Rasters in R

GEOG 450/650 (A. Bevington)

2025-01-21

```
# Set up the environment by clearing any existing objects and loading required libraries.
rm(list=ls())
library(terra)
library(sf)
library(tidyterra)
library(mapview)
library(tidyverse)

# Set the working directory to the directory of the R Markdown file.
knitr::opts_knit$set(root.dir = dirname(rstudioapi::getSourceEditorContext()$path),
                     fig.width = 6, fig.height = 4, dpi = 72)
# Example: Uncomment these lines if you need to source a script or set a specific working directory.
# source("Week_03_2025-01-21_Tue_getImg.R")
# setwd("lectures")
```

Intro

Read Raster

We start by loading raster and vector data using the `terra` and `sf` packages. The rasters represent different spectral bands, and the vector file defines an area of interest (AOI).

```
B03 <- terra::rast("Week_3_2025-01-21_Tue_data/B03.tif")
B04 <- terra::rast("Week_3_2025-01-21_Tue_data/B04.tif")
B08 <- terra::rast("Week_3_2025-01-21_Tue_data/B08.tif")
B11 <- terra::rast("Week_3_2025-01-21_Tue_data/B11.tif")
aoi <- st_read("Week_3_2025-01-21_Tue_data/aoi.gpkg")
```

```
## Reading layer `aoi' from data source
##   `G:\My Drive\6_TEACHING\UNBC\2025_GEOG_450\lectures\Week_3_2025-01-21_Tue_data\aoi.gpkg'
##   using driver `GPKG'
## Simple feature collection with 1 feature and 0 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: -118.8305 ymin: 33.86802 xmax: -118.3959 ymax: 34.23278
## Geodetic CRS:  WGS 84
```

Inspect

Examine the raster data for metadata (e.g., coordinate reference system, dimensions, resolution) and display them interactively on a map.

```r
print(B11)
```

```
## class       : SpatRaster
## dimensions  : 2022, 2012, 1  (nrow, ncol, nlyr)
## resolution  : 20, 20  (x, y)
## extent      : 330980, 371220, 3748700, 3789140  (xmin, xmax, ymin, ymax)
## coord. ref. : WGS 84 / UTM zone 11N (EPSG:32611)
## source      : B11.tif
## name        : T11SLT_20250112T183731_B11_20m
## min value   :                            978
## max value   :                          16164
```

```r
mapview(B11, maxpixels = 1000, col.regions = RColorBrewer::brewer.pal(9, "Spectral")) +
  mapview(aoi)
```

```
## Number of pixels is above 1000.Only about 1000 pixels will be shown.
## You can increase the value of `maxpixels` to 4068264 to avoid this.
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed, please
```

Explore additional raster properties such as resolution, number of layers, and data type.

```r
crs(B08)
```

```
## [1] "PROJCRS[\"WGS 84 / UTM zone 11N\",\n    BASEGEOGCRS[\"WGS 84\",\n        DATUM[\"World Geodetic
```

```r
ncell(B08)
```

```
## [1] 16273012
```

```r
dim(B08)
```

```
## [1] 4046 4022    1
```

```r
res(B08)
```

```
## [1] 10 10
```

```r
nlyr(B08)
```

```
## [1] 1
```

```r
datatype(B08)
```

```
## [1] "INT2U"
```

```r
#compareGeom(B08,B04)
#compareGeom(B11,B04)
```

Combine Bands

False Color Infrared

Combine bands to create a multi-layer raster. Here, we combine NIR (B08), Red (B04), and Green (B03) bands to create a false-color infrared image.

```r
fcir <- c(B08, B04, B03)
nlyr(fcir)
```

```
## [1] 3
```

```r
fcir
```

```
## class       : SpatRaster
## dimensions  : 4046, 4022, 3  (nrow, ncol, nlyr)
## resolution  : 10, 10  (x, y)
## extent      : 330990, 371210, 3748690, 3789150  (xmin, xmax, ymin, ymax)
## coord. ref. : WGS 84 / UTM zone 11N (EPSG:32611)
## sources     : B08.tif
##               B04.tif
##               B03.tif
## names       : T11SLT_202~31_B08_10m, T11SLT_202~31_B04_10m, T11SLT_202~31_B03_10m
## min values  :                     1,                   563,                   140
## max values  :                 16736,                 17264,                 17936
```

Rename the bands for clarity.

```r
names(fcir)
```

```
## [1] "T11SLT_20250112T183731_B08_10m" "T11SLT_20250112T183731_B04_10m"
## [3] "T11SLT_20250112T183731_B03_10m"
```

```r
names(fcir) <- c("nir","red","green")
names(fcir)
```

```
## [1] "nir"   "red"   "green"
```

```r
fcir
```

```
## class       : SpatRaster
## dimensions  : 4046, 4022, 3  (nrow, ncol, nlyr)
## resolution  : 10, 10  (x, y)
## extent      : 330990, 371210, 3748690, 3789150  (xmin, xmax, ymin, ymax)
## coord. ref. : WGS 84 / UTM zone 11N (EPSG:32611)
## sources     : B08.tif
##               B04.tif
##               B03.tif
## names       :   nir,   red, green
## min values  :     1,   563,   140
## max values  : 16736, 17264, 17936
```

False Color Shortwave Infrared

Create a multi-band raster using SWIR, NIR, and Red bands. Since the resolutions of the SWIR band (B11) and other bands differ, resample the bands to match the SWIR resolution before combining.

```r
# Resample bands to match the SWIR resolution
B04_20 <- resample(B04, B11, method = "bilinear")
B08_20 <- resample(B08, B11, method = "bilinear")
```

Combine the resampled bands into a new multi-layer raster and rename the layers.

```r
fcswir <- c(B11, B08_20, B04_20)
names(fcswir) <- c("swir","nir","red")
fcswir
```

```
## class       : SpatRaster
## dimensions  : 2022, 2012, 3  (nrow, ncol, nlyr)
## resolution  : 20, 20  (x, y)
## extent      : 330980, 371220, 3748700, 3789140  (xmin, xmax, ymin, ymax)
```

