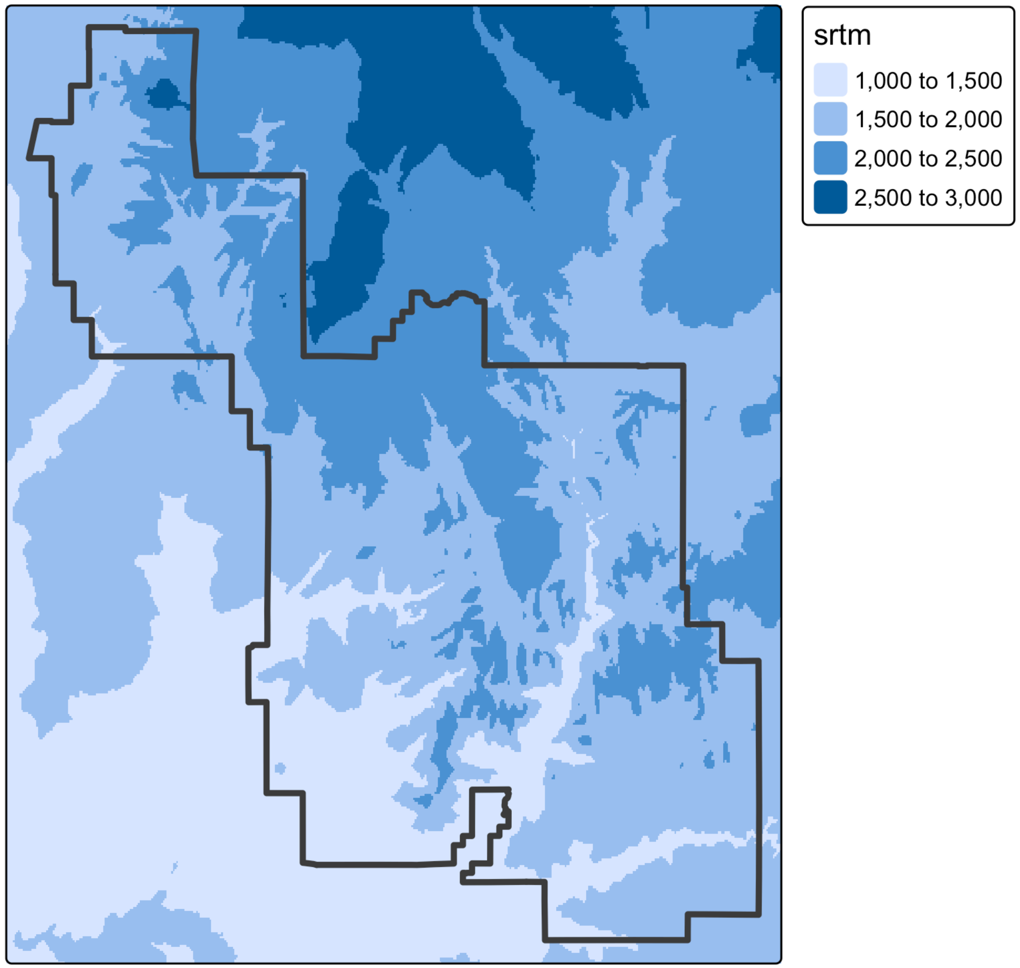
Load Zion Data from spDataLarge

```
srtm = rast(system.file("raster/srtm.tif", package = "spDataLarge"))
zion = read_sf(system.file("vector/zion.gpkg", package = "spDataLarge"))
zion = st_transform(zion, st_crs(srtm))
```

Explore the data

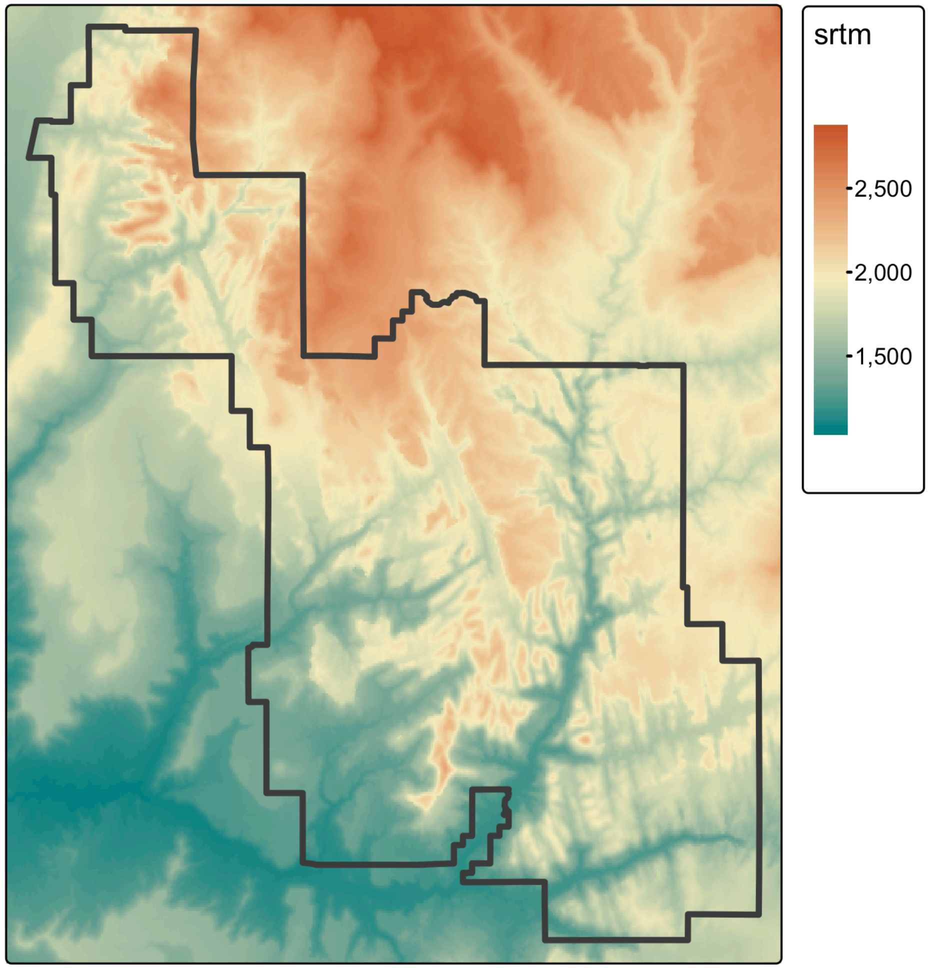
- Elevation model of Zion National Park

```
qtm(srtm) + qtm(zion, fill = NULL, lwd = 3)
```



