Lab: Spatial data 1: Oregon Fires (Student Version)

Conservation/ecology Topics

- Explore how Oregon fires are changing due to fire suppression and climate change.
- Describe fundamental concepts in fire ecology, including fire severity.

Statistical Topics

- Describe the fundamental attributes of a raster dataset.

Computational Topics

- Explore raster attributes and metadata using R.
- Import rasters into R using the `terra` package.
- Plot raster files in R using the `ggplot2` package.
- Reproject raster and vector data
- Layer raster and vector data together

Lab part 1: reading in fire raster data and plotting

We will be working with the soil burn severity data from the 2020 Holiday Farm Fire (up the McKenzie E of Eugene), the 2020 Beachie Fire (near Portland) and the 2018 Terwilliger fire (up the McKenzie E of Eugene, near Cougar hotsprings).

We will use data downloaded from the USGS: https://burnseverity.cr.usgs.gov/products/baer

Specifically, BARC Fire Severity layers are created by first calculating spectral indices from pre- and post-fire satellite imagery that are sensitive to changes caused by fire. The two images are then subtracted showing the difference between them which is then binned into 4 burn severity classes (high, moderate, low, very low/unburned). Field crews ground-truth the severity classes.

The metadata files provide additional details on how the continuous data was binned into discrete catagories.

- 1a. Read in each fire severity rasters, name them [fire name]_rast. The .tif files are the rasters.

HINT: The files are nested within folders so be aware of your file paths.

```
holiday_rast <- rast("~/Downloads/ds-environ-main/labs/data/soil-burn-severity/2020_holidayfarm_sbs/Hol

beachie_rast <- rast("~/Downloads/ds-environ-main/labs/data/soil-burn-severity/2020_beachiecreek_sbs/Bea

terwilliger_rast <- rast("/Users/vgarfield03/Downloads/ds-environ-main/labs/data/soil-burn-severity/2018
```

- 1b. Summarize the values of the rasters. Take note of the labels associated with the data values because you will need it for plotting.

```
holiday_sum <- freq(holiday_rast)
beachie_sum <- freq(beachie_rast)
terwilliger_sum <- freq(terwilliger_rast)

holiday_sum
```

```
##   layer value   count
## 1     1     1   75666
## 2     1     2  427277
## 3     1     3 1084897
## 4     1     4  162930
## 5     1   127 1536190
```

```
beachie_sum
```

```
##   layer value   count
## 1     1     1   85614
## 2     1     2  640217
## 3     1     3 1021024
## 4     1     4  202022
## 5     1   127 2437627
```

```
terwilliger_sum
```

```
##   layer    value count
## 1     1 Unburned 11509
## 2     1      Low 33578
## 3     1 Moderate  5677
## 4     1     High   780
```

- 1c. Plot each raster.. Set the scale to be `scale_fill_brewer(palette = "Spectral", direction=-1)`

HINT: Remember we have to turn them into "data.frames" for ggplot to recognize them as plot-able.

```
rast_to_df <- function(r){
  as.data.frame(r, xy = TRUE) %>%
    rename(severity = names(r)) %>%
    mutate(severity = factor(severity,
                             levels = c(1,2,3,4),
                             labels = c("Unburned/Very Low", "Low", "Moderate", "High")))
}
```

HINT HINT: Remember to check the labels of the data values to be able to set the fill.

```
## dataframes for burns
holiday_df <- rast_to_df(holiday_rast)
beachie_df <- rast_to_df(beachie_rast)
terwilliger_df <- rast_to_df(terwilliger_rast)

holiday_rast_df <- as.data.frame(holiday_rast, xy = TRUE)
beachie_rast_df <- as.data.frame(beachie_rast, xy = TRUE)
terwilliger_rast_df <- as.data.frame(terwilliger_rast, xy = TRUE)


holiday_rast_df$Layer_1 <- as.factor(holiday_rast_df$Layer_1)
beachie_rast_df$Layer_1 <- as.factor(beachie_rast_df$Layer_1)
terwilliger_rast_df$SoilBurnSe <- as.factor(terwilliger_rast_df$SoilBurnSe)
```

```r
ggplot(holiday_df, aes(x = x, y = y, fill = severity)) +
geom_raster() +
  scale_fill_brewer(palette = "Spectral", direction = -1) +
  coord_equal() +
  labs(
    title = "holiday farm fire, burn severity",
    fill = "Severity") +
theme_classic()
```

