

Operations on Raster Data I

HES 505 Fall 2023: Session 12

Carolyn Koehn



Today's Plan

Objectives

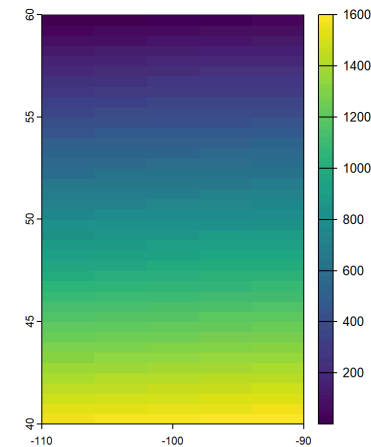
- By the end of today, you should be able to:
 - Align rasters for spatial processing
 - Adjust the resolution of raster data
 - Combine (or reduce) rasters to match the extent of your analysis

Aligning rasters for spatial processing

Projecting raster data

- Transformation from lat/long to planar CRS involves some loss of precision
- New cell values estimated using overlap with original cells
- Interpolation for continuous data, nearest neighbor for categorical data
- Equal-area projections are preferred; especially for large areas

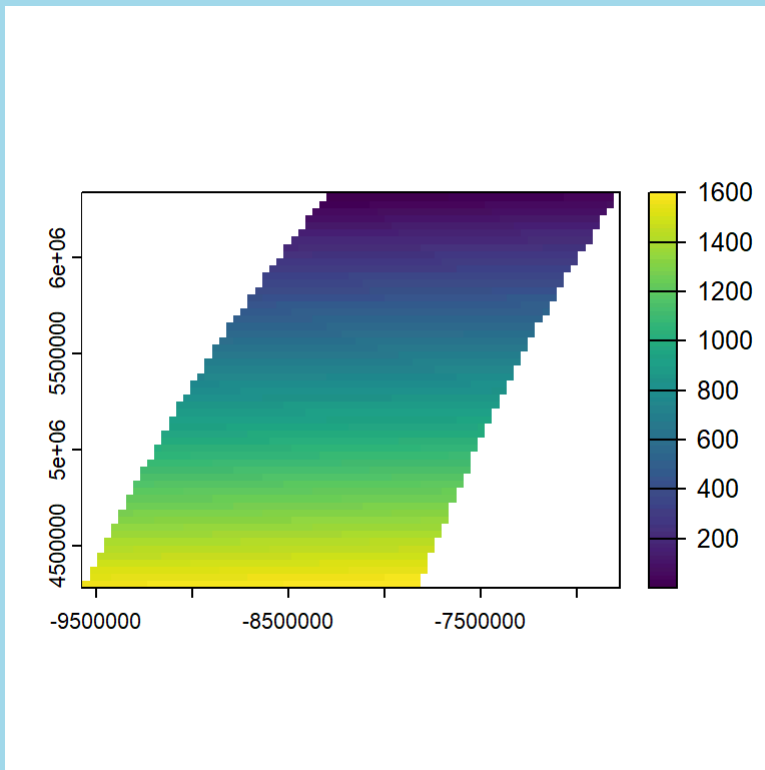
```
1 library(sf)
2 library(terra)
3 library(spDataLarge)
4 r <- rast(xmin=-110, xmax=-90, ymin=40, ymax=60)
5 values(r) <- 1:ncell(r)
6 plot(r)
```