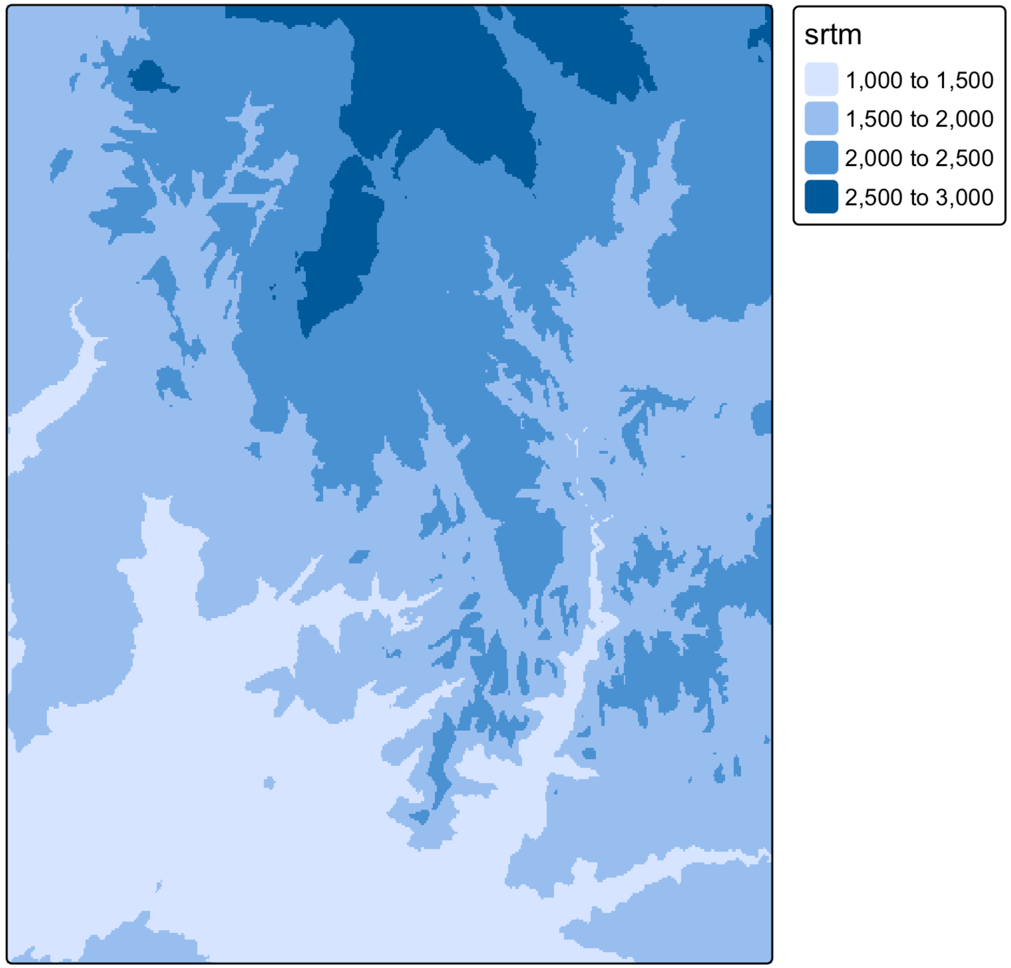
Fine tune the plot

```
tm_shape(srtm) + tm_raster(col = "srtm", col.scale =  tm_scale_continuous(values = "carto.geyser"))
    qtm(zion, fill = NULL, lwd = 3)
```