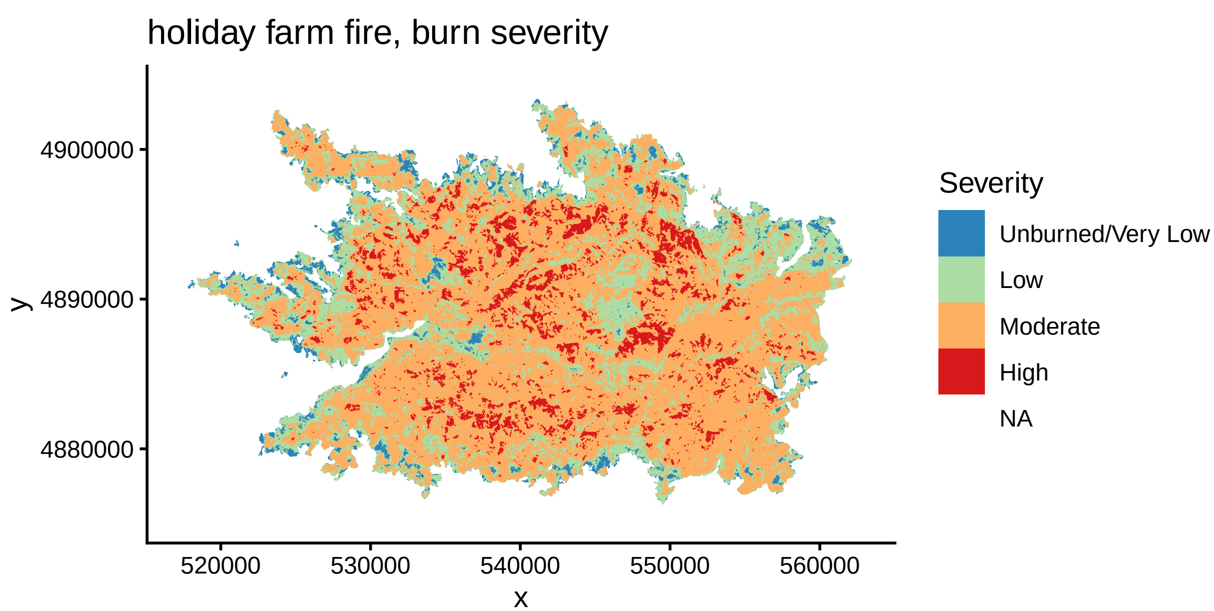


Figure 1: Holiday plot with ggplot2 using the Spectral color scale

```r
ggplot(beachie_df, aes(x = x, y = y, fill = severity)) +
  geom_raster() +
  scale_fill_brewer(palette = "Spectral", direction = -1) +
  coord_equal() +
  labs(
    title = "beachie fire, burn severity",
    fill = "Severity"
  ) +
theme_classic()
```

```r
ggplot(terwilliger_df, aes(x = x, y = y, fill = severity)) +
  geom_raster() +
```

