Introduction to Mapping Geographic Data

HES 505 Fall 2023: Session 10

Matt Williamson

Today's Plan

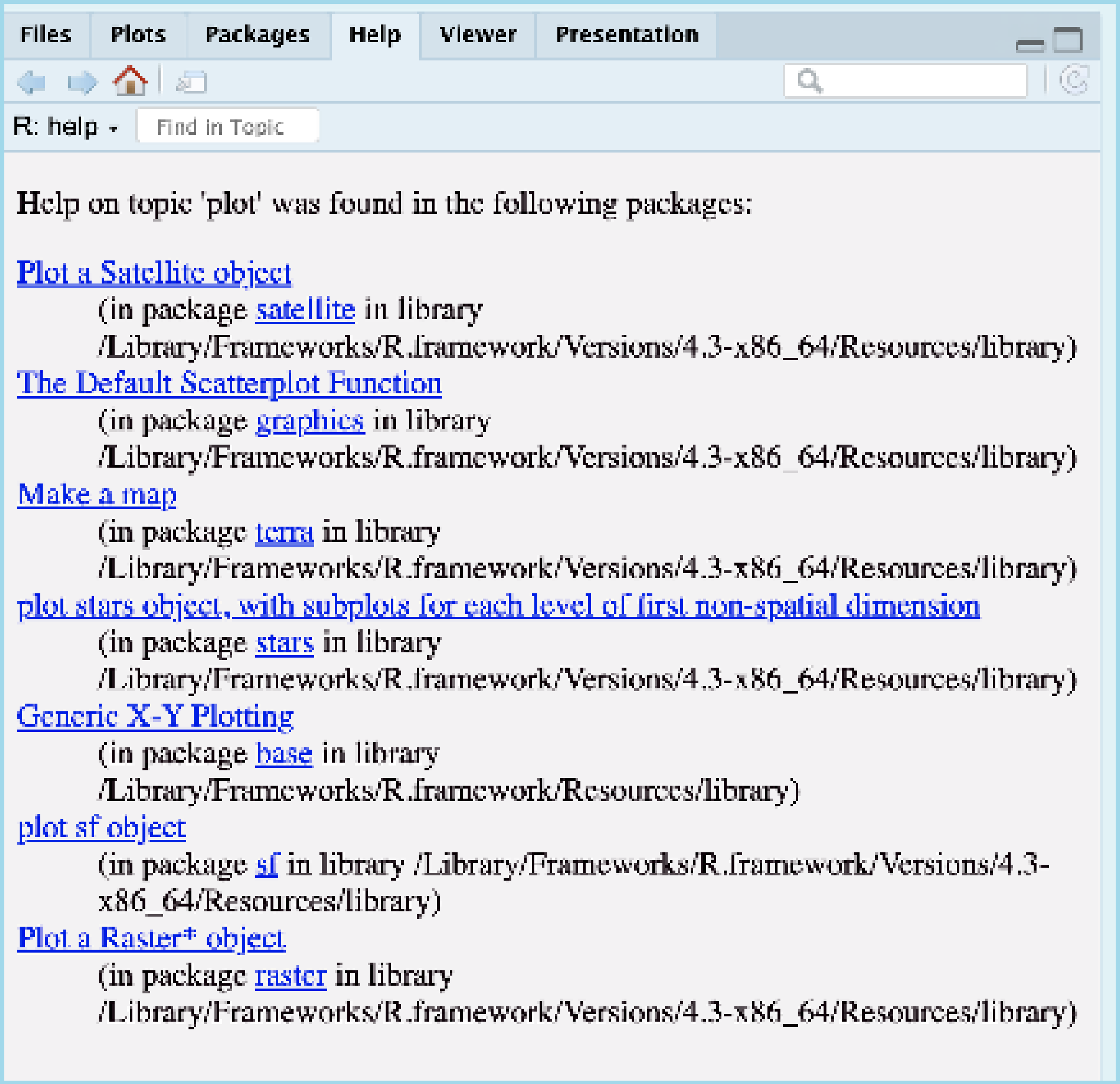
Objectives

- By the end of today, you should be able to:
- Describe the basic components of data visualization as a foundation for mapping syntax
- Understand layering in both base `plot` and `tmap`
- Make basic plots of multiple spatial data objects

Using `plot`

Which packages have `plot` methods?

- Often the fastest way to view data
- Use `?plot` to see which packages export a method for the `plot` function
- Or you can use `?plot.***` to see which classes of objects have plot functions defined



Help on topic 'plot' was found in the following packages:

Plot a Satellite object
(in package satellite in library /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library)
The Default Scatterplot Function
(in package graphics in library /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library)
Make a map
(in package terra in library /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library)
plot stars object, with subplots for each level of first non-spatial dimension
(in package stars in library /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library)
Generic X-Y Plotting
(in package base in library /Library/Frameworks/R.framework/Resources/library)
plot sf object
(in package sf in library /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library)
Plot a Raster* object
(in package raster in library /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/library)