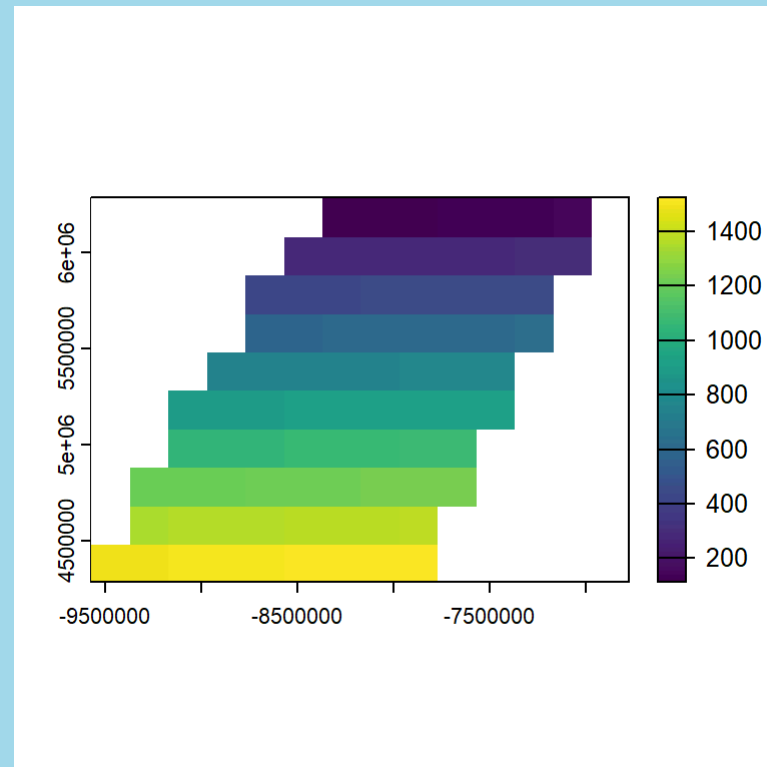
Projecting raster data

- simple method; alignment not guaranteed
- providing a template to ensure alignment

```
1 newcrs <- "+proj=robin +da  
2 pr1 <- terra::project(r, r  
3 plot(pr1)
```

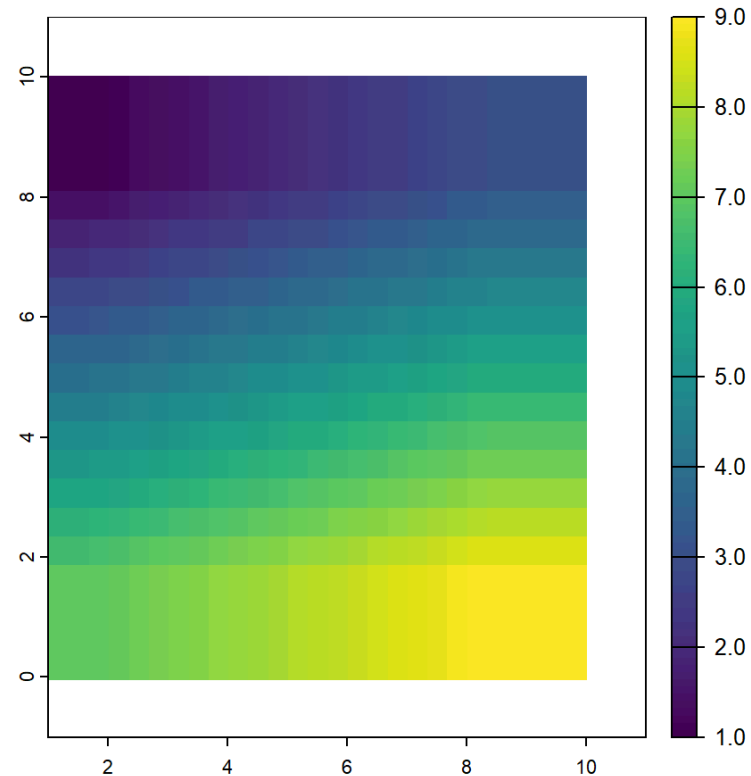
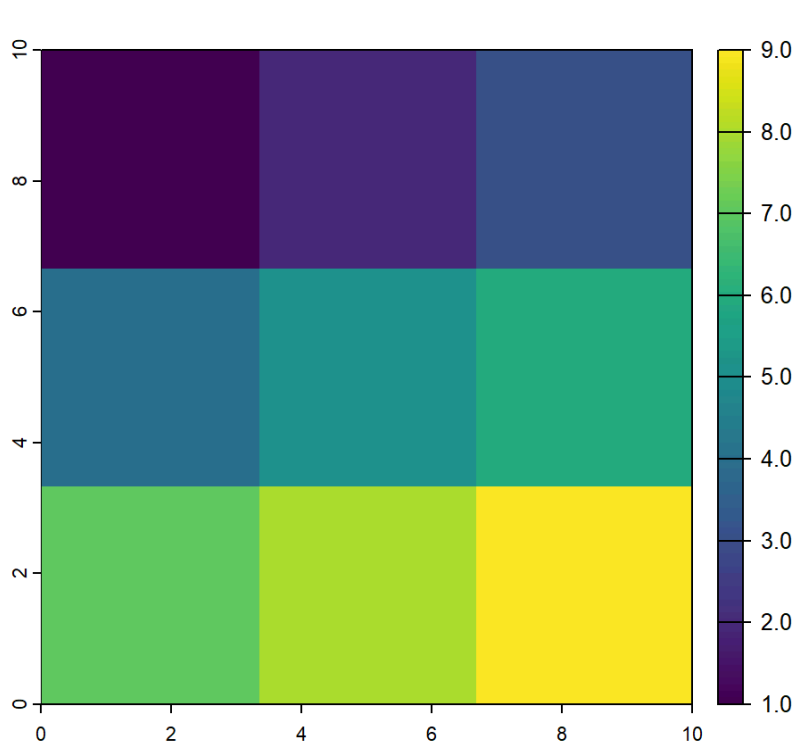


```
1 x <- rast(pr1)  
2 # Set the cell size  
3 res(x) <- 200000  
4 pr3 <- terra::project(r, x  
5 plot(pr3)
```