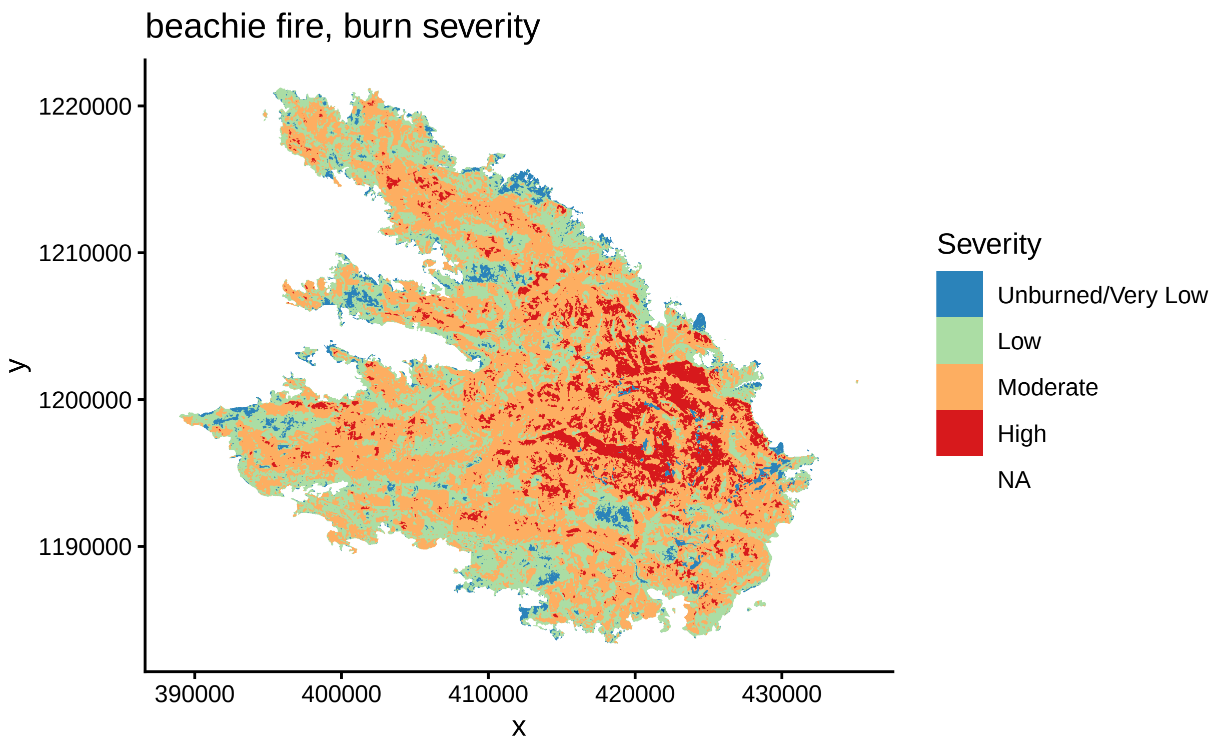


Figure 2: Beachie plot with ggplot2 using the Spectral color scale

```r
  scale_fill_brewer(palette = "Spectral", direction = -1) +
  coord_equal() +
  labs(
    title = "Terwilliger Fire, Burn Severity",
    fill = "Severity"
  ) +
  theme_classic()
```

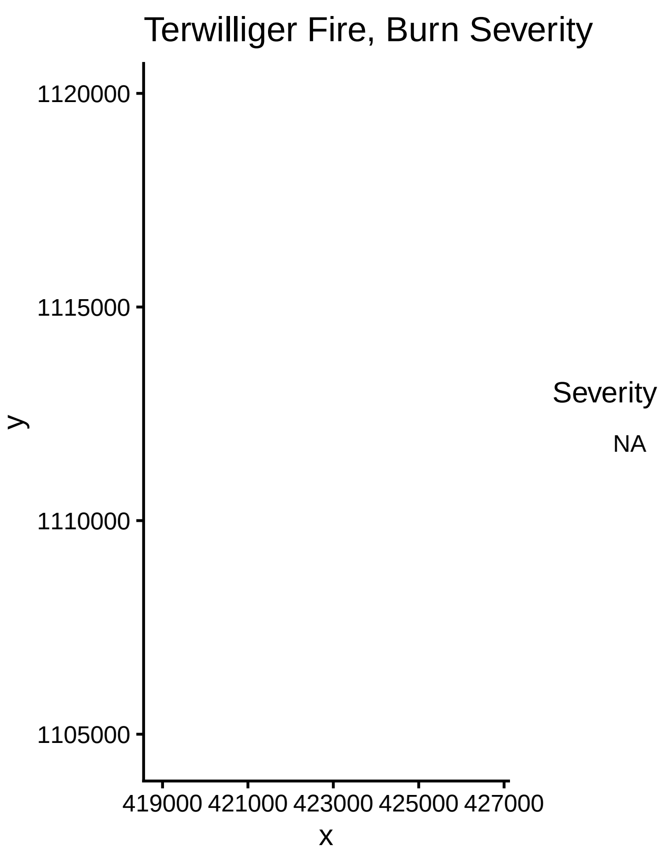


Figure 3: Terwilliger plot with ggplot2 using the Spectral color scale

- 1d. Compare these visualizations what is something you notice? -ANSWER: holiday had moderate-high severity. beachie has low to mod

Lab part 2: Exploring the attributes of our spatial data.