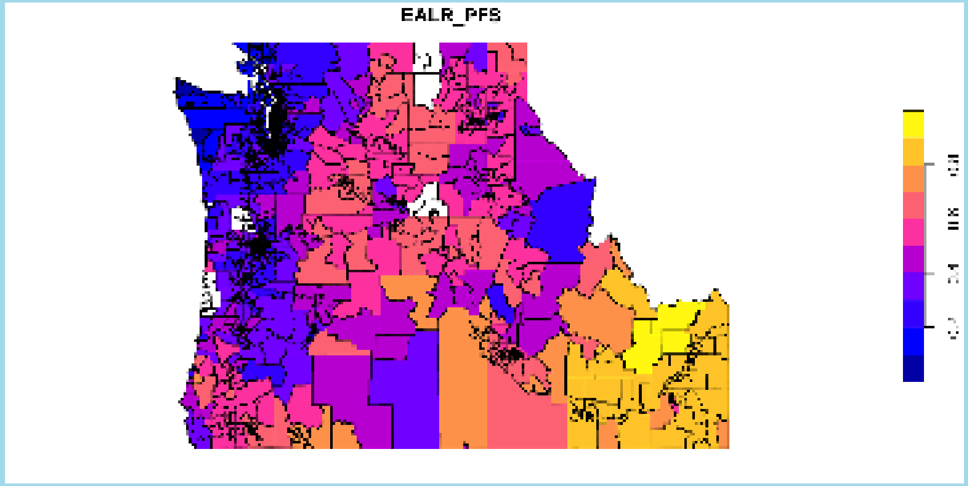
`plot` for `sf` objects

- Can plot outlines using `plot(st_geometry(your.shapfile))` or `plot(your.shapefile$geometry)`
- Plotting attributes requires “extracting” the attributes (using `plot(your.shapefile["ATTRIBUTE"])`)
- Controlling aesthetics can be challenging
- layering requires `add=TRUE`

`plot` for `sf` objects

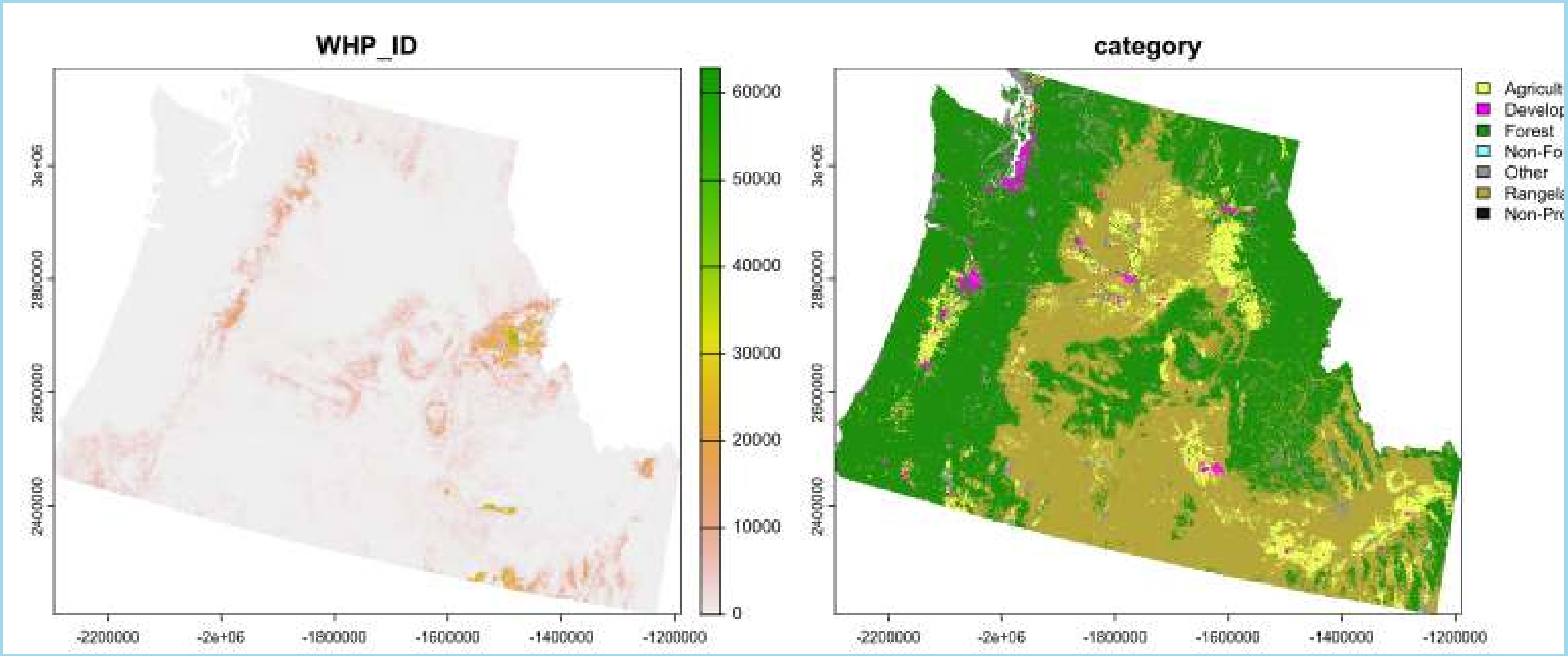
```r
plot(st_geometry(cejst))
```

```r
plot(cejst["EALR_PFS"])
```