Aligning Data: **resample**

```
1 r <- rast(nrow=3, ncol=3, xmin=0, xmax=10, ymin=0, ymax=10)
2 values(r) <- 1:ncell(r)
3 s <- rast(nrow=25, ncol=30, xmin=1, xmax=11, ymin=-1, ymax=11)
4 x <- resample(r, s, method="bilinear")
```



Adjusting resolution

Downscaling and Upscaling

- Aligning data for later analysis
- Remembering *scale*
- Thinking about support

Changing resolutions

- **aggregate**, **disaggregate**, **resample** allow changes in cell size
- **aggregate** requires a function (e.g., **mean()** or **min()**) to determine what to do with the grouped values
- **resample** allows changes in cell size **and** shifting of cell centers (slower)

Changing resolutions: aggregate

• **Aggregating** data from a lower resolution to a higher resolution

• **Downsampling** data from a higher resolution to a lower resolution

• **Upsampling** data from a lower resolution to a higher resolution

• **Interpolation** data from a lower resolution to a higher resolution

• **Resampling** data from a higher resolution to a lower resolution

• **Resampling** data from a lower resolution to a higher resolution

• **Resampling** data from a higher resolution to a lower resolution

• **Resampling** data from a lower resolution to a higher resolution

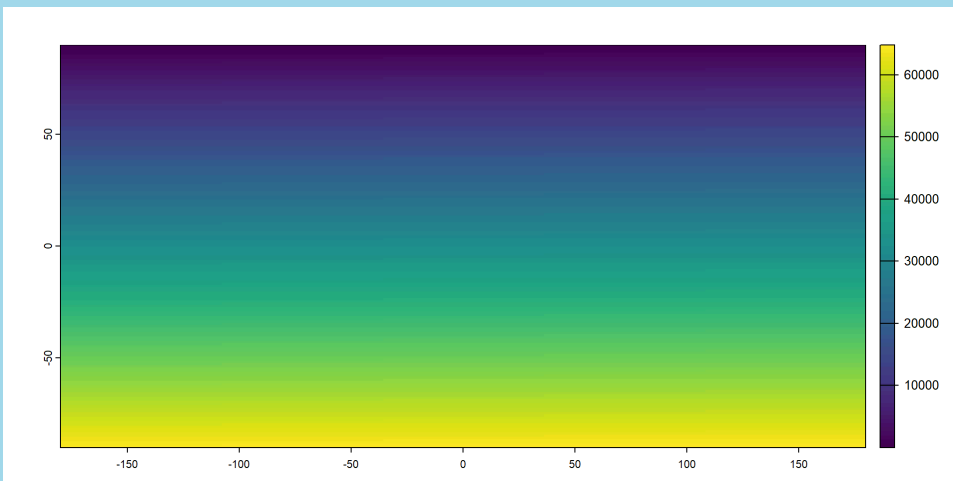
• **Resampling** data from a higher resolution to a lower resolution