- 2a. What are the crs of the rasters? What are the units? Are they all the same?

```r
crs(holiday_rast)
```

```
## [1] "PROJCRS[\"NAD83 / UTM zone 10N\",\n    BASEGEOGCRS[\"NAD83\",\n        DATUM[\"North American Da
```

```r
crs(beachie_rast)
```

```
## [1] "PROJCRS[\"IMAGINE GeoTIFF Support\nERDAS IMAGINE 2014  14.0.0.181\nProjection = Albers Conical E
```

```r
crs(terwilliger_rast)
```

```
## [1] "PROJCRS[\"NAD83 / USFS R6 Albers\",\n    BASEGEOGCRS[\"NAD83\",\n        DATUM[\"North American
```

- ANSWER crs: Holiday: 26910 Beachie:8827 Terwilliger:9001
- ANSWER units: Holiday:metre Beachie:metre Terwilliger:metre
- ANSWER the same? : crs is different, length is the same
- 2b. What about the resolution of each raster?

```r
res(holiday_rast)
```

```
## [1] 20 20
```

```r
res(beachie_rast)
```

```
## [1] 20 20
```

```r
res(terwilliger_rast)
```

```
## [1] 30 30
```

- ANSWER resolution: Holiday: 20 Beachie: 20 Terwilliger:30
- ANSWER the same? not the same, teri is higher
- 2c. Calculate the min and max values of each raster. Are they all the same?

```r
minmax(holiday_rast)
```

```
##     Layer_1
## min       1
## max     127
```

```r
minmax(beachie_rast)
```

```
##     Layer_1
## min       1
## max     127
```

```r
minmax(terwilliger_rast)
```

```
##     SoilBurnSe
## min          1
## max          4
```

- ANSWER minmax: Holiday:1/127 Beachie:1/127 Terwilliger:1/4
- ANSWER the same? : teri. is lower max

Given we expect there to be 4 values for each bin of severity (high, moderate, low, very low/unburned), let's try to work out why there are values other than 1-4. After checking the metadata .txt and inspecting the metadata in the raster itself, I could not find an explicit mention of the meaning on the non 1-4 data (maybe you can?). Not great practices USGS! :(But it is likely missing data. Let's convert the Holiday data greater than 4 to NA, just like we would a regular matrix of data.

Uncomment the below.

```
holiday_rast[holiday_rast > 4] <- NA
summary(values(holiday_rast))
```

```
##     Layer_1
##  Min.   :1.00
##  1st Qu.:2.00
##  Median :3.00
##  Mean   :2.76
##  3rd Qu.:3.00
##  Max.   :4.00
##  NA's   :1536190
```

That's better :)

- 2d. Do the same conversion for Beachie.

```
beachie_rast[beachie_rast > 4] <- NA
summary(values(beachie_rast))
```

```
##     Layer_1
##  Min.   :1.00
##  1st Qu.:2.00
##  Median :3.00
##  Mean   :2.69
##  3rd Qu.:3.00
##  Max.   :4.00
##  NA's   :2437627
```

Lab part 3: Reprojection From our exploration above, the rasters are not in the same projection, so we will need to re-project them if we are going to be able to plot them together.

We can use the `project()` function to reproject a raster into a new CRS. The syntax is `project(RasterObject, crs)`

- 3a. First we will reproject our `beachie_rast` raster data to match the `holidat_rast` CRS. If the resolution is different, change it to match Holiday's resolution.