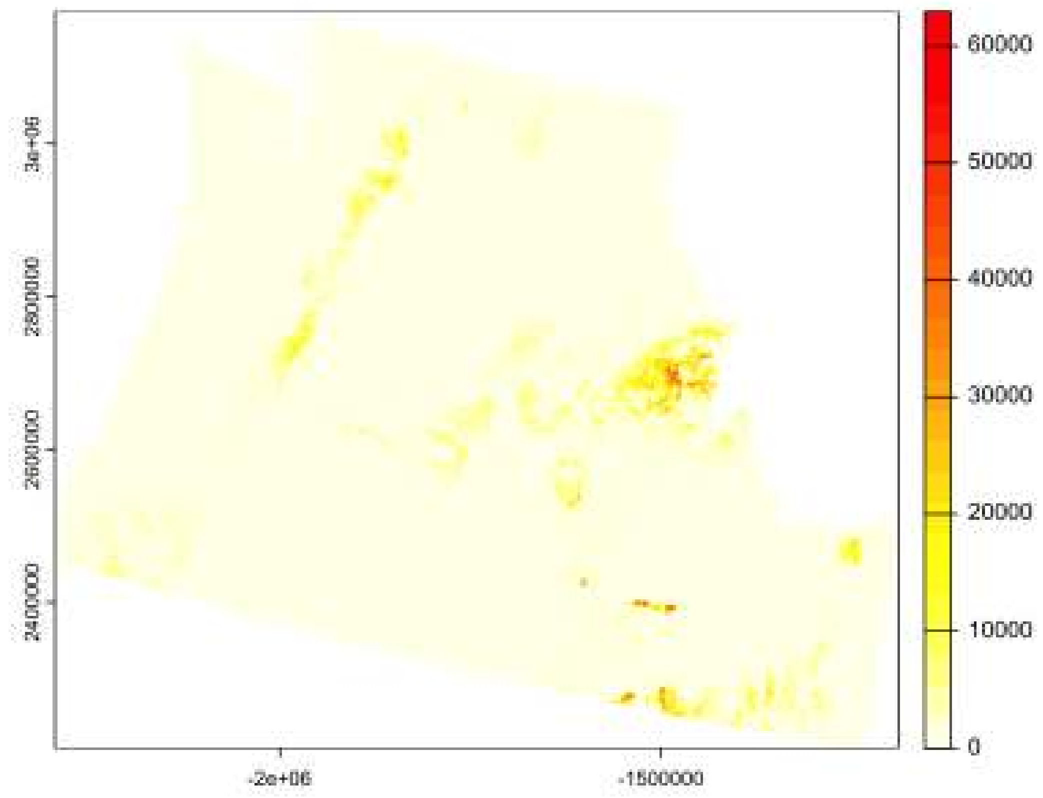
plot for SpatRasters

```r
plot(rast.data)
```



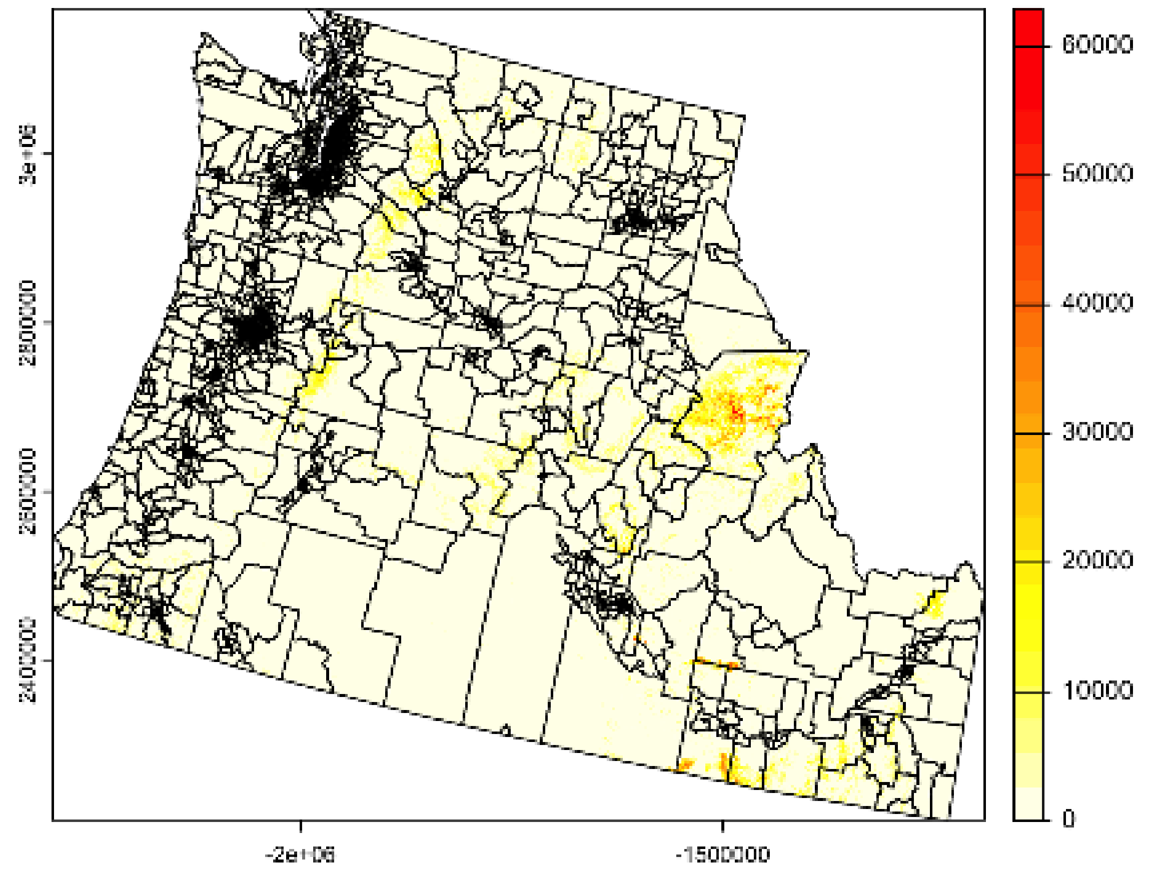
plot for SpatRasters

```r
plot(rast.data["WHP_ID"], col=heat.colors(24, rev=TRUE))
```