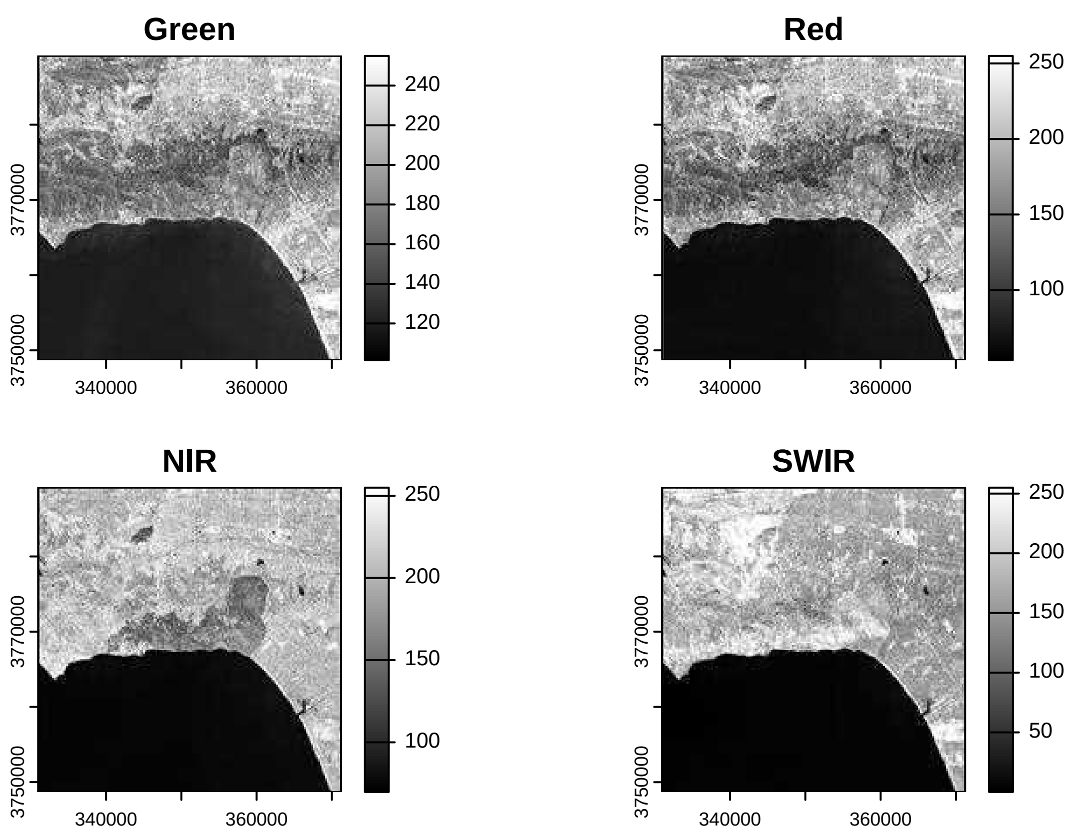
```
## coord. ref. : WGS 84 / UTM zone 11N (EPSG:32611)
## sources      : B11.tif
##                memory
##                memory
## varnames     : B11
##                B11
##                B11
## names        :  swir,       nir,       red
## min values   :   978,  847.2344,  980.5781
## max values   : 16164, 9424.7031, 9400.0938
```

BASE PLOTS

Per Band

Visualize individual bands using base R plotting. Adjust the color scheme and stretch values for better visualization.

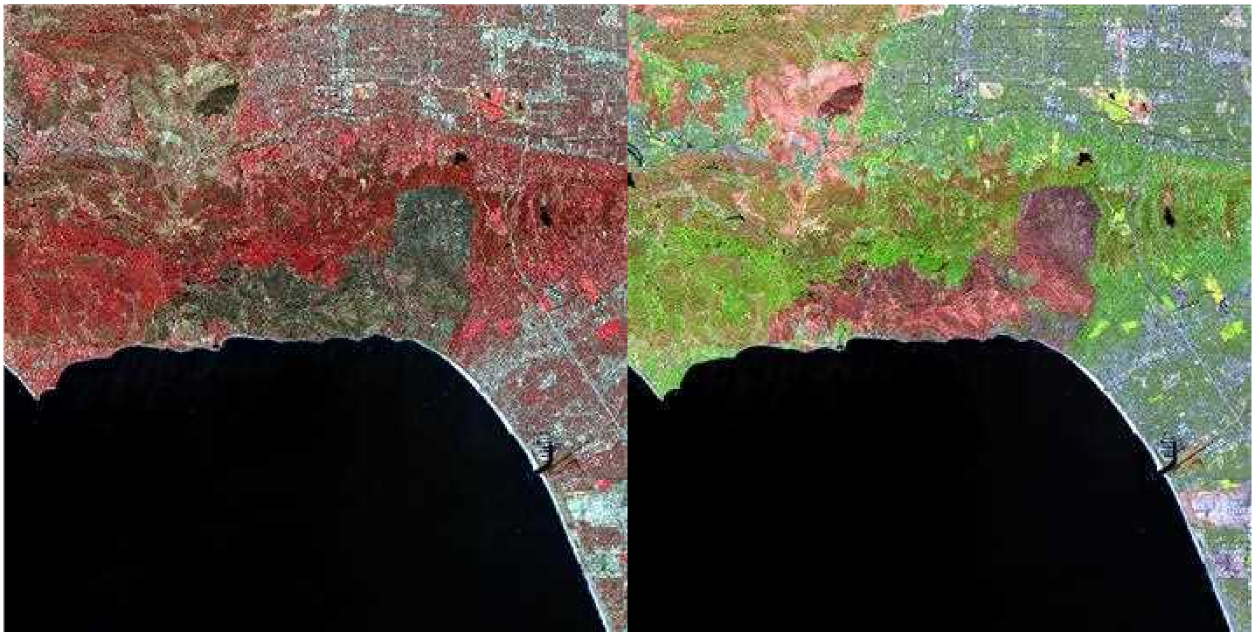
```
par(mfrow = c(2,2))
plot(stretch(B03, minq = 0, maxq = 0.9), main = "Green", col = gray(0:100 / 100))
plot(stretch(B04, minq = 0, maxq = 0.9), main = "Red", col = gray(0:100 / 100))
plot(stretch(B08, minq = 0, maxq = 0.9), main = "NIR", col = gray(0:100 / 100))
plot(stretch(B11, minq = 0, maxq = 0.9), main = "SWIR", col = gray(0:100 / 100))
```



RGB

Create RGB composite visualizations for both false-color infrared and shortwave infrared images.

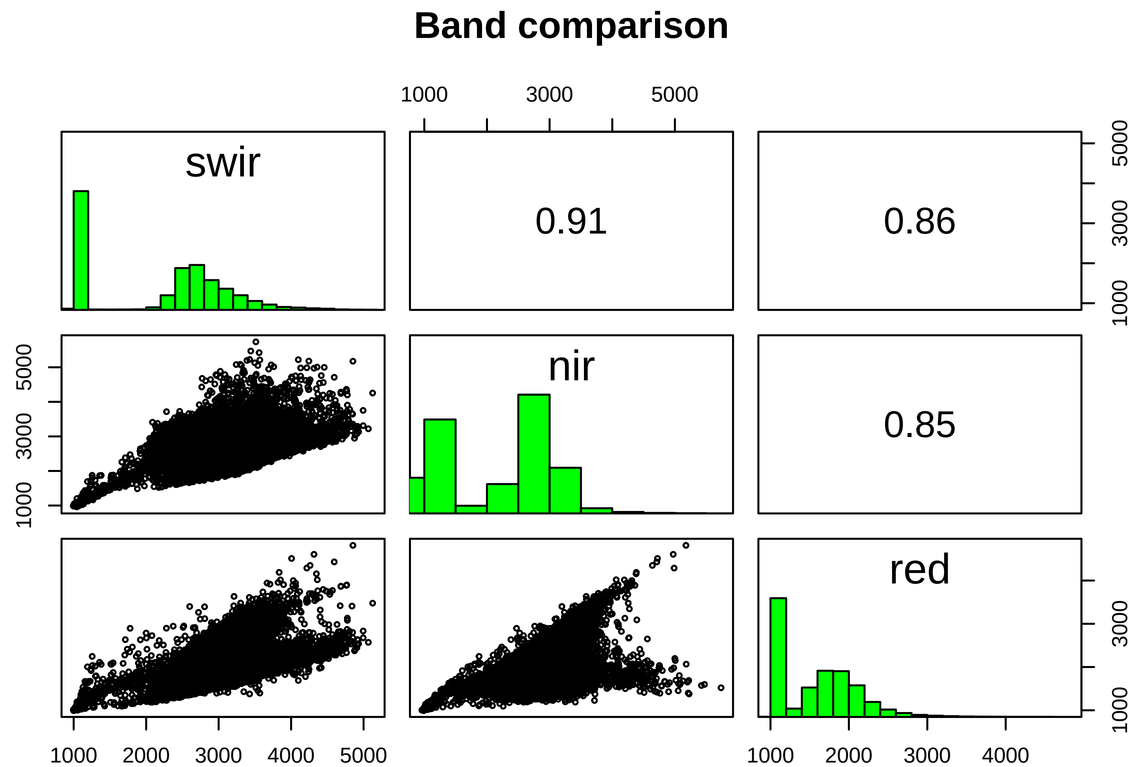
```r
par(mfrow = c(1,2))
plotRGB(fcir, stretch = "lin")
plotRGB(fcswir, stretch = "lin")
```