• **Resampling** data from a lower resolution to a higher resolution

```
1 r <- rast()  
2 r
```

```
class      : SpatRaster  
dimensions : 180, 360, 1 (nrow,  
ncol, nlyr)  
resolution : 1, 1 (x, y)  
extent     : -180, 180, -90, 90  
(xmin, xmax, ymin, ymax)  
coord. ref.: lon/lat WGS 84 (CRS84)  
(OGC:CRS84)
```

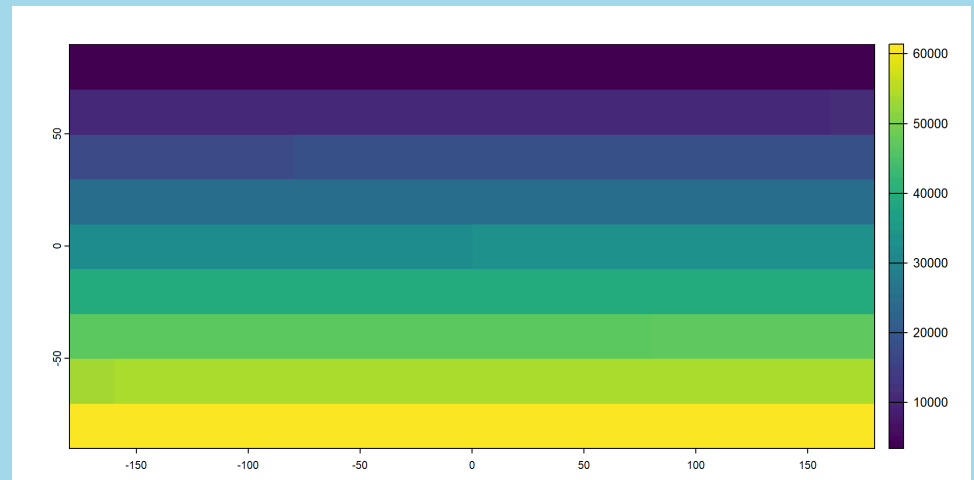
```
1 values(r) <- 1:ncell(r)  
2 plot(r)
```



```
1 ra <- aggregate(r, 20)  
2 ra
```

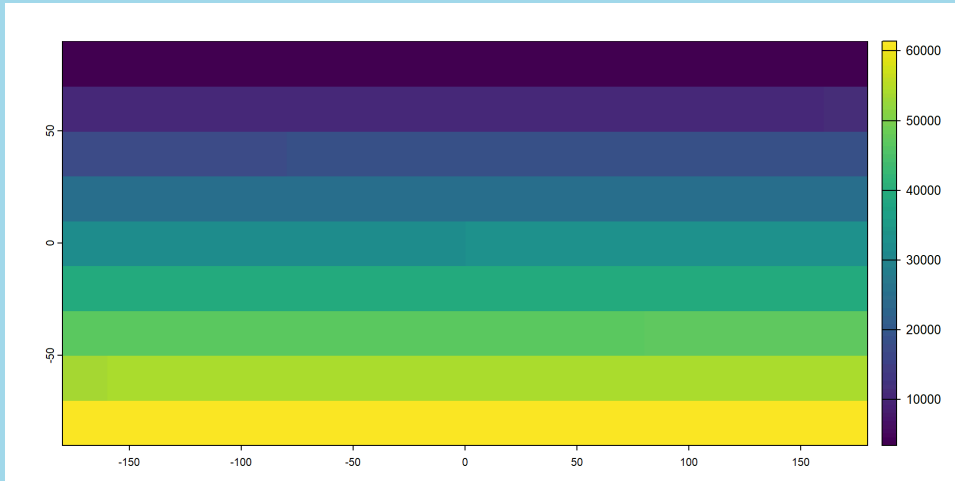
```
class      : SpatRaster  
dimensions : 9, 18, 1 (nrow, ncol,  
nlyr)  
resolution : 20, 20 (x, y)  
extent     : -180, 180, -90, 90  
(xmin, xmax, ymin, ymax)  
coord. ref.: lon/lat WGS 84 (CRS84)  
(OGC:CRS84)  
source(s)  : memory  
name       : lyr.1  
min value  : 3430.5  
max value  : 61370.5
```

```
1 plot(ra)
```