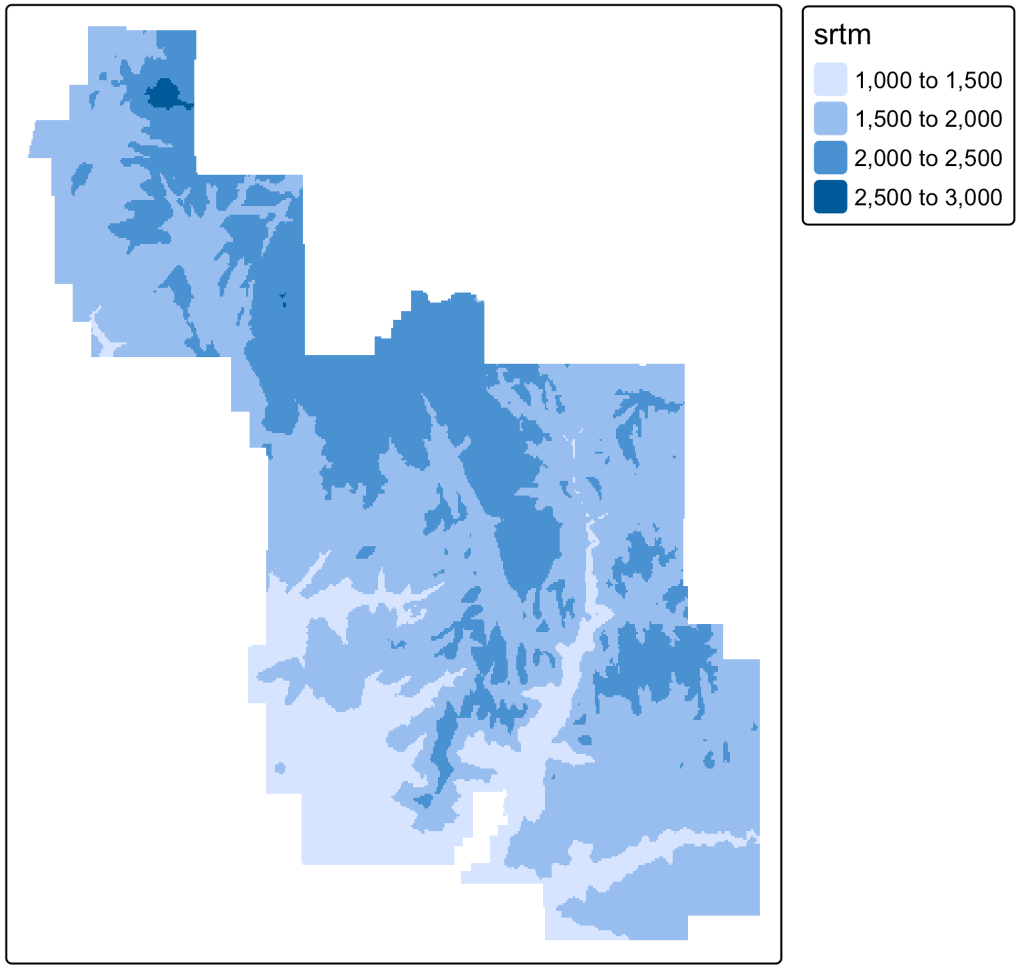
Crop Raster to Park Extent

```
srtm_cropped = crop(srtm, zion)

qtm(srtm_cropped)
```



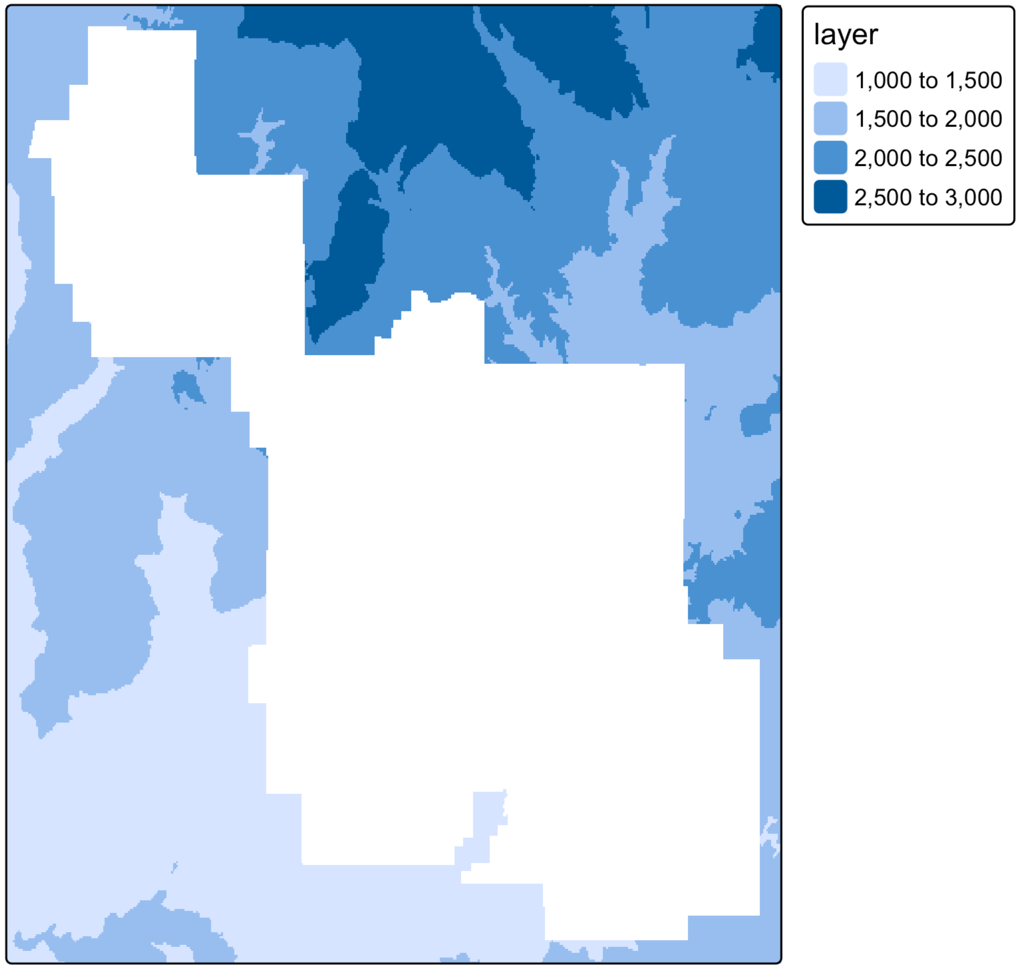
Mask Raster to Polygon

```
srtm_masked = mask(srtm, zion)

qtm(srtm_masked)
```