Pairs

Generate scatterplots to compare pixel values across different bands.

```r
pairs(aggregate(fcswir, 10), main = "Band comparison")
```

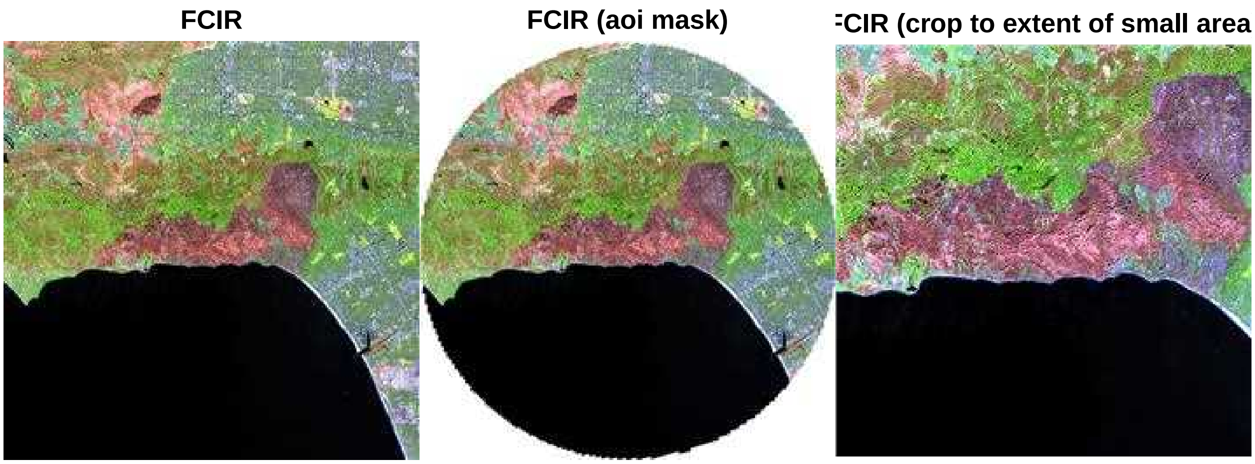


Conversions

Mask (Clip) and Crop

Clip and crop the raster to focus on the area of interest (AOI). The `mask` function retains pixels within the AOI, while the `crop` function limits the raster to the AOI's extent.

```
my_mask <- st_transform(aoi, crs=st_crs(fcswir))
fcswir_mask <- terra::mask(fcswir, my_mask)

my_crop <- ext(st_buffer(st_transform(aoi, crs=st_crs(fcswir)), -10000))
fcswir_crop <- terra::crop(fcswir, my_crop)

par(mfrow = c(1,3))
plotRGB(fcswir, stretch = "lin", main = "FCIR")
plotRGB(fcswir_mask, stretch = "lin", main = "FCIR (aoi mask)")
plotRGB(fcswir_crop, stretch = "lin", main = "FCIR (crop to extent of small area)")
```



Data Types

Explore and modify raster data types. These include logical, integer, and floating-point types. Save rasters in different formats for demonstration purposes.

```
# Unsigned 16-bit integer
datatype(B08)
```

```
## [1] "INT2U"
```

```
terra::writeRaster(B08, "Week_3_2025-01-21_Tue_data/datatype_test/B08_INT2U.tif", datatype = "INT2U", o

# Logical (1-bit)
terra::writeRaster(B08>1000, "Week_3_2025-01-21_Tue_data/datatype_test/B08_LOG1S.tif", datatype = "LOG1S
```

```
## Warning: [options] LOG1S is not a valid datatype
```

```
# 32-bit float
terra::writeRaster(B08/10000, "Week_3_2025-01-21_Tue_data/datatype_test/B08_FLT4S.tif", datatype = "FLT4

# 64-bit float
terra::writeRaster(B08/10000, "Week_3_2025-01-21_Tue_data/datatype_test/B08_FLT8S.tif", datatype = "FLT8

# Unsigned 8-bit integer
terra::writeRaster(clamp(round((B08/10000)*255), lower = 0, upper = 255), "Week_3_2025-01-21_Tue_data/da
```

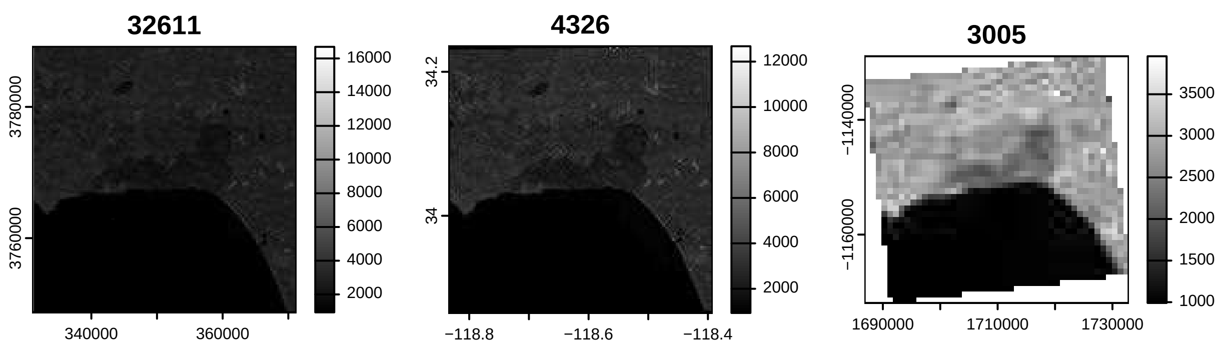
Transform / Warp / Resample

Transform raster data to different coordinate reference systems (CRS). Use bilinear resampling to adjust resolution as needed.

```
B08_4326 <- terra::project(B08, aoi)
```

```
## Warning: [project,SpatRaster] argument y (the crs) should be a character value
```

```
B08_3005 <- terra::project(x = B08,
                           y = "EPSG:3005",
                           method = "bilinear",
                           res = 1000)

par(mfrow = c(1,3))
plot(B08, main = "32611", col = gray(0:100 / 100))
plot(B08_4326, main = "4326", col = gray(0:100 / 100))
plot(B08_3005, main = "3005", col = gray(0:100 / 100))
```