Changing resolutions: disagg

```
1 ra <- aggregate(r, 20)
2 plot(ra)
```

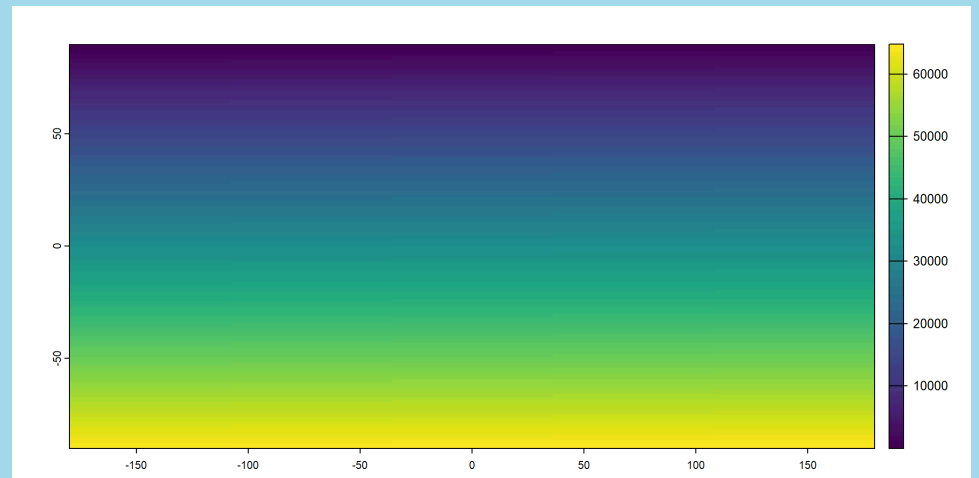


```
1 rd <- disagg(r, 20)
```

```
1 rd
```

```
class      : SpatRaster
dimensions : 3600, 7200, 1  (nrow,
ncol, nlyr)
resolution : 0.05, 0.05  (x, y)
extent      : -180, 180, -90, 90
(xmin, xmax, ymin, ymax)
coord. ref. : lon/lat WGS 84 (CRS84)
(OGC:CRS84)
source      :
spat_451c35f722d0_17692.tif
name        : lyr.1
min value   :      1
max value   : 64800
```