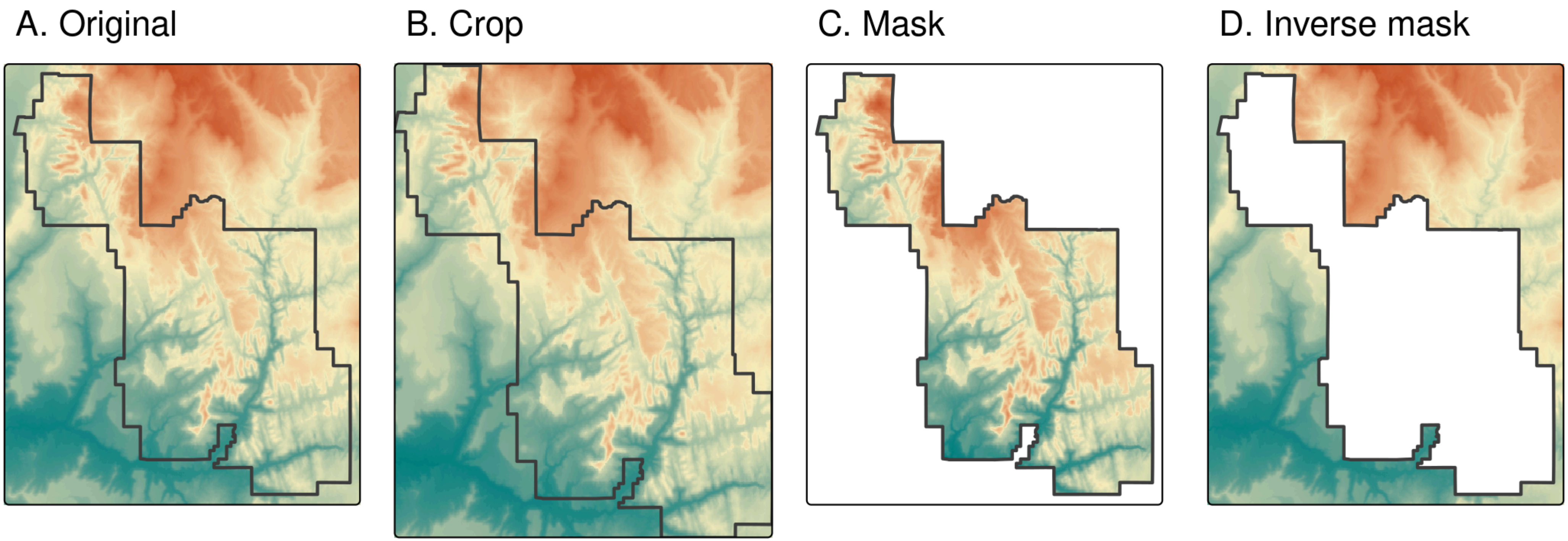


Invert Mask Raster to Polygon

```
srtm_inv_masked = mask(srtm, zion, inverse = TRUE)

qtm(srtm_inv_masked)
```