Algebra

Perform raster algebra operations to derive new information or apply mathematical transformations.

```
max(fcswir)
```

```
## class       : SpatRaster
## dimensions  : 2022, 2012, 1  (nrow, ncol, nlyr)
## resolution  : 20, 20  (x, y)
## extent      : 330980, 371220, 3748700, 3789140  (xmin, xmax, ymin, ymax)
```

```
## coord. ref. : WGS 84 / UTM zone 11N (EPSG:32611)
## source(s)   : memory
## name        :   max
## min value   :   991
## max value   : 16164
```

```
mean(fcswir)
```

```
## class       : SpatRaster
## dimensions  : 2022, 2012, 1  (nrow, ncol, nlyr)
## resolution  : 20, 20  (x, y)
## extent      : 330980, 371220, 3748700, 3789140  (xmin, xmax, ymin, ymax)
## coord. ref. : WGS 84 / UTM zone 11N (EPSG:32611)
## source(s)   : memory
## name        :       mean
## min value   :   951.2031
## max value   : 11394.9062
```

```
min(fcswir)
```

```
## class       : SpatRaster
## dimensions  : 2022, 2012, 1  (nrow, ncol, nlyr)
## resolution  : 20, 20  (x, y)
## extent      : 330980, 371220, 3748700, 3789140  (xmin, xmax, ymin, ymax)
## coord. ref. : WGS 84 / UTM zone 11N (EPSG:32611)
## source(s)   : memory
## name        :       min
## min value   :  847.2344
## max value   : 8721.1875
```

```
sum(fcswir)
```

```
## class       : SpatRaster
## dimensions  : 2022, 2012, 1  (nrow, ncol, nlyr)
## resolution  : 20, 20  (x, y)
## extent      : 330980, 371220, 3748700, 3789140  (xmin, xmax, ymin, ymax)
## coord. ref. : WGS 84 / UTM zone 11N (EPSG:32611)
## source(s)   : memory
## name        :       sum
## min value   :  2853.609
## max value   : 34184.719
```

```
B08*B04
```

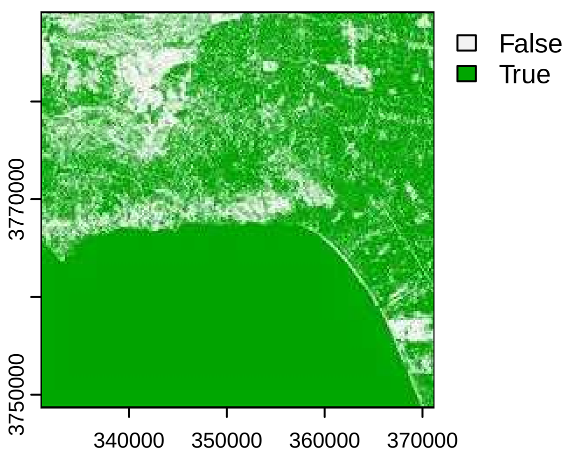
```
## class       : SpatRaster
## dimensions  : 4046, 4022, 1  (nrow, ncol, nlyr)
## resolution  : 10, 10  (x, y)
## extent      : 330990, 371210, 3748690, 3789150  (xmin, xmax, ymin, ymax)
## coord. ref. : WGS 84 / UTM zone 11N (EPSG:32611)
## source(s)   : memory
## varname     : B08
## name        : T11SLT_20250112T183731_B08_10m
## min value   :                            986
## max value   :                      288118896
```

```
B08+100
```

```
## class       : SpatRaster
```

```
## dimensions  : 4046, 4022, 1  (nrow, ncol, nlyr)
## resolution  : 10, 10  (x, y)
## extent      : 330990, 371210, 3748690, 3789150  (xmin, xmax, ymin, ymax)
## coord. ref. : WGS 84 / UTM zone 11N (EPSG:32611)
## source(s)   : memory
## varname     : B08
## name        : T11SLT_20250112T183731_B08_10m
## min value   :                            101
## max value   :                          16836
```

```
par(mfrow = c(1,2))
plot(B11<3000)
plotRGB(fcswir*(B11<3000), stretch = "lin")
```

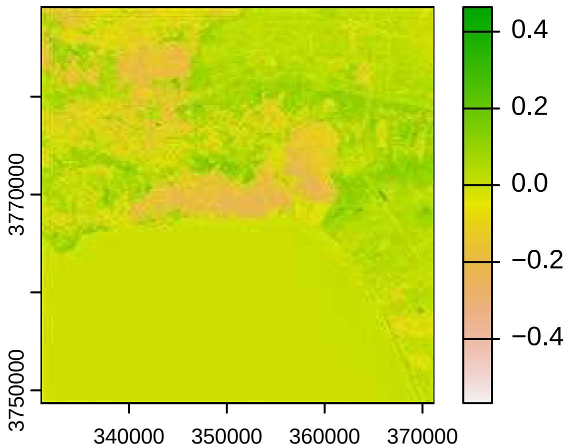


Functions

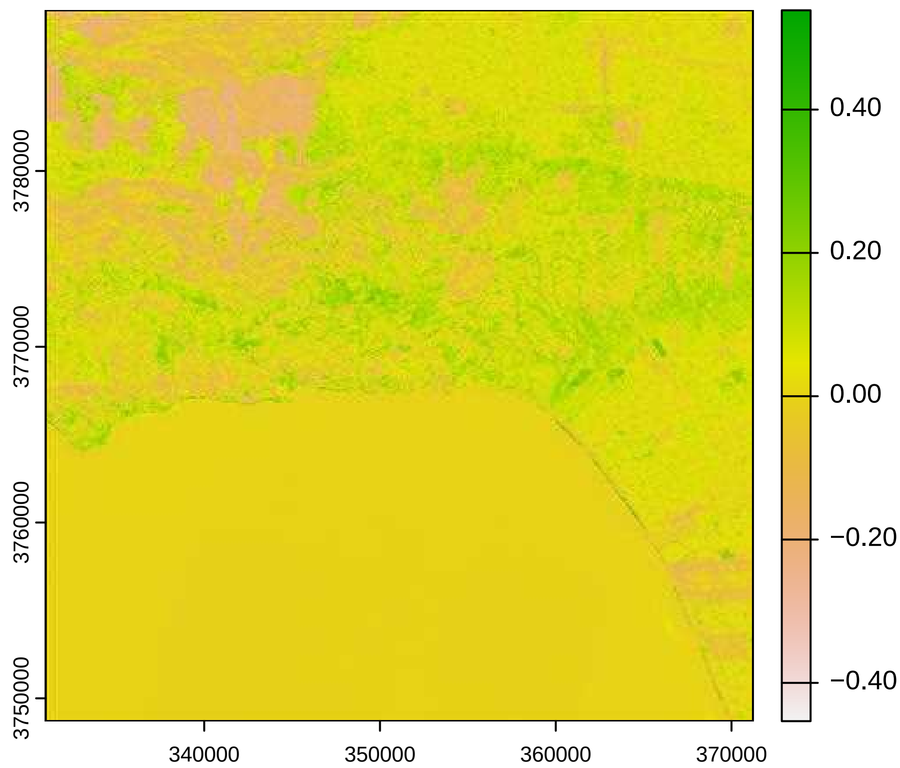
Define a custom function to calculate indices, such as the Normalized Difference.

```
nd_fun <- function(x, y, band_name) {
  nd <- (x - y) / (x + y)
  names(nd) <- band_name
  return(nd)
}

nbr <- nd_fun(B08_20, B11, "NBR")

par(mfrow = c(1,2))
```