Combining the two with add=TRUE

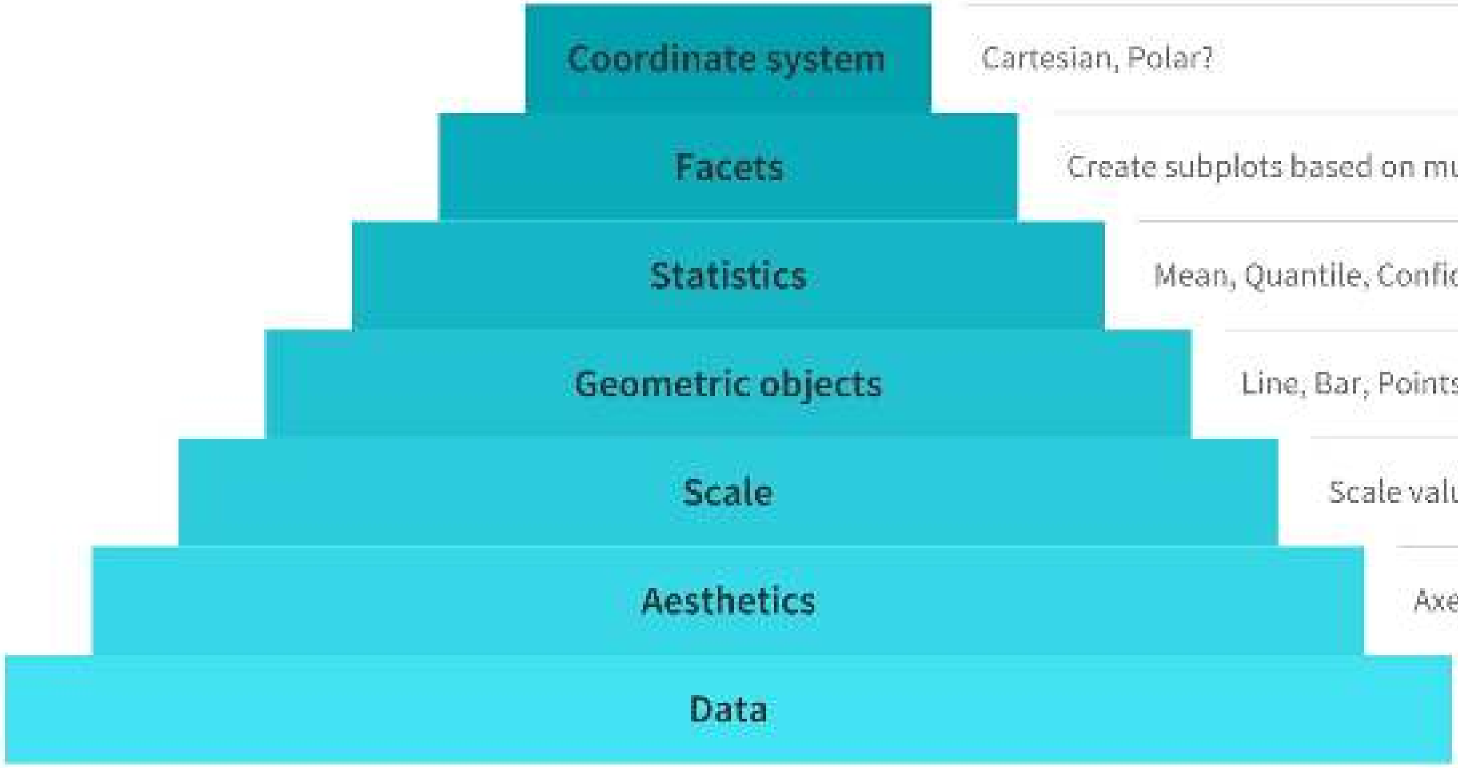
```r
plot(rast.data["WHP_ID"], col=heat.colors(24, rev=TRUE))
plot(st_geometry(st_transform(cejst, crs=crs(rast.data))), add=TRUE)
```



Thinking about map construction

Grammar of Graphics (Wilkinson 2005)

Major Components of the Grammar of Graphics



Coordinate system	Cartesian, Polar?
Facets	Create subplots based on multiple dimensions
Statistics	Mean, Quantile, Confidence Intervals?
Geometric objects	Line, Bar, Points?
Scale	Scale values, represent multiple values?
Aesthetics	Axes, plot positions, encodings?
Data	Our datasets

Aesthetics: Mapping Data to Visual Elements

- Define the systematic conversion of data into elements of the visualization
- Are either categorical or continuous (exclusively)
- Examples include `x`, `y`, `fill`, `color`, and `alpha`