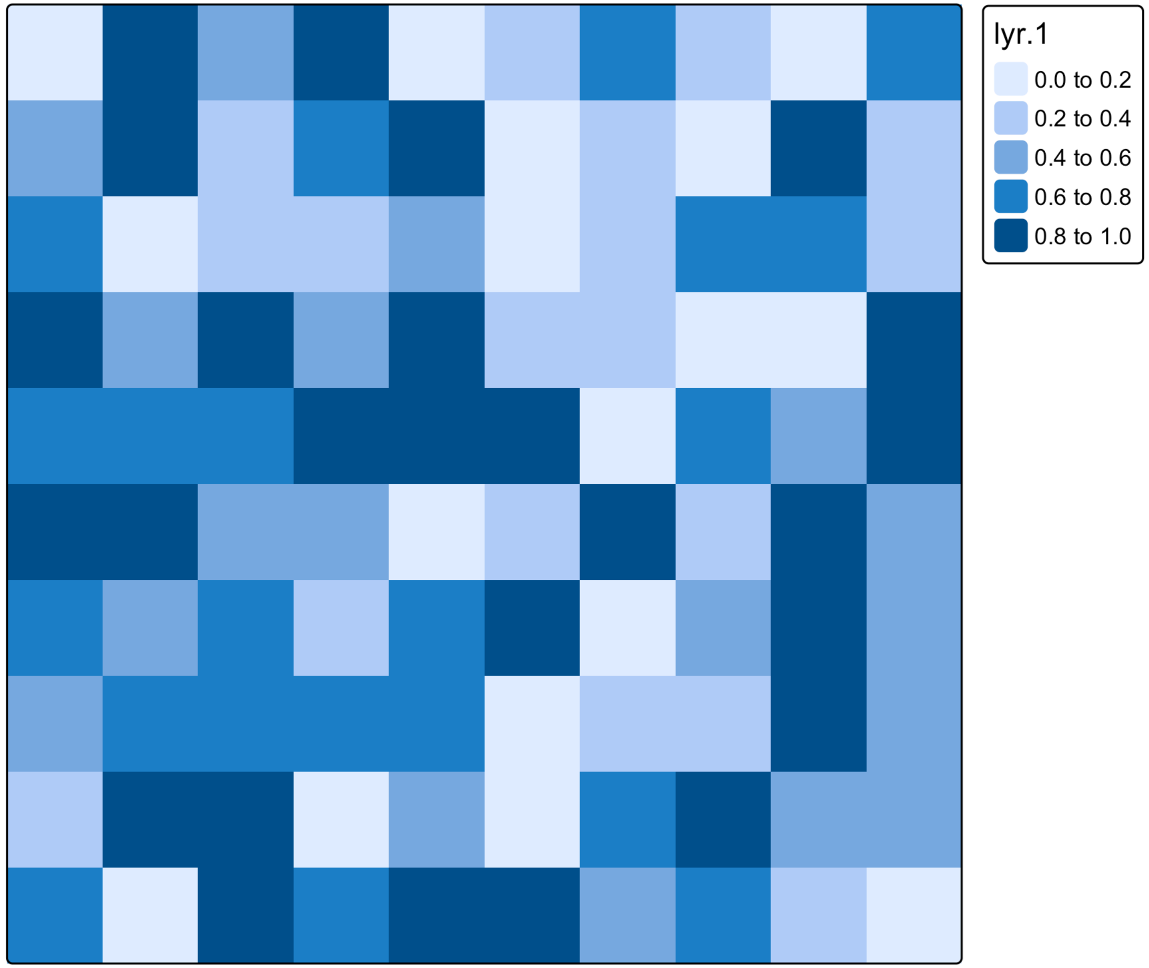


Compared



Create Raster from Matrix (terra)

```
m <- matrix(runif(100), 10, 10)
rst <- rast(m)
ext(rst) <- ext(0, 10, 0, 10)
crs(rst) <- "EPSG:4326"

qtm(rst)
```


lyr.1
0.0 to 0.2
0.2 to 0.4
0.4 to 0.6
0.6 to 0.8
0.8 to 1.0

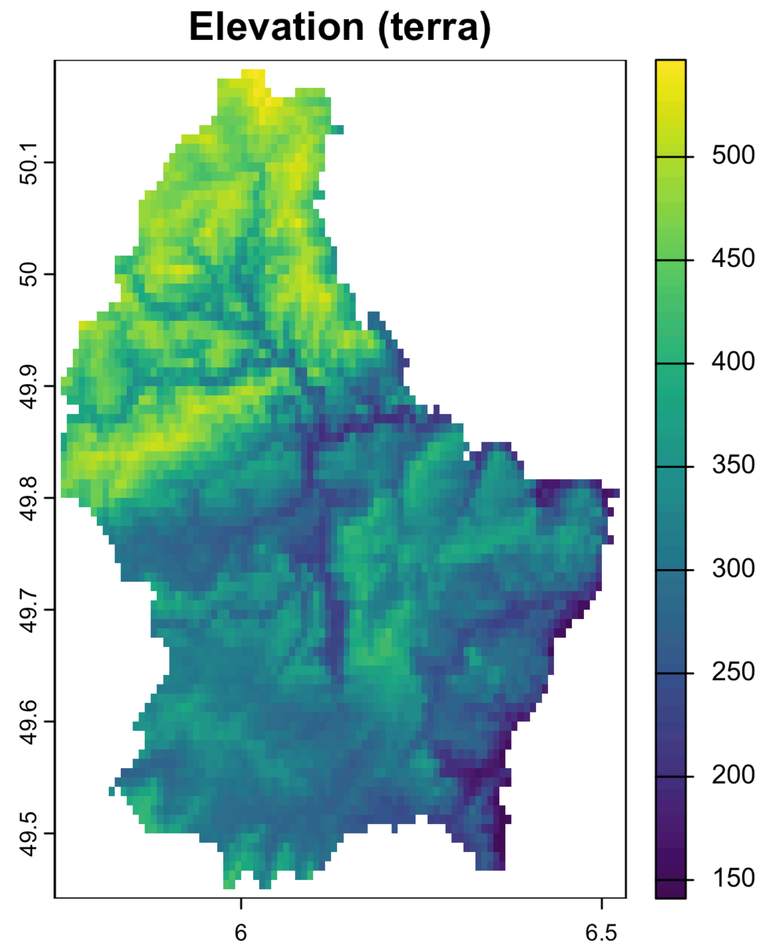
Reading and writing with terra

Common Raster Formats

- `.tif` – GeoTIFF (most common)
- `.asc` – ASCII grid
- `.img` – ERDAS Imagine
- `.grd`/`.gri` – Raster legacy format
- `.nc` – NetCDF (time-series/climate data)

Read a GeoTIFF with terra

```r
r = rast(system.file("ex/elev.tif", package = "terra"))
plot(r, main = "Elevation (terra)")
```



Raster Info and Metadata

```r
r
```

```
class       : SpatRaster
dimensions  : 90, 95, 1  (nrow, ncol, nlyr)
resolution  : 0.008333333, 0.008333333  (x, y)
extent      : 5.741667, 6.533333, 49.44167, 50.19167  (xmin, xmax, ymin, ymax)
coord. ref. : lon/lat WGS 84 (EPSG:4326)
source      : elev.tif
name        : elevation
min value   :       141
max value   :       547
```

```r
crs(r)
```

```
[1] "GEOGCRS[\"WGS 84\",\n    ENSEMBLE[\"World Geodetic System 1984 ensemble\",\n        MEMBER[\"World Geodetic System 1984 (Transit)\"],\n        MEMBER[\"World Geodetic System 1984 (G730)\"],\n        MEMBER[\"World Geodetic System 1984 (G873)\"],\n        MEMBER[\"World Geodetic System 1984 (G1150)\"],\n        MEMBER[\"World Geodetic System 1984 (G1674)\"],\n        MEMBER[\"World Geodetic System 1984 (G1762)\"],\n        MEMBER[\"World Geodetic System 1984 (G2139)\"],\n        MEMBER[\"World Geodetic System 1984 (G2296)\"],\n        ELLIPSOID[\"WGS 84\",6378137,298.257223563,\n            LENGTHUNIT[\"metre\",1]],\n        ENSEMBLEACCURACY[2.0]],\n    PRIMEM[\"Greenwich\",0,\n        ANGLEUNIT[\"degree\",0.0174532925199433]],\n    CS[ellipsoidal,2],\n        AXIS[\"geodetic latitude (Lat)\",north,\n            ORDER[1],\n            ANGLEUNIT[\"degree\",0.0174532925199433]],\n        AXIS[\"geodetic longitude (Lon)\",east,\n            ORDER[2],\n            ANGLEUNIT[\"degree\",0.0174532925199433]],\n    USAGE[\n        SCOPE[\"Horizontal component of 3D system.\"],\n        AREA[\"World.\"],\n        BBOX[-90,-180,90,180]],\n    ID[\"EPSG\",4326]]"
```