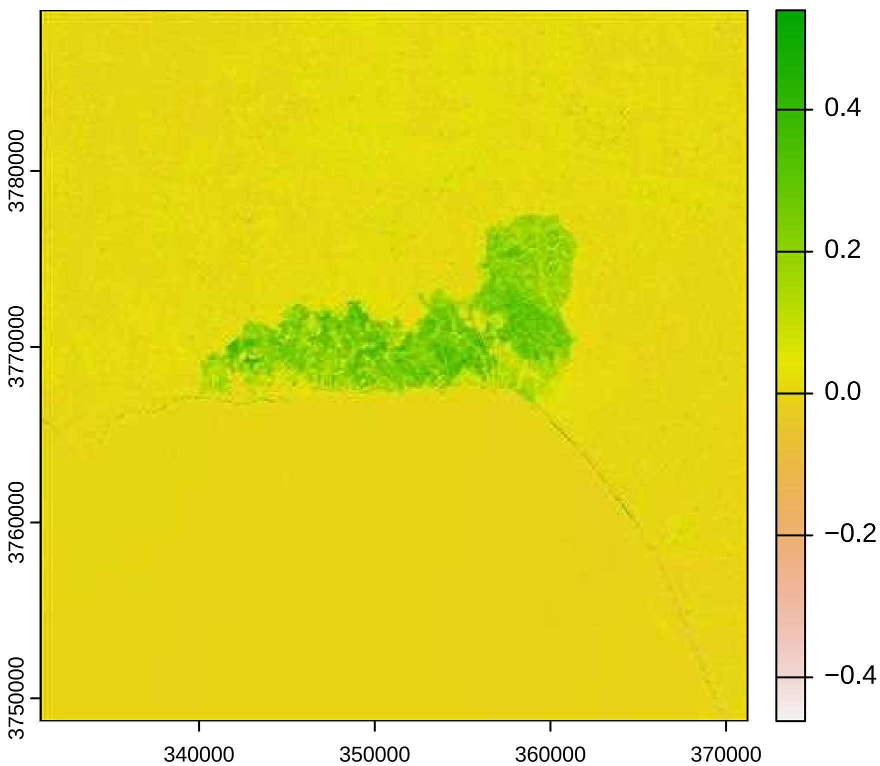
```r
plot(nbr)
```




```r
B08_pre <- terra::rast("Week_3_2025-01-21_Tue_data/pre-fire/B08.tif")
B11_pre <- terra::rast("Week_3_2025-01-21_Tue_data/pre-fire/B11.tif")
B08_pre_20 <- resample(B08_pre, B11_pre, method = "bilinear")

nbr_pre <- nd_fun(B08_pre_20, B11_pre, "NBR")
plot(nbr_pre)
```




```
par(mfrow = c(1,1))
barc <- nbr_pre-nbr
plot(barc)
```



Histogram

Convert raster data to a data frame for analysis and visualization. Plot histograms to explore the distribution of values.

```
head(as.data.frame(barc, xy = T))
```

```
##        x       y          NBR
## 1 330990 3789130  0.021508083
## 2 331010 3789130  0.002705406
## 3 331030 3789130  0.005527909
## 4 331050 3789130 -0.008392835
## 5 331070 3789130 -0.009653164
## 6 331090 3789130  0.014618796
```

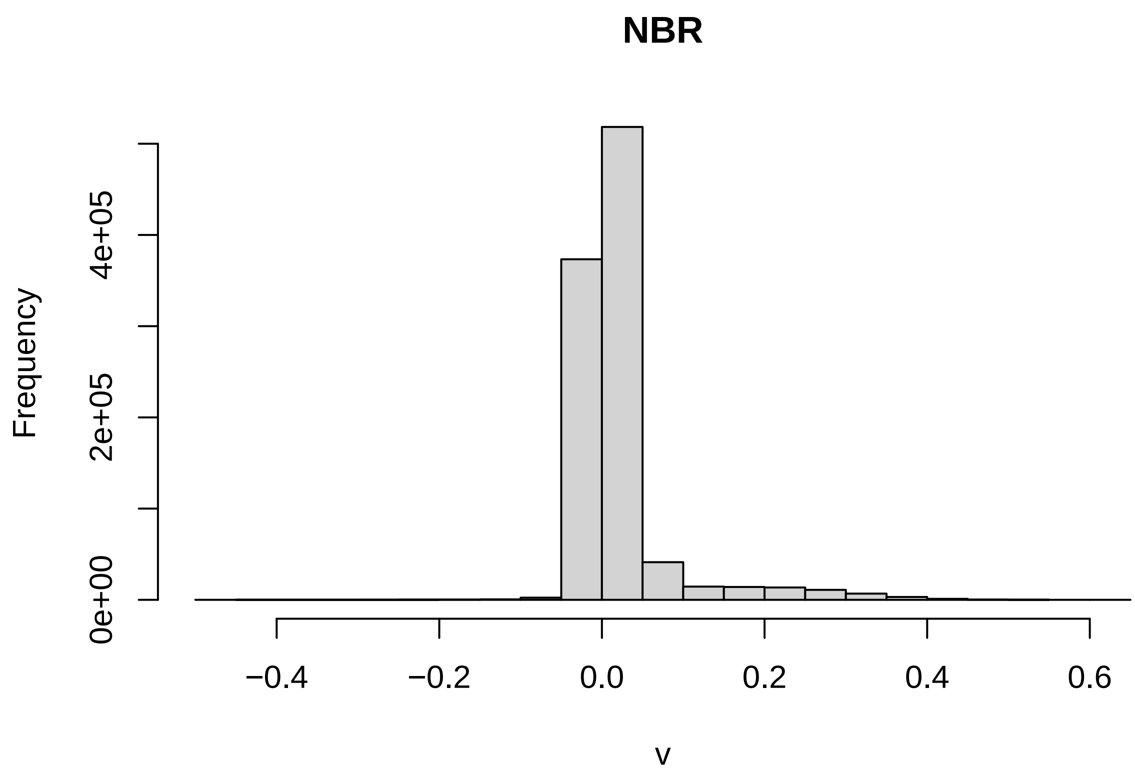
```
as_tibble(barc, xy = T)
```

```
## # A tibble: 4,068,264 x 3
##         x       y      NBR
##     <dbl>   <dbl>    <dbl>
##  1 330990 3789130  0.0215
##  2 331010 3789130  0.00271
##  3 331030 3789130  0.00553
##  4 331050 3789130 -0.00839
##  5 331070 3789130 -0.00965
##  6 331090 3789130  0.0146
##  7 331110 3789130  0.0132
```

```
## 8 331130 3789130  0.00773
## 9 331150 3789130  0.0180
## 10 331170 3789130  0.0101
## # i 4,068,254 more rows
```

```r
hist(barc)
```

```
## Warning: [hist] a sample of25% of the cells was used (of which 0% was NA)
```




```r
nbr_df <- as_tibble(barc, xy = T)
hist(barc$NBR)
```

```
## Warning: [hist] a sample of25% of the cells was used (of which 0% was NA)
```