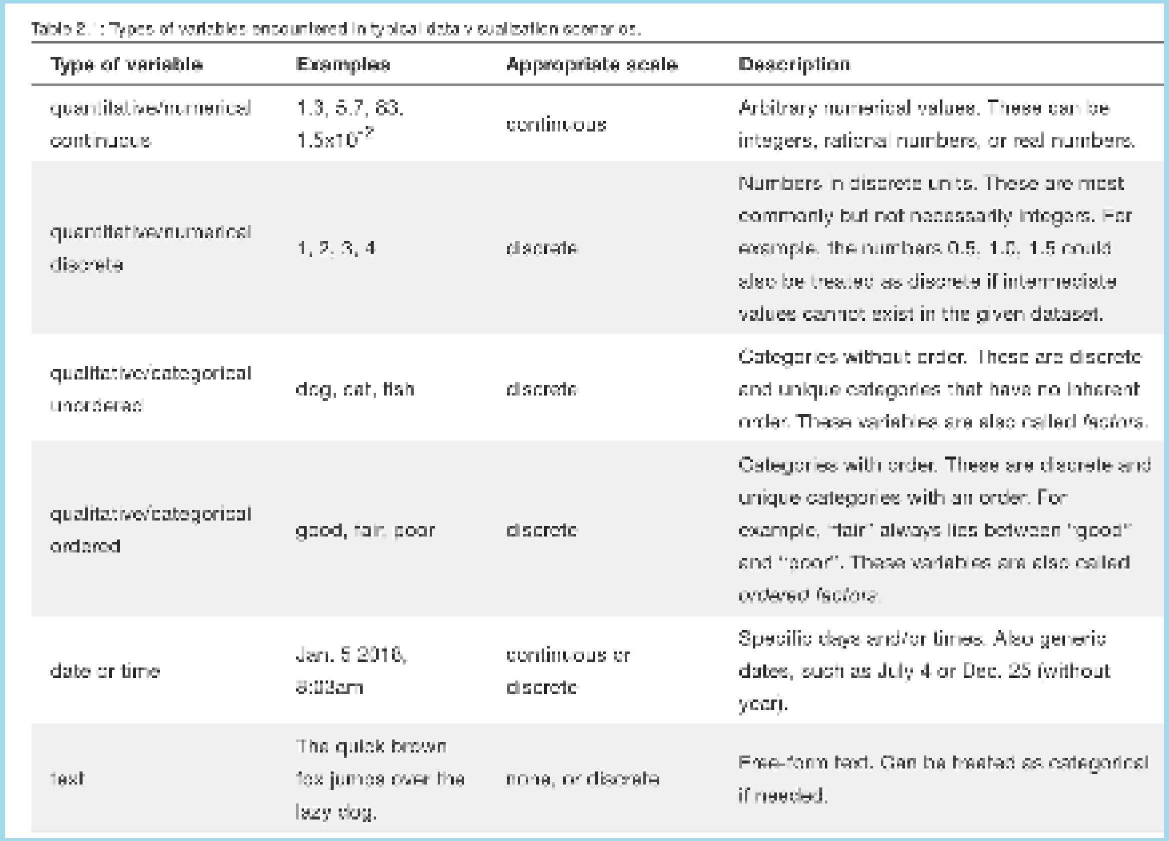


Table 2.1: Types of variables encountered in typical data visualization scenarios.

Type of variable	Examples	Appropriate scale	Description
quantitative/numerical continuous	1.3, 5.7, 83, 1.5×10^{-2}	continuous	Arbitrary numerical values. These can be integers, rational numbers, or real numbers.
quantitative/numerical discrete	1, 2, 3, 4	discrete	Numbers in discrete units. These are most commonly but not necessarily integers. For example, the numbers 0.5, 1.0, 1.5 could also be treated as discrete if intermediate values cannot exist in the given dataset.
qualitative/categorical unordered	dog, cat, fish	discrete	Categories without order. These are discrete and unique categories that have no inherent order. These variables are also called *factors*.
qualitative/categorical ordered	good, fair, poor	discrete	Categories with order. These are discrete and unique categories with an order. For example, "fair" always lies between "good" and "poor". These variables are also called *ordered factors*.
date or time	Jan. 5 2018, 8:03am	continuous or discrete	Specific days and/or times. Also generic dates, such as July 4 or Dec. 25 (without year).
text	The quick brown fox jumps over the lazy dog.	none, or discrete	Free-form text. Can be treated as categorical if needed.