Raster Info and Metadata (cont)

```r
ext(r)
```

```
SpatExtent : 5.74166666666667, 6.53333333333333, 49.4416666666667, 50.1916666666667 (xmin, xmax,
ymin, ymax)
```

```r
res(r)
```

```
[1] 0.008333333 0.008333333
```

```r
sources(r)
```

```
[1] "/Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/terra/ex/elev.tif"
```

Write to GeoTIFF (terra)

```
writeRaster(r, "elevation_output.tif", overwrite = TRUE)
```

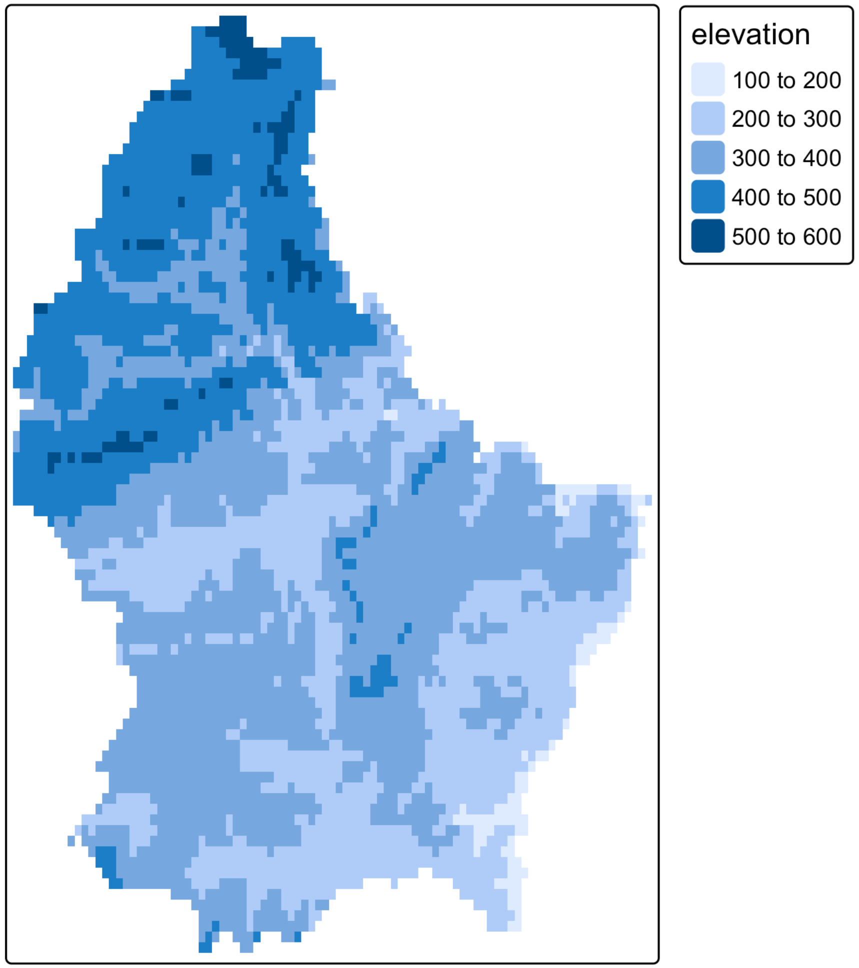
Write to ASCII and IMG

```
writeRaster(r, "elevation.asc", overwrite = TRUE)
writeRaster(r, "elevation.img", filetype = "HFA", overwrite = TRUE)
```

Downsampling, Warping, and Transforming Raster Data with terra

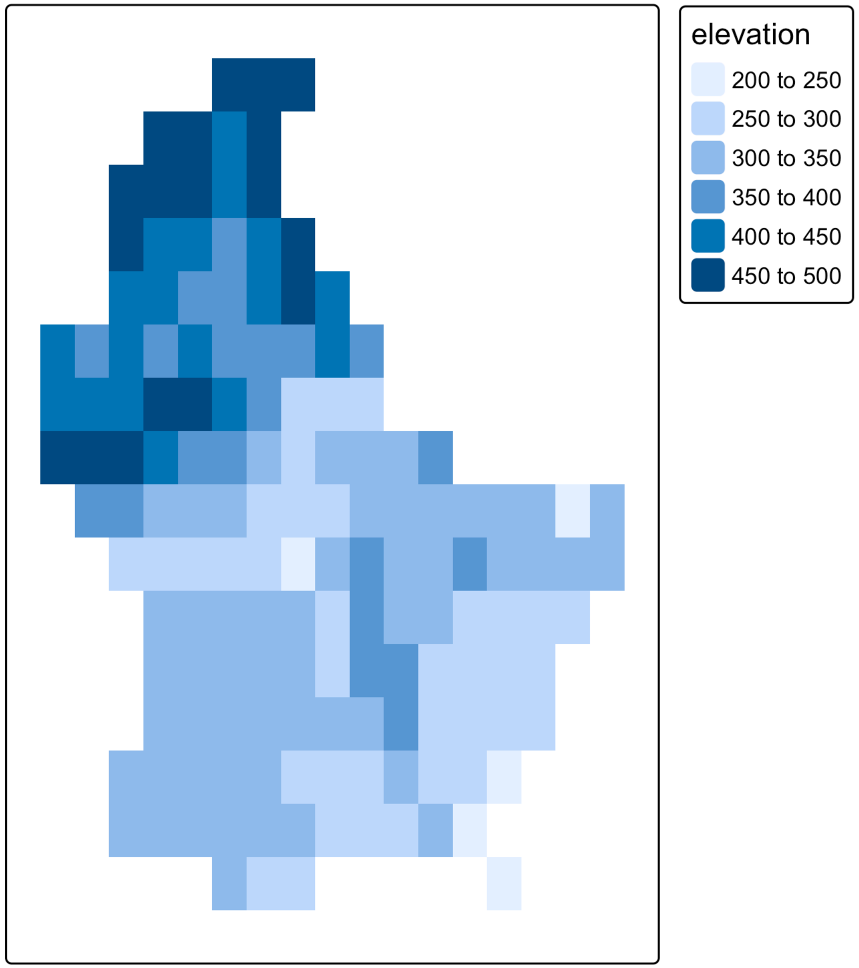
Load a Raster with terra

```r
r <- rast(system.file("ex/elev.tif", package = "terra"))
qtm(r)
```