Don't change the name from beachie_rast.

```r
beachie_rast <- project(beachie_rast, holiday_rast)

## This should return TRUE
crs(beachie_rast, proj = TRUE) == crs(holiday_rast, proj = TRUE)
```

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

```r
same.crs(beachie_rast, holiday_rast)
```

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

- 3b. Now convert the Terwilliger crs to the holiday crs. If the resolution is different, change it to match Holiday's resolution.

```r
terwilliger_rast <- project(beachie_rast, holiday_rast)
# This should return TRUE TRUE
crs(terwilliger_rast, proj = TRUE) == crs(holiday_rast, proj = TRUE)
```

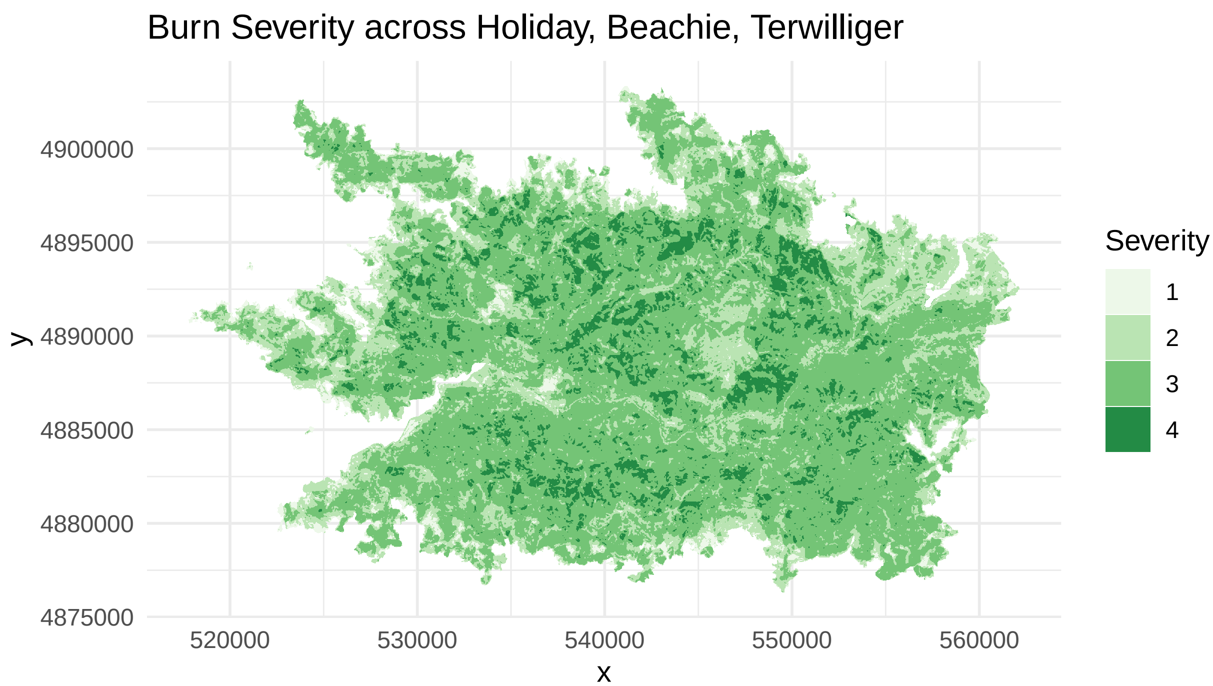
```
## [1] TRUE
```

```r
res(terwilliger_rast)[2] == res(holiday_rast)[2]
```

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

- 3c. Now you can plot all of the fires on the same map! HINT: Remember to re-make the dataframes.

```r
holiday_df <- as.data.frame(holiday_rast, xy = TRUE)
beachie_df <- as.data.frame(beachie_rast, xy = TRUE)
terwilliger_df <- as.data.frame(terwilliger_rast, xy = TRUE)

holiday_df$severity <- as.factor(holiday_df$Layer_1)
beachie_df$severity <- as.factor(beachie_df$Layer_1)
terwilliger_df$severity <- as.factor(terwilliger_df$Layer_1)

holiday_df$fire <- "holiday farm"
beachie_rast_df$fire <- "beachie fire"
terwilliger_rast_df$fire <- "terwilliger fire"

all_fires_df <- rbind(holiday_df, beachie_df, terwilliger_df)

ggplot(all_fires_df, aes(x = x, y = y, fill = severity)) +
  geom_tile() +
  coord_equal() +
scale_fill_brewer(palette = "Spectral, direction = -1") +
   labs(
      title = "Burn Severity across Holiday, Beachie, Terwilliger",
      fill = "Severity"
    ) +
    theme_minimal()
```