From Wilke 2019

Scales

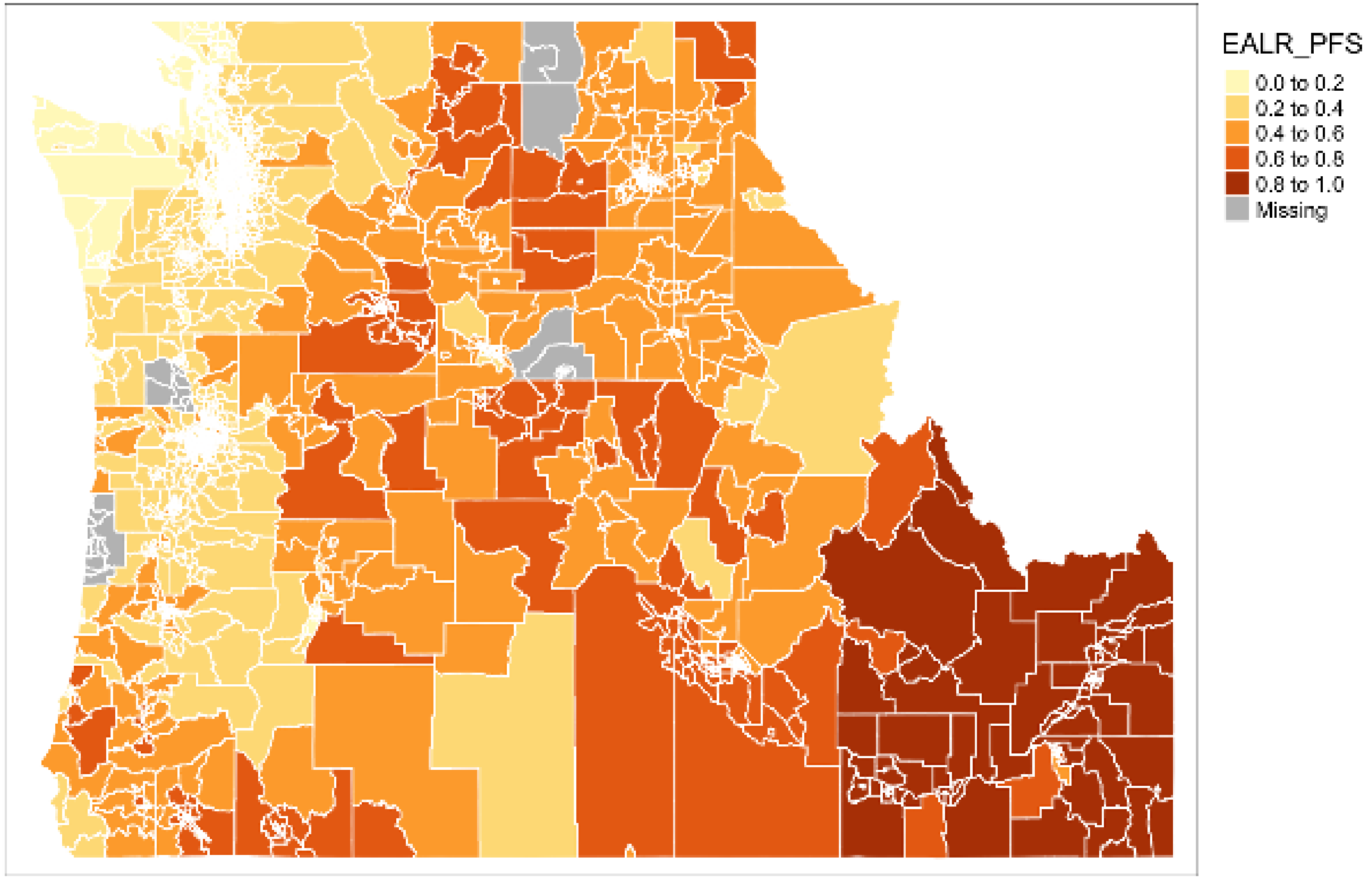
- Scales map data values to their aesthetics
- Must be a one-to-one relationship; each specific data value should map to only one aesthetic

Adding aesthetics with `tmap`

Using tmap

```r
library(sf)
library(terra)
library(tmap)
pt <- tm_shape(cejst) +
  tm_polygons(col = "EALR_PFS",
              border.col = "white") +
  tm_legend(outside = TRUE)
```