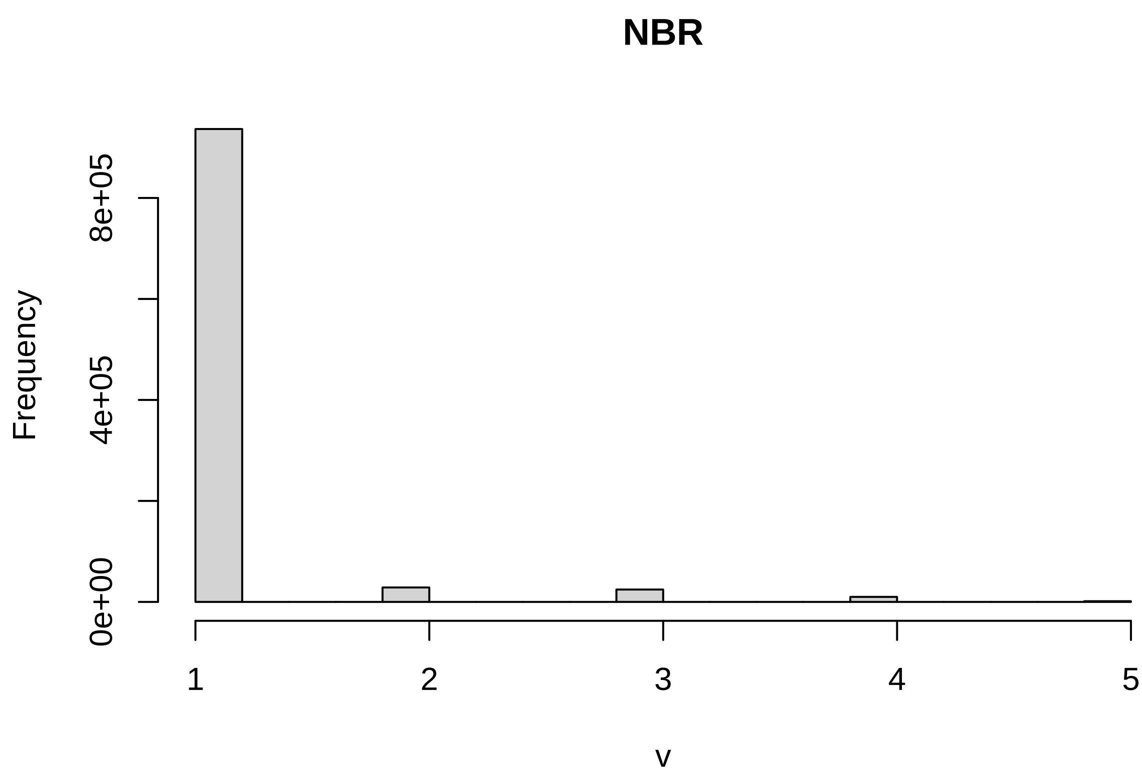
Reclassify

Reclassify raster values into discrete classes based on thresholds. Visualize the original and reclassified raster.

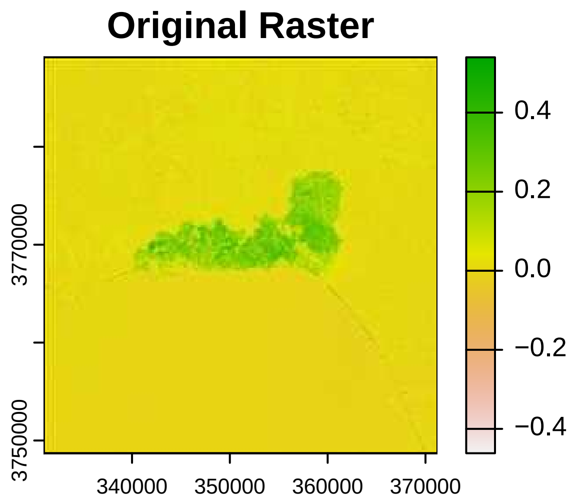
```r
reclass_matrix <- matrix(c(
  -Inf, .1, 1,    # Values 0-20 -> 1
  .1, .2, 2,   # Values 20-50 -> 2
  .2, .3, 3,   # Values 20-50 -> 2
  .3, .4, 4,   # Values 20-50 -> 2
  .4, 1, 5  # Values 50-100 -> 3
  ), ncol=3, byrow=TRUE)

# Reclassify the raster
barc_classified <- classify(barc, rcl=reclass_matrix)
```

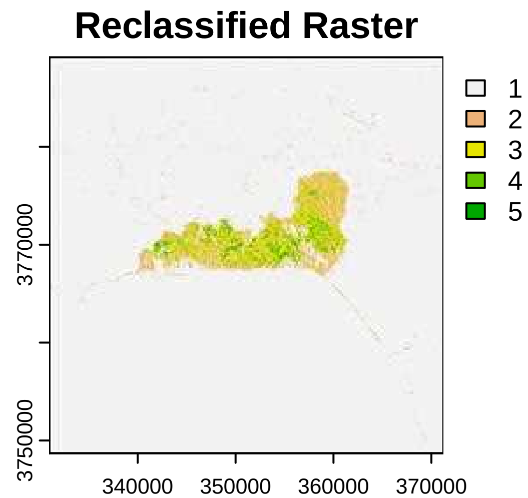
```r
par(mfrow = c(1,1))
hist(barc_classified)
```

```
## Warning: [hist] a sample of25% of the cells was used (of which 0% was NA)
```




```r
par(mfrow = c(1,2))
plot(barc, main="Original Raster")
plot(barc_classified, main="Reclassified Raster")
```

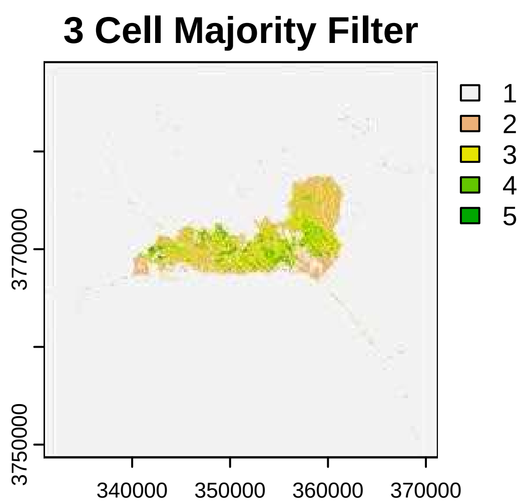


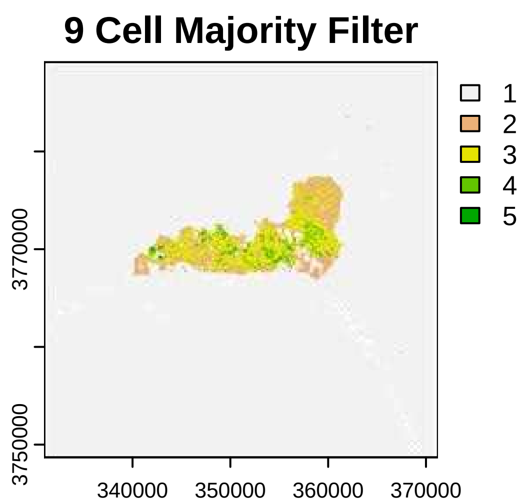


Focal

Apply focal functions, such as majority filtering, to smooth raster data. Compare outputs with different window sizes.

```
barc_classified_mod_3 <- focal(barc_classified, w=3, fun="modal")
barc_classified_mod_9 <- focal(barc_classified, w=9, fun="modal")

par(mfrow = c(1,2))
plot(barc_classified_mod_3, main="3 Cell Majority Filter")
plot(barc_classified_mod_9, main="9 Cell Majority Filter")
```