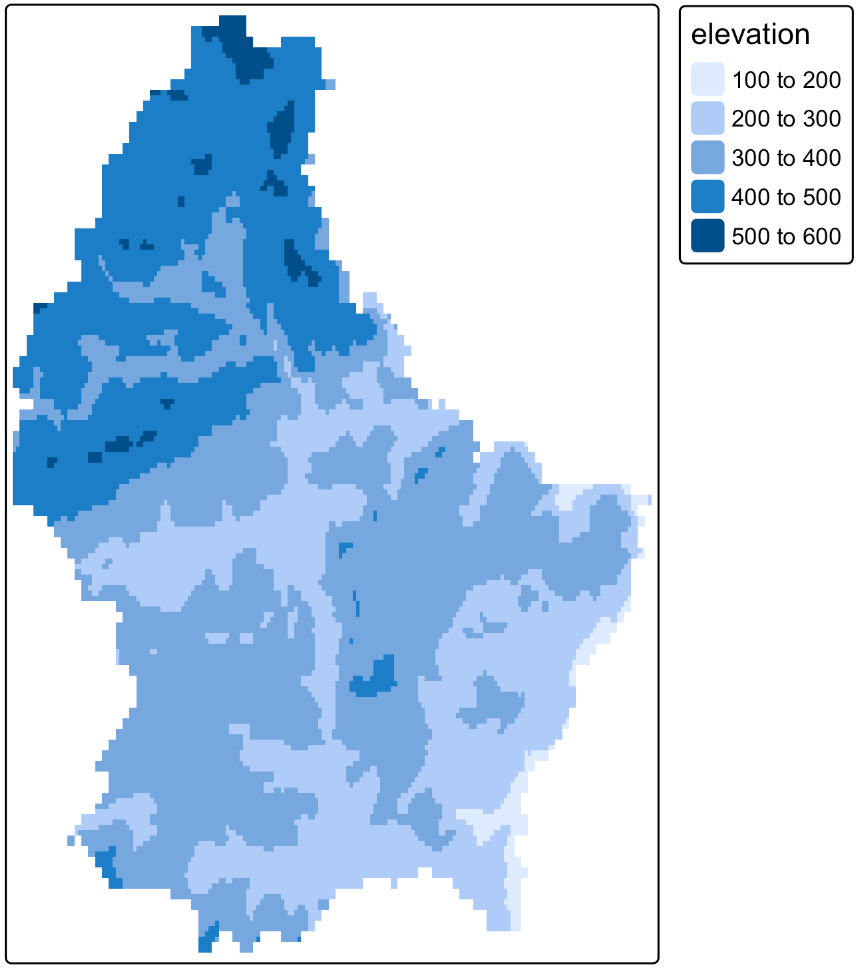
Downsample (Aggregate)

```r
r_lowres <- aggregate(r, fact = 5, fun = mean)
qtm(r_lowres)
```



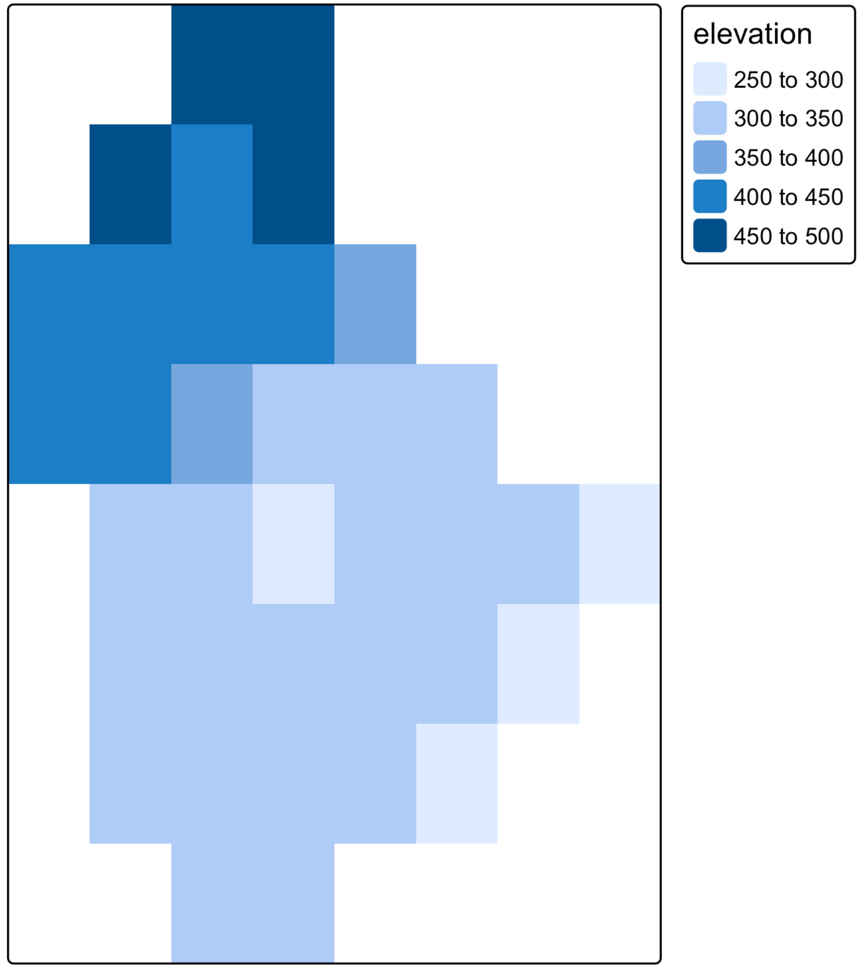
Change Resolution (Disaggregate)

```
r_hires <- disagg(r, fact = 2, method = "bilinear")
qtm(r_hires)
```



