**Errors in modules:**

<https://datacarpentry.org/r-raster-vector-geospatial/01-raster-structure/index.html>

Challenge: Explore Raster Metadata

“Notice: this file is a hillshade. We will learn about hillshades in the [Working with Multi-band Rasters in R](https://datacarpentry.org/r-raster-vector-geospatial/05-raster-multi-band-in-r/) episode” caused one of my groups to think that this was the next module in the series. I think the comment should be removed, or at least the link to the multi-band episode so they are not tempted to click on it.

<https://datacarpentry.org/r-raster-vector-geospatial/03-raster-reproject-in-r/index.html>

Challenge: Reproject, then Plot a Digital Terrain Model

 *# reproject raster*

DTM\_hill\_UTMZ18N\_SJER <- projectRaster(DSM\_hill\_SJER\_WGS,

 crs = crs(DSM\_SJER),

 res = 1)

*# convert to data.frames*

DSM\_SJER\_df <- as.data.frame(DSM\_SJER, xy = **TRUE**)

DSM\_hill\_SJER\_df <- as.data.frame(DTM\_hill\_UTMZ18N\_SJER, xy = **TRUE**)

**CORRECTION:**

DTM\_hill\_UTMZ18N\_SJER <- projectRaster(DTM\_hill\_SJER\_WGS,

 crs = crs(DSM\_SJER),

 res = 1)

*# convert to data.frames*

DTM\_SJER\_df <- as.data.frame(DTM\_SJER, xy = **TRUE**)

DTM\_hill\_SJER\_df <- as.data.frame(DTM\_hill\_UTMZ18N\_SJER, xy = **TRUE**)

<https://datacarpentry.org/r-raster-vector-geospatial/07-vector-shapefile-attributes-in-r/index.html>

Section: Explore values within one attribute

To see only unique values within the TYPE field, we can use the levels() function for extracting the possible values of a categorical variable. The special term for categorical variables within R is factor. We worked with factors a little bit in an earlier lesson.

> levels(lines\_HARV$TYPE)

> NULL

#This answer does not tell us the unique values within the TYPE field!

**Correction:**

> levels(factor(lines\_HARV$TYPE))

#now you see 4 unique factors ordered alphabetically

<https://datacarpentry.org/r-raster-vector-geospatial/07-vector-shapefile-attributes-in-r/index.html>

Challenge: Plot Polygon by Attribute

Apply a fill color to each state using its region value. Add a legend.

If you want fill colors, answer should be:

> ggplot() +

geom\_sf(data=state\_boundary\_US, aes(fill = region), colour = 'white', size = 0.7, linetype = 'solid') +

 ggtitle("Continental U.S. State Boundaries") +

 coord\_sf()

<https://datacarpentry.org/r-raster-vector-geospatial/08-vector-plot-shapefiles-custom-legend/index.html>

Challenge: Plot Polygon by Attribute

* Should be called: Plot Points by Attribute
* Errors in code that creates both plots:

Error in check.length(gparname): 'gpar' element 'fontsize' must not be length 0

<https://datacarpentry.org/r-raster-vector-geospatial/11-vector-raster-integration/index.html>

Section: Extract Raster Pixels Values Using Vector Polygons

In text right before code to create histogram: “We will use the column layer from our data frame as our x values” – there is no column called layer in tree height data frame; there is a column called ID.

<https://datacarpentry.org/r-raster-vector-geospatial/12-time-series-raster/index.html>

Section: Explore Unusual Data Patterns

We only want to look at the data from 2011:

> yr\_11\_daily\_avg <- subset(har\_met\_daily,

 date >= as.Date('2011-01-01') &

 date <= as.Date('2011-12-31'))

as.Date is not necessary because the section directly above already used as.Date() to convert

<https://datacarpentry.org/r-raster-vector-geospatial/14-extract-ndvi-from-rasters-in-r/index.html>

Section: Calculate Average NDVI

We need to create a data frame from the named numeric vector:

> avg\_NDVI\_HARV <- as.data.frame(avg\_NDVI\_HARV)

The code works fine, but assigns the same name to the output data frame. This is generally confusing, but also means that later code can act on either one. I named the data frame **avg\_NDVI\_HARV\_df** and replaced the name in all code going forward.

Also, I think this last module took longer than expected: about 75-80 minutes.