To Polygon

Convert raster data to vector polygons for further analysis. Add attributes and filter the dataset as needed.

```
barc_polygon <- terra::as.polygons(barc_classified_mod_9)
barc_polygon_sf <- st_as_sf(barc_polygon)
barc_polygon_sf <- barc_polygon_sf %>%
  mutate(severity = case_when(focal_modal == 1 ~ "0: No",
                              focal_modal == 2 ~ "1: Low",
                              focal_modal == 3 ~ "2: Moderate",
                              focal_modal == 4 ~ "3: High",
                              focal_modal == 5 ~ "4: Severe",
                              TRUE ~ NA)) %>%
  filter(!is.na(severity))
```

```
viewRGB(satellite::brick(fcswir), r = 1,g = 2,b = 3, quantiles = NULL) +
  mapview(filter(barc_polygon_sf, focal_modal > 1), zcol = "severity")
```

```
## Warning in rasterCheckSize(x, maxpixels = maxpixels): maximum number of pixels for Raster* viewing i
## the supplied Raster* has 4068264
##  ... decreasing Raster* resolution to 5e+05 pixels
##  to view full resolution set 'maxpixels =  4068264 '
```

```
## Warning in CPL_crs_from_input(x): GDAL Message 1: +init=epsg:XXXX syntax is
## deprecated. It might return a CRS with a non-EPSG compliant axis order.
```

DEM

Visualize and process Digital Elevation Models (DEMs). Combine tiles, calculate terrain attributes, and render hillshades.

```
dem_grid_1 <- rast("Week_3_2025-01-21_Tue_data/dem/grid1.tif")
dem_grid_2 <- rast("Week_3_2025-01-21_Tue_data/dem/grid2.tif")
mapview(dem_grid_1) +
  mapview(dem_grid_2, col.regions = hcl.colors(palette = "Spectral", n = 4))
```

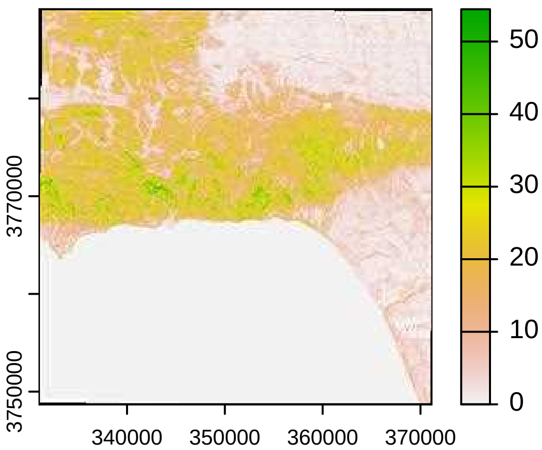
```
## Number of pixels is above 5e+05.Only about 5e+05 pixels will be shown.
## You can increase the value of `maxpixels` to 1310632 to avoid this.
```

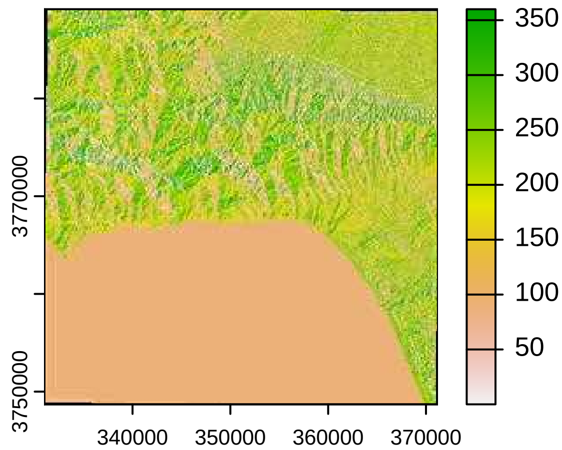
```
## Number of pixels is above 5e+05.Only about 5e+05 pixels will be shown.
## You can increase the value of `maxpixels` to 744464 to avoid this.
```

```
dem <- mosaic(dem_grid_1, dem_grid_2)
dem
```

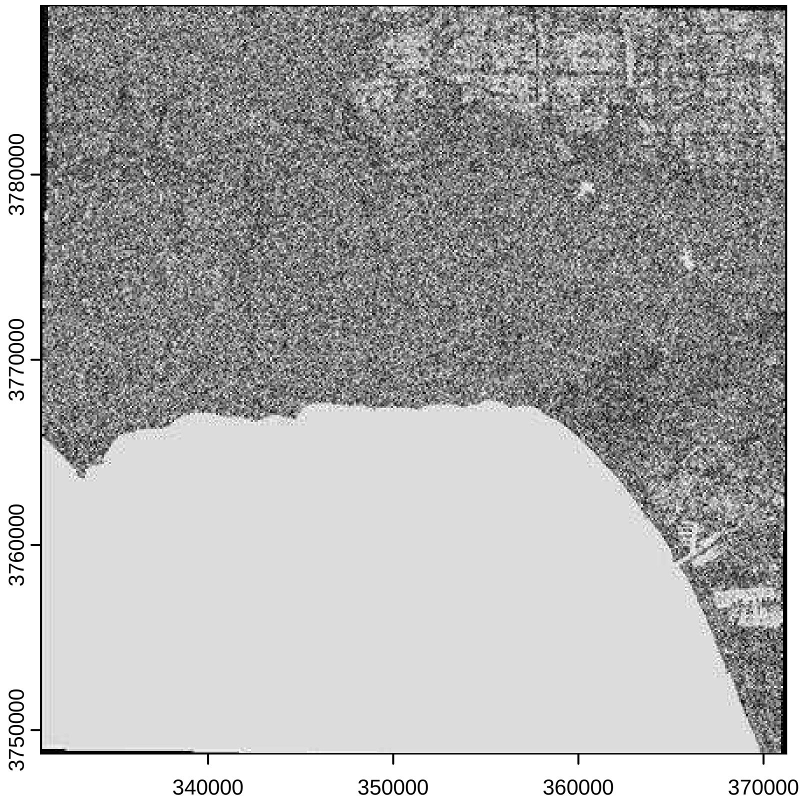
```
## class       : SpatRaster
## dimensions  : 1314, 1564, 1  (nrow, ncol, nlyr)
## resolution  : 0.0002777778, 0.0002777778  (x, y)
## extent      : -118.8304, -118.396, 33.86792, 34.23292  (xmin, xmax, ymin, ymax)
## coord. ref. : lon/lat WGS 84 (EPSG:4326)
## source(s)   : memory
## varname     : grid1
## name        : Copernicus_DSM_COG_10_N34_00_W119_00_DEM
## min value   :                                 -1.413523
## max value   :                                860.688782
```

```
dem_prj <- project(dem, B11)
names(dem_prj) <- "dem"
slp_deg <- terrain(dem_prj, "slope", unit="degrees")
asp_deg <- terrain(dem_prj, "aspect", unit="degrees")
par(mfrow = c(1,2))
plot(slp_deg)
plot(asp_deg)
```