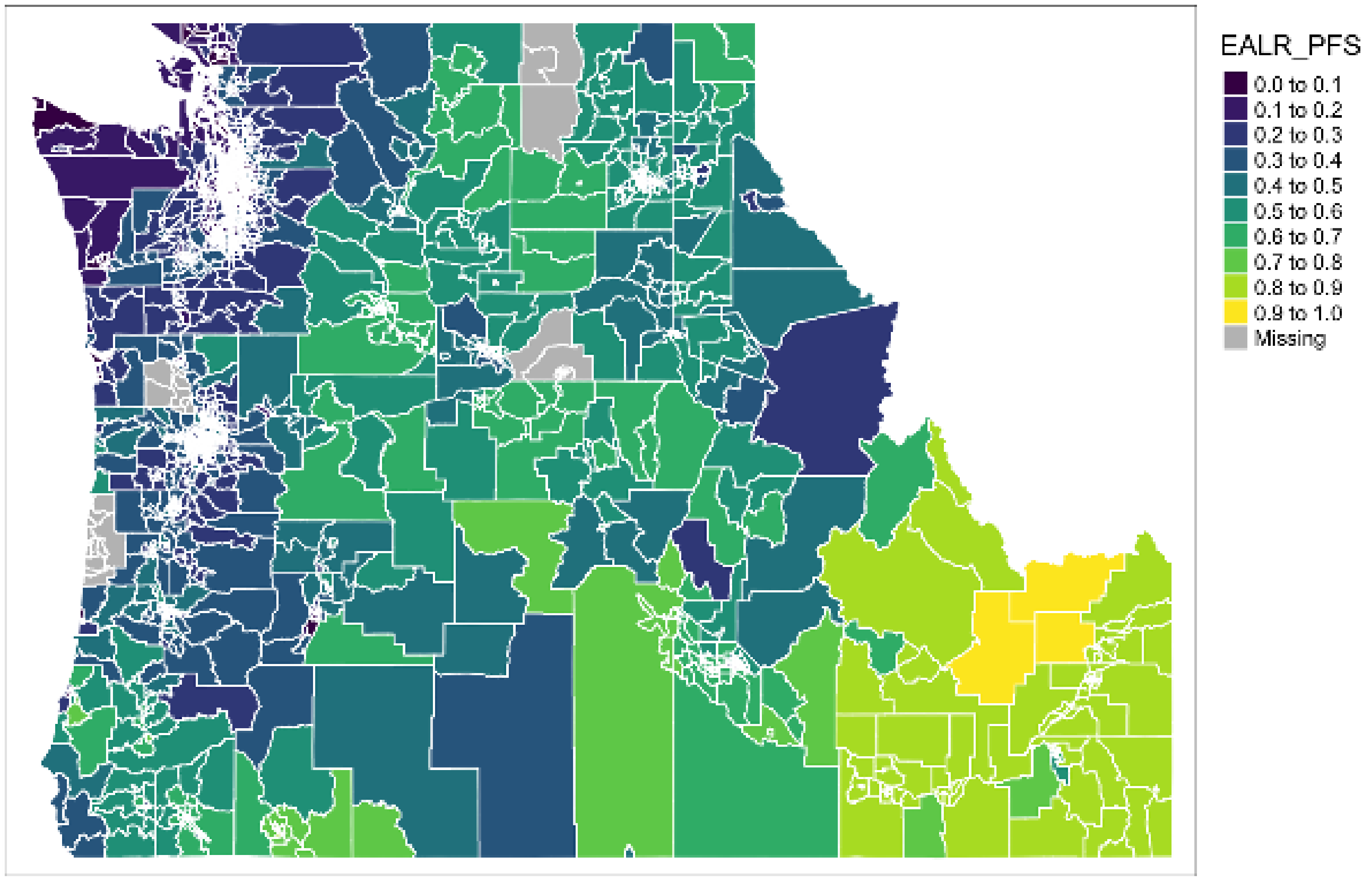
Using tmap



Changing aesthetics

```r
pt <- tm_shape(cejst) +
  tm_polygons(col = "EALR_PFS", n=10,palette=viridis(10),
              border.col = "white") +
  tm_legend(outside = TRUE)
```

Changing aesthetics

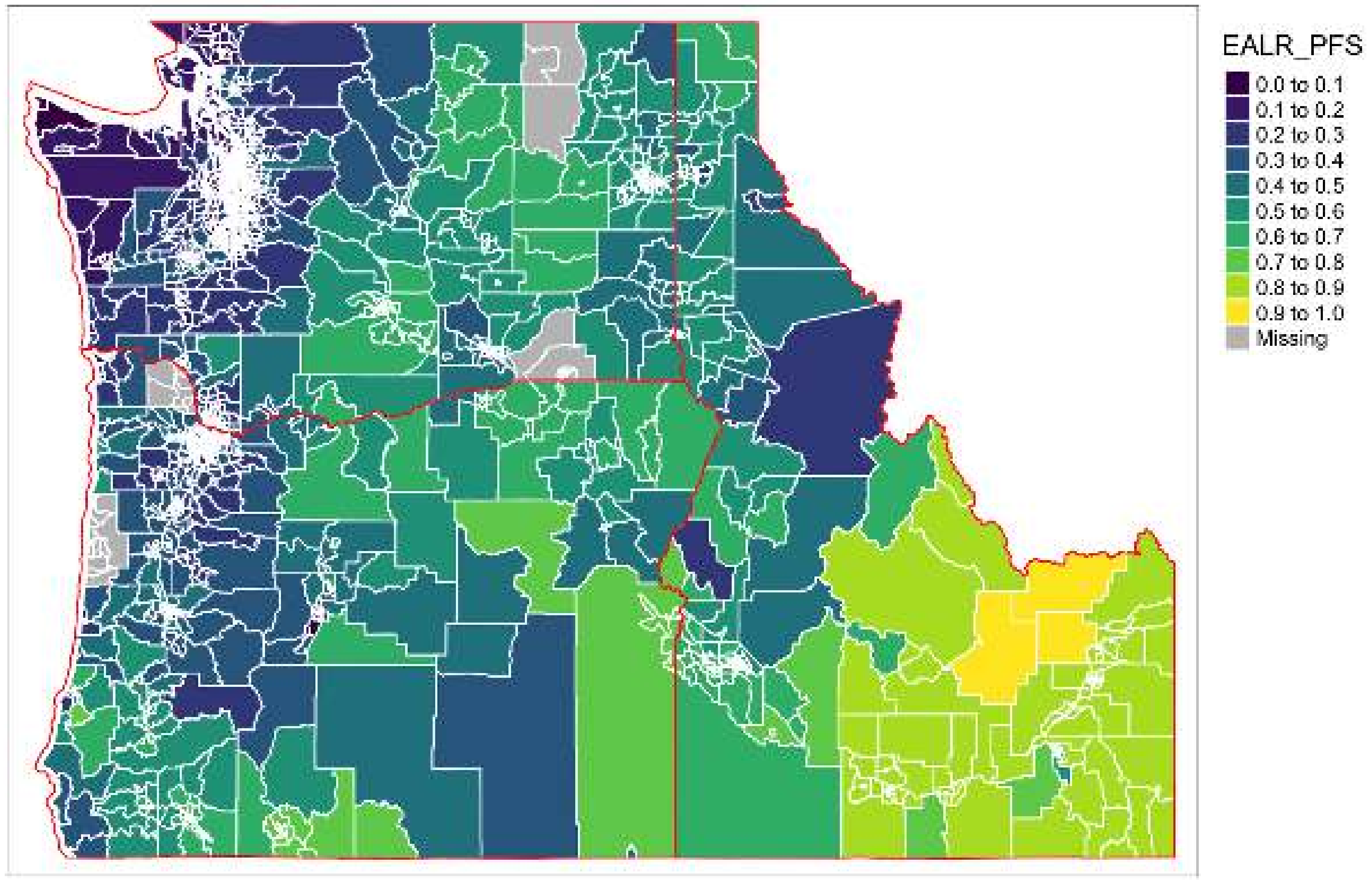


Adding layers

ORDER MATTERS

```r
st <- tigris::states(progress_bar=FALSE) %>% filter(STUSPS %in% c("ID", "WA
pt <- tm_shape(cejst) +
  tm_polygons(col = "EALR_PFS", n=10,palette=viridis(10),
                    border.col = "white") +
  tm_shape(st) +
  tm_borders("red") +
  tm_legend(outside = TRUE)
```