Well that's annoying. It appears as though in 2018 the makers of these data decided to give 1,2,3,4 categorical names which are being interpreted as two different scales. If we look at the terwilliger_rast values we can see that in min max.

```
terwilliger_rast$Layer_1
```

```
## class       : SpatRaster
## size        : 1448, 2270, 1  (nrow, ncol, nlyr)
## resolution  : 20, 20  (x, y)
## extent      : 517347.8, 562747.8, 4875154, 4904114  (xmin, xmax, ymin, ymax)
## coord. ref. : NAD83 / UTM zone 10N (EPSG:26910)
## source(s)   : memory
## varname     : HolidayFarm_SBS_final
## name        : Layer_1
## min value   :     NaN
## max value   :     NaN
```

- 3d. Let's deal with the the easy way and modify the dataframe. Convert High to 4, Moderate to 3, Low to 2, and Unburned to 1 using your data subsetting skills.

Somethings you will need to be careful of: - If you check the class of terwilliger_rast_df$SoilBurnSe it is a factor, which is a special class of data that are ordered categories with specific levels. R will not let you convert add a level. So first, convert the data to characters (using as.character()). - Now the data are characters, so you will not be able to add in numerics. So code the 1,2,3 as characters i.e., "1", "2"… - We will eventually want the data to be factors again so it will match up with the other rasters. So lastly, convert the data to a factor (using as.factor()).

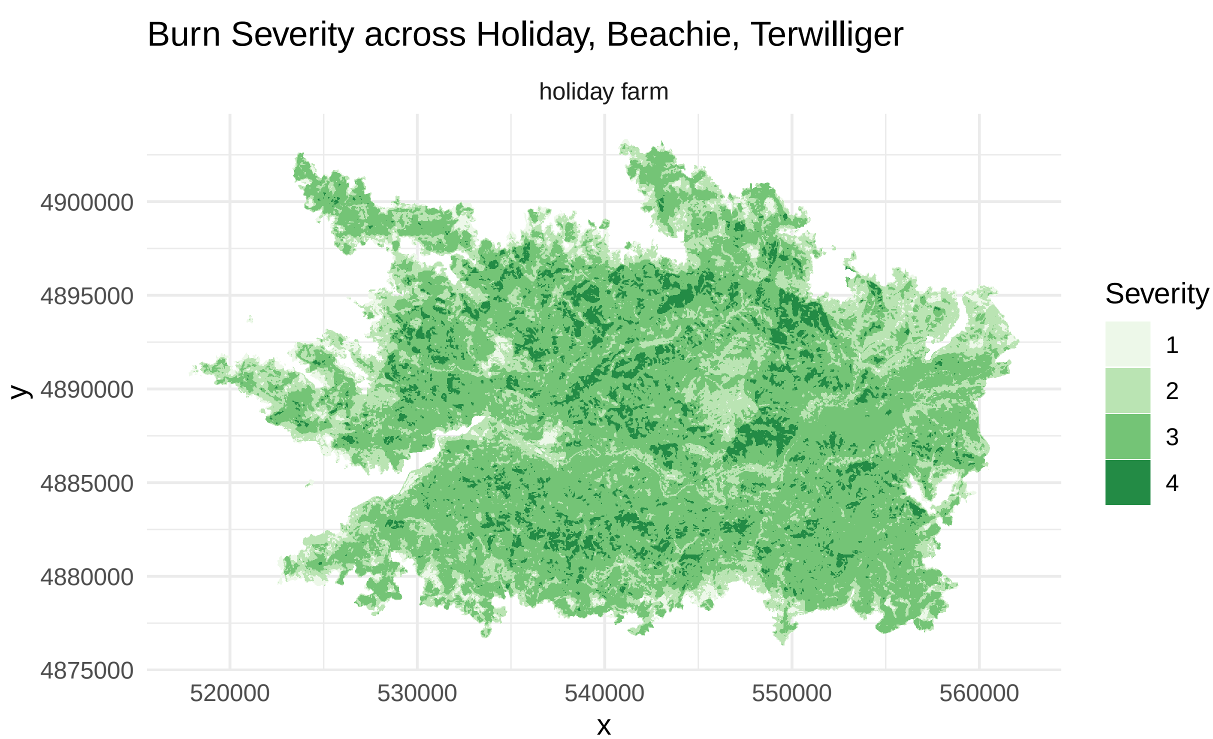
```r
class(terwilliger_rast_df$SoilBurnSe)
```

```
## [1] "factor"
```

```r
terwilliger_df$severity <- as.character(terwilliger_df$severity)

terwilliger_df$severity[terwilliger_df$severity == "Unburned"] <- "1"
terwilliger_df$severity[terwilliger_df$severity == "Low"]      <- "2"
terwilliger_df$severity[terwilliger_df$severity == "Moderate"] <- "3"
terwilliger_df$severity[terwilliger_df$severity == "High"]     <- "4"

terwilliger_df$severity <- as.factor(terwilliger_df$severity)
```