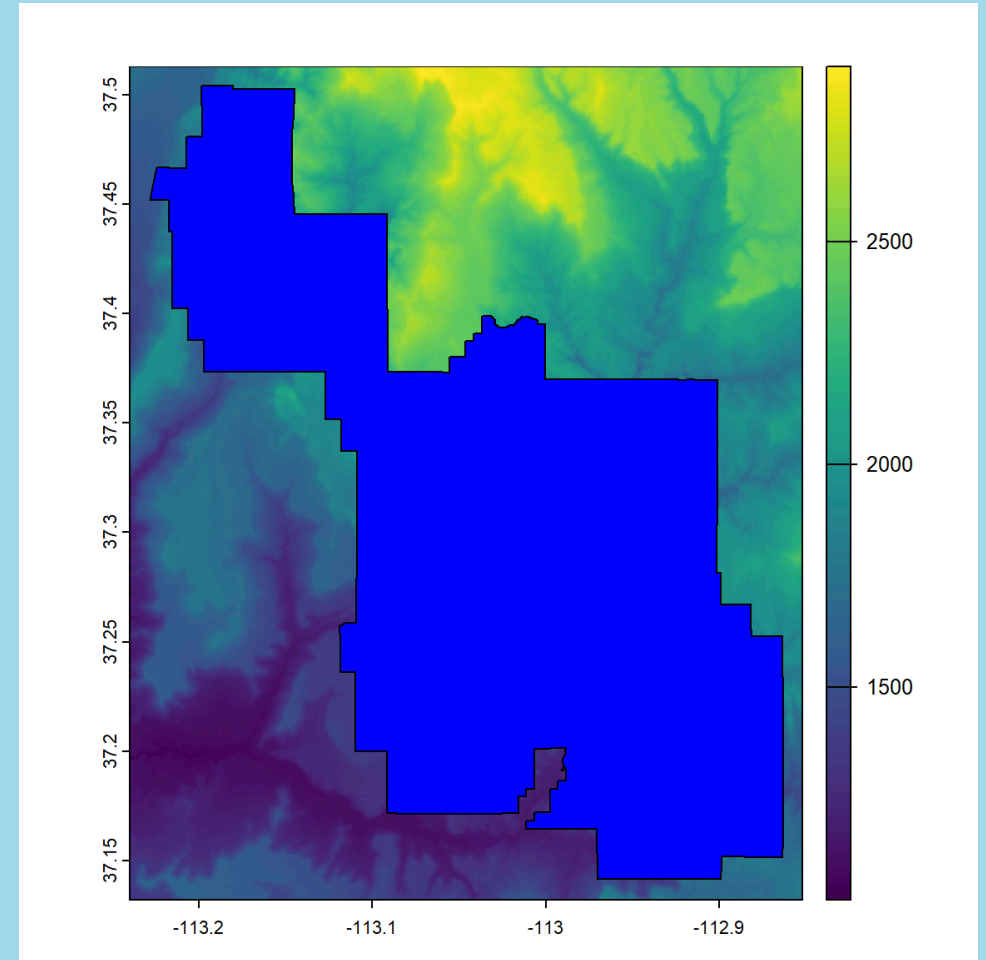
```
1 plot(rd)
```



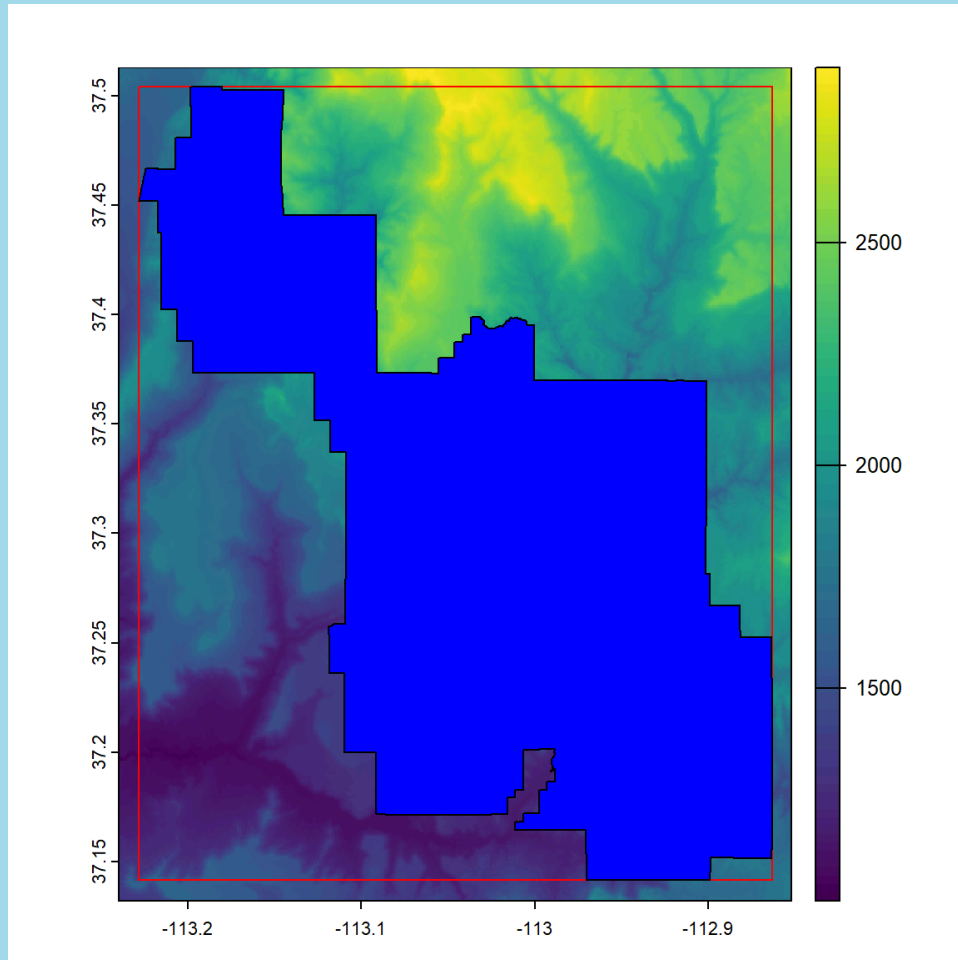
Modifying the Extent

Dealing with Different Extents

- Raster extents often larger than our analysis
- Reducing memory and computational resources
- Making attractive maps



Using `terra::crop()`



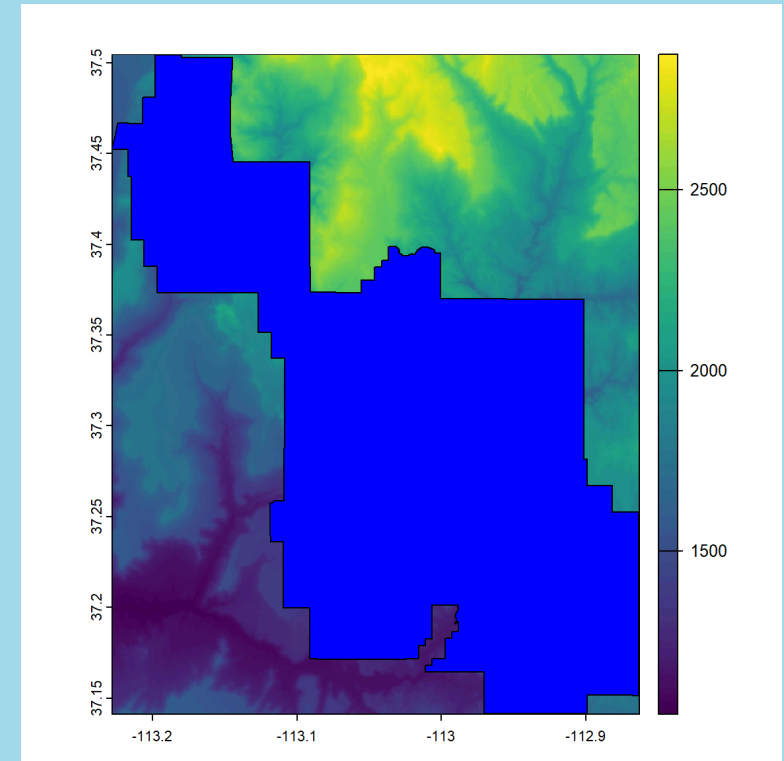
- Coordinate Reference System must be the same for both objects
- Crop is based on the (converted) **SpatExtent** of the 2nd object
- **snap** describes how **y** will be aligned to the raster
- Returns all data within the extent

Using `terra::crop()`

```
1 library(sf)
2 library(terra)
3 library(spDataLarge)
4 srtm = rast(system.file("raster/srtm.tif", package =
5 zion = read_sf(system.file("vector/zion.gpkg", package =
6 zion = st_transform(zion, crs(srtm))
7
8 crs(srtm) == crs(zion)
```

```
[1] TRUE
```