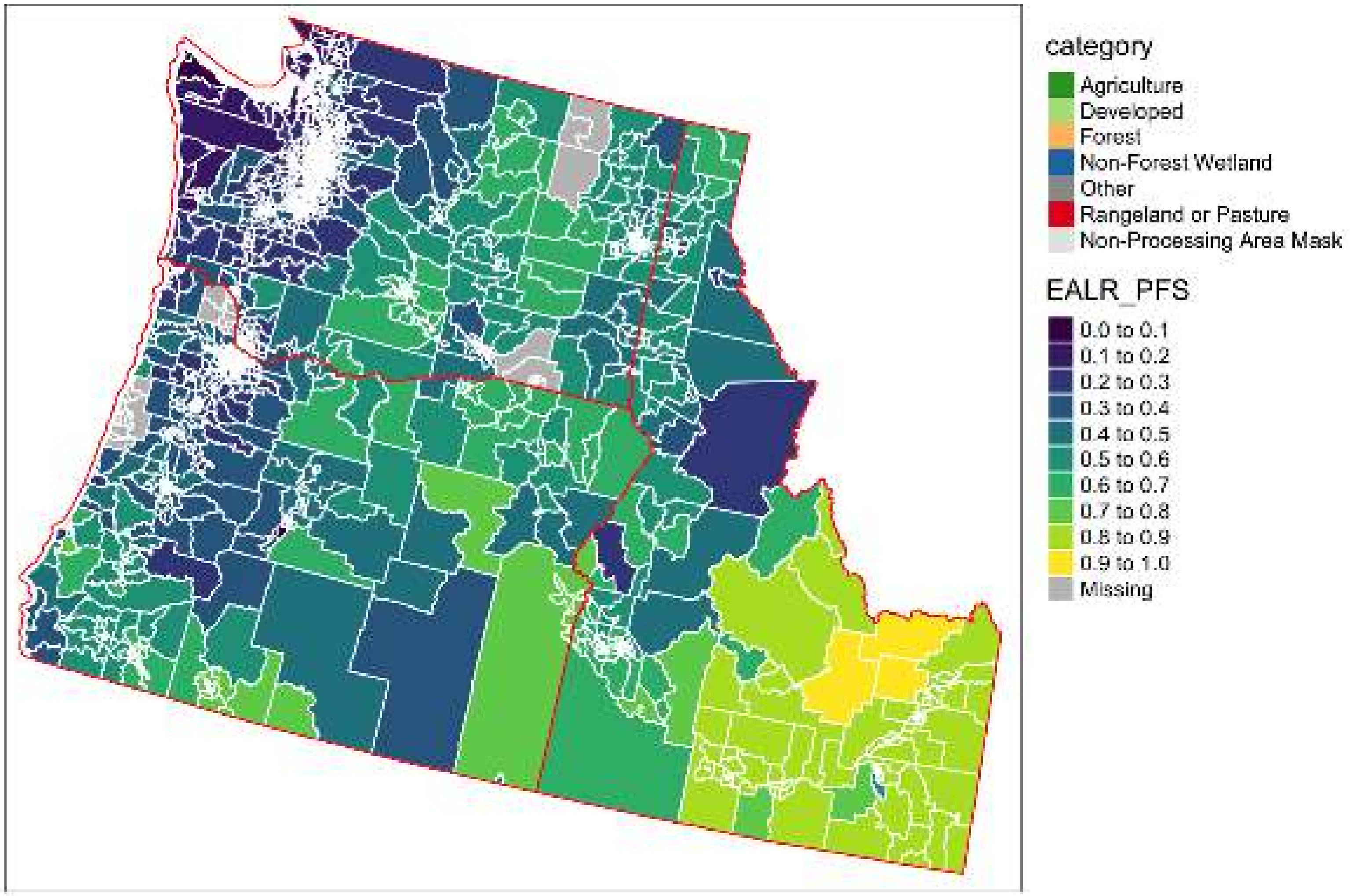
Adding layers



Integrating Rasters

```r
cejst.proj <- st_transform(cejst, crs=crs(rast.data)) %>% filter(!st_is_emp
states.proj <- st_transform(st, crs=crs(rast.data))
pal8 <- c("#33A02C", "#B2DF8A", "#FDBF6F", "#1F78B4", "#999999", "#E31A1C",
pt <- tm_shape(rast.data["category"]) +
  tm_raster(palette = pal8) +
  tm_shape(cejst.proj) +
  tm_polygons(col = "EALR_PFS", n=10,palette=viridis(10),
              border.col = "white") +
  tm_shape(states.proj) +
  tm_borders("red") +
  tm_legend(outside = TRUE)
```

Integrating Rasters


category
Agriculture
Developed
Forest
Non-Forest Wetland
Other
Rangeland or Pasture
Non-Processing Area Mask
EALR_PFS
0.0 to 0.1
0.1 to 0.2
0.2 to 0.3
0.3 to 0.4
0.4 to 0.5
0.5 to 0.6
0.6 to 0.7
0.7 to 0.8
0.8 to 0.9
0.9 to 1.0
Missing