- 3e. Try plotting again.

```r
ggplot(all_fires_df, aes(x = x, y = y, fill = severity)) +
  geom_tile() +
  coord_equal() +
scale_fill_brewer(palette = "Spectral, direction = -1") +
  facet_wrap(~ fire) +
   labs(
      title = "Burn Severity across Holiday, Beachie, Terwilliger",
      fill = "Severity"
    ) +
    theme_minimal()
```



The scale bar make sense! It would be nice to have a baselayer map to see where is Oregon these fires are.

Lab part 4: Adding in vector data

I found a nice ecoregion map on the OR spatial data website. https://spatialdata.oregonexplorer.info/geoportal/details;id=3c7862c4ae664993ad1531907b1e413e

- 4a. Load the data into R, it is in the OR-ecoregions folder.

```
oreco <- st_read("~/Downloads/ds-environ-main/ds-environ/labs/data/OR-ecoregions/Ecoregions_OregonConse
```

```
## Reading layer ‘Ecoregions_OregonConservationStrategy’ from data source
##   ‘/Users/vgarfield03/Downloads/ds-environ-main/ds-environ/labs/data/OR-ecoregions/Ecoregions_Oregon
##   using driver ‘ESRI Shapefile’
## Simple feature collection with 9 features and 6 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 183871.7 ymin: 88600.88 xmax: 2345213 ymax: 1675043
## Projected CRS: NAD83 / Oregon GIC Lambert (ft)
```

- 4b. Check the projection and re-project if needed. We did not cover this in the lecture demo, but for vector data, use st_transform()

```
st_crs(oreco)
```

```
## Coordinate Reference System:
##   User input: NAD83 / Oregon GIC Lambert (ft)
##   wkt:
## PROJCRS["NAD83 / Oregon GIC Lambert (ft)",
##     BASEGEOGCRS["NAD83",
##         DATUM["North American Datum 1983",
##             ELLIPSOID["GRS 1980",6378137,298.257222101,
##                 LENGTHUNIT["metre",1]]],
##         PRIMEM["Greenwich",0,
##             ANGLEUNIT["degree",0.0174532925199433]],
##         ID["EPSG",4269]],
##     CONVERSION["Oregon GIC Lambert (international foot)",
##         METHOD["Lambert Conic Conformal (2SP)",
##             ID["EPSG",9802]],
##         PARAMETER["Latitude of false origin",41.75,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8821]],
##         PARAMETER["Longitude of false origin",-120.5,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8822]],
##         PARAMETER["Latitude of 1st standard parallel",43,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8823]],
##         PARAMETER["Latitude of 2nd standard parallel",45.5,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8824]],
##         PARAMETER["Easting at false origin",1312335.958,
##             LENGTHUNIT["foot",0.3048],
```

```
##             ID["EPSG",8826]],
##         PARAMETER["Northing at false origin",0,
##             LENGTHUNIT["foot",0.3048],
##             ID["EPSG",8827]]],
##     CS[Cartesian,2],
##         AXIS["easting (X)",east,
##             ORDER[1],
##             LENGTHUNIT["foot",0.3048]],
##         AXIS["northing (Y)",north,
##             ORDER[2],
##             LENGTHUNIT["foot",0.3048]],
##     USAGE[
##         SCOPE["State-wide spatial data management."],
##         AREA["United States (USA) - Oregon."],
##         BBOX[41.98,-124.6,46.26,-116.47]],
##     ID["EPSG",2992]]
```