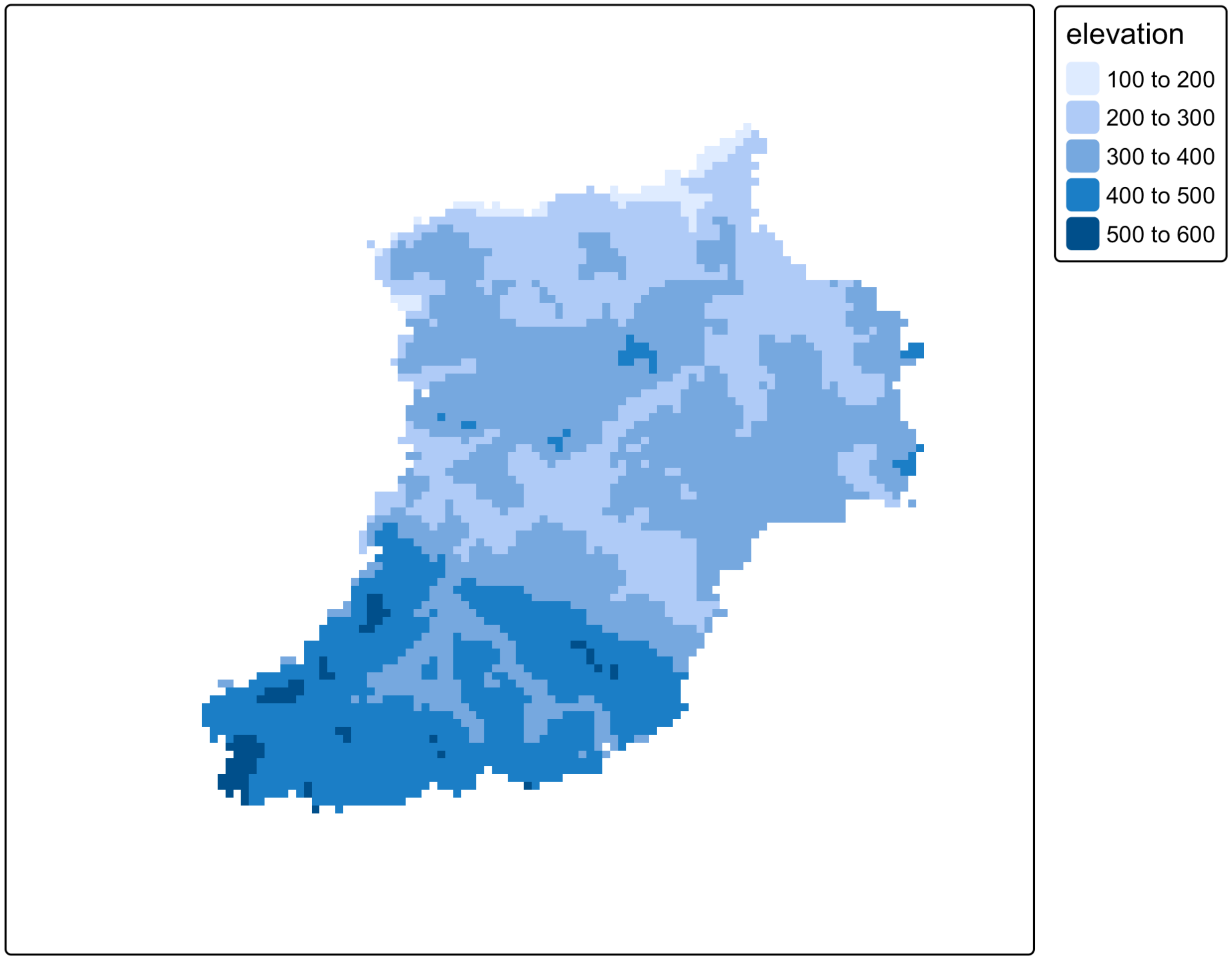
Warp to a New Raster Template

```
# Create template
template <- rast(ext(r), resolution = .1)

# Warp using bilinear interpolation
r_warp <- resample(r, template, method = "bilinear")
qtm(r_warp)
```


elevation
250 to 300
300 to 350
350 to 400
400 to 450
450 to 500

Transform CRS with terra

```r
r_reproj <- project(r, "EPSG:32612")
qtm(r_reproj)
```



Summary

- terra is optimized for performance
- Excellent for:
  - → large rasters
  - → format support
  - → simple workflows
- Focus on spatial alignment, extents, and projections

STOP