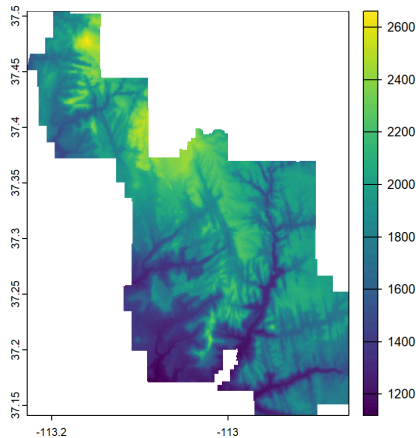
```
1 srtm.crop <- crop(x=srtm, y=zion, snap="near")
```



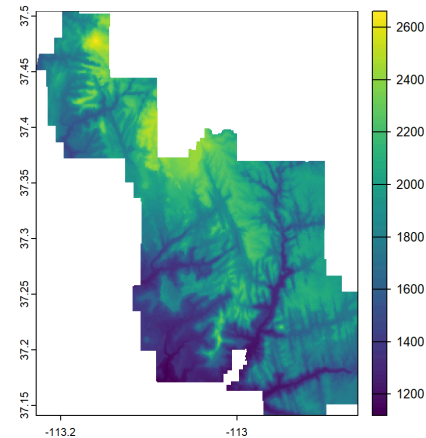
Using `mask()`

- Often want to get rid of all values outside of vector
- Can set `mask=TRUE` in `crop()` (y must be `SpatVector`)
- Or use `mask()`

```
1 srtm.crop.msk <- crop(x=srtm, y=srtm.poly, mask=TRUE)  
2 plot(srtm.crop.msk)
```



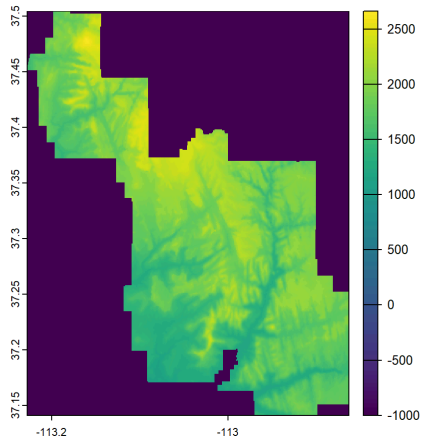
```
1 srtm.msk <- mask(srtm.crop, srtm.poly)  
2 plot(srtm.msk)
```



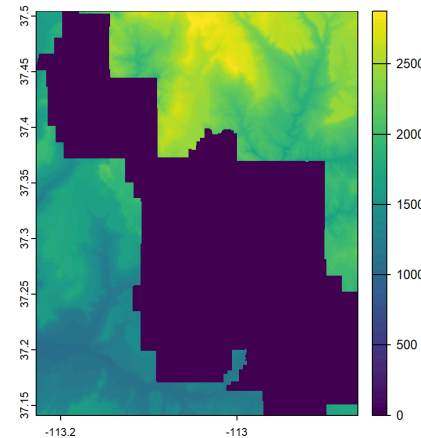
Using `mask()`

- Allows more control over what the mask does
- Can set `maskvalues` and `updatevalues` to change the resulting raster
- Can also use `inverse` to mask out the vector

```
1 srtm.msk <- mask(srtm.crop  
2 plot(srtm.msk)
```



```
1 srtm.msk <- mask(srtm.crop  
2 plot(srtm.msk)
```



Extending boundaries

- Vector slightly larger than raster
- Especially when using buffered datasets
- Can use `extend`
- Not exact; depends on `snap()`

Practice

`mosaic` is a function that combines adjacent rasters, but they need to have the same origin and resolution. Let's practice preparing some rasters for a mosaic.

1. Load wildfire hazard data from the `rasterexample` folder for OR and ID: `Copy of CRPS_OR.tif` and `Copy of CRPS_ID.tif`.
2. These rasters have a fine resolution that will make our calculations slow. Transform them to have a resolution of 900 km.
3. Do the rasters have the same CRS, origin, resolution, and extent? Check this with `==`.
4. Use new functions from this lecture to align the properties mentioned in #3 (plot often to check your work). Why not use `project`?
5. `mosaic` the two rasters together.