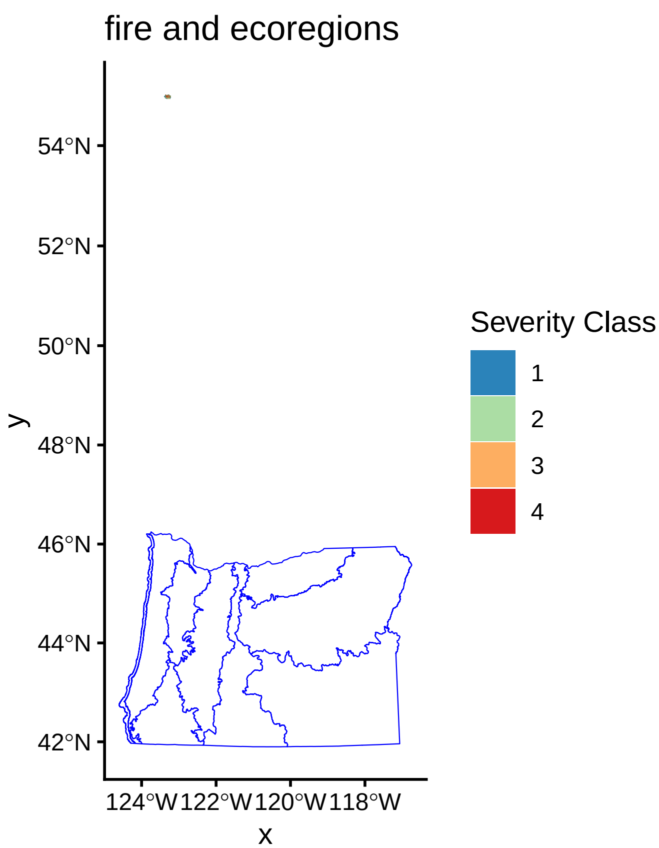
```r
reprojected_oreco <- st_transform(oreco, crs = 2992)

st_crs(reprojected_oreco)
```

```
## Coordinate Reference System:
##   User input: EPSG:2992
##   wkt:
## PROJCRS["NAD83 / Oregon GIC Lambert (ft)",
##     BASEGEOGCRS["NAD83",
##         DATUM["North American Datum 1983",
##             ELLIPSOID["GRS 1980",6378137,298.257222101,
##                 LENGTHUNIT["metre",1]]],
##         PRIMEM["Greenwich",0,
##             ANGLEUNIT["degree",0.0174532925199433]],
##         ID["EPSG",4269]],
##     CONVERSION["Oregon GIC Lambert (international foot)",
##         METHOD["Lambert Conic Conformal (2SP)",
##             ID["EPSG",9802]],
##         PARAMETER["Latitude of false origin",41.75,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8821]],
##         PARAMETER["Longitude of false origin",-120.5,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8822]],
##         PARAMETER["Latitude of 1st standard parallel",43,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8823]],
##         PARAMETER["Latitude of 2nd standard parallel",45.5,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8824]],
##         PARAMETER["Easting at false origin",1312335.958,
##             LENGTHUNIT["foot",0.3048],
##             ID["EPSG",8826]],
##         PARAMETER["Northing at false origin",0,
##             LENGTHUNIT["foot",0.3048],
##             ID["EPSG",8827]]],
##     CS[Cartesian,2],
```

```
##         AXIS["easting (X)",east,
##             ORDER[1],
##             LENGTHUNIT["foot",0.3048]],
##         AXIS["northing (Y)",north,
##             ORDER[2],
##             LENGTHUNIT["foot",0.3048]],
##     USAGE[
##         SCOPE["State-wide spatial data management."],
##         AREA["United States (USA) - Oregon."],
##         BBOX[41.98,-124.6,46.26,-116.47]],
##     ID["EPSG",2992]]
```

- 4c. Plot all of the data together (the rasters and vector data). You can layer on geom_sf into ggplot with the other rasters just like you would add another raster.

```r
# rasters
ggplot() +
  geom_tile(
    data = all_fires_df,
    aes(x = x, y = y, fill = severity)
  