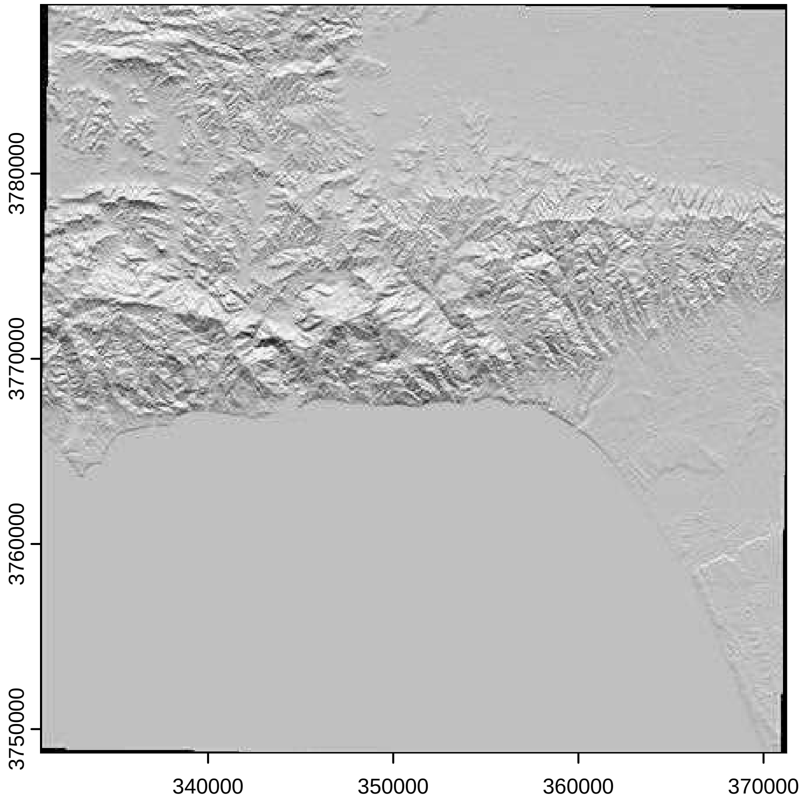





```
hs <- terra::shade(slp_deg, asp_deg)
par(mfrow = c(1,1))
plot(hs, col=grey(0:100/100), legend=FALSE, mar=c(2,2,1,4))
```




```
slp_rad <- terrain(dem_prj, "slope", unit="radians")
asp_rad <- terrain(dem_prj, "aspect", unit="radians")
hs_rad <- terra::shade(slp_rad, asp_rad)
par(mfrow = c(1,1))
plot(hs_rad, col=grey(0:100/100), legend=FALSE, mar=c(2,2,1,4))
```



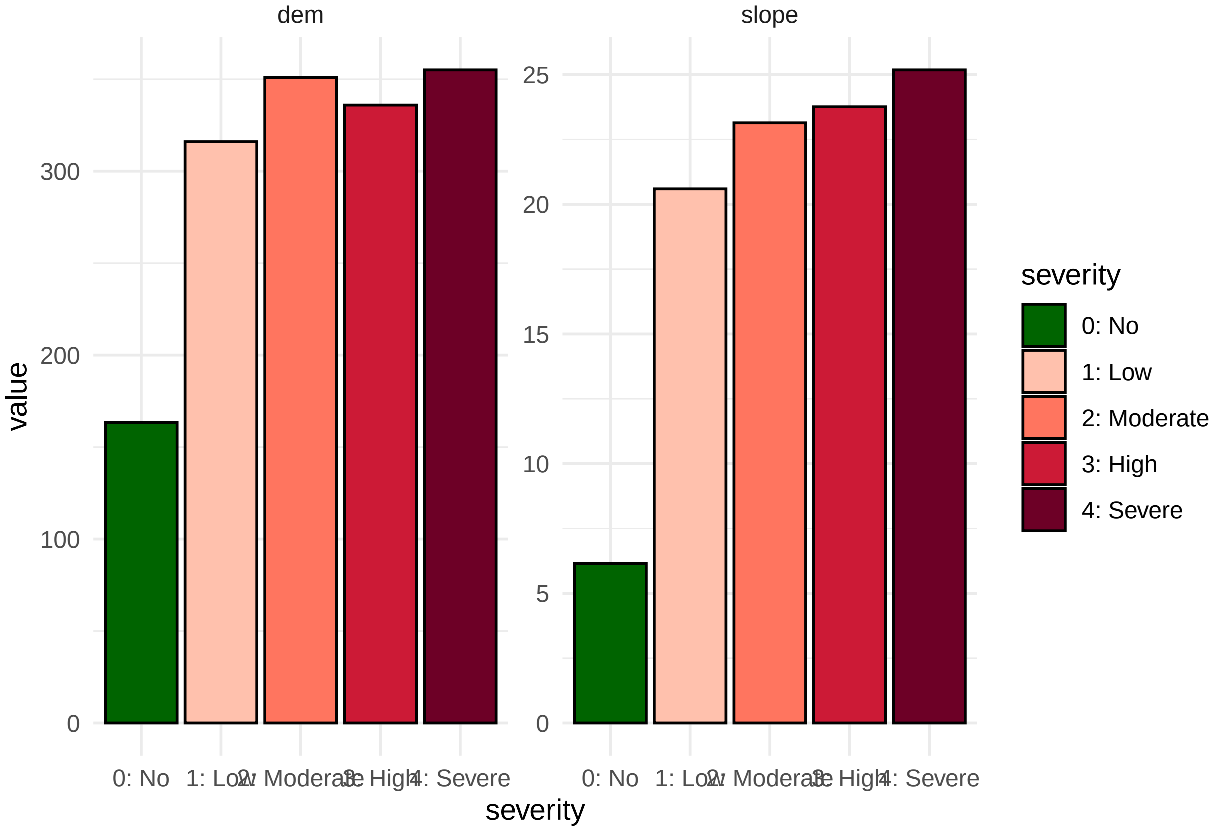
ZONAL

Calculate zonal statistics to summarize raster values within specific zones.

```
zonal_mean <- zonal(c(dem_prj,slp_deg),
                    barc_classified_mod_9,
                    fun="mean", na.rm = T)


barc_polygon_df <- barc_polygon_sf %>%
  st_drop_geometry() %>%
  full_join(zonal_mean)
```

```
## Joining with `by = join_by(focal_modal)`
```

```
barc_polygon_df %>%
  pivot_longer(cols = -c(focal_modal,severity)) %>%
  ggplot() +
  geom_col(aes(severity, value, fill = severity), position = "dodge", color = "black") +
  facet_wrap(.~name, scales = "free_y", ncol = 3) +
  scale_fill_manual(values = c("darkgreen",rev(hcl.colors(5, "reds"))[2:5])) +
  theme_minimal()
```


dem
slope
value
300
200
100
0
25
20
15
10
5
0
0: No
1: Low
2: Moderate
3: High
4: Severe
severity
severity
0: No
1: Low
2: Moderate
3: High
4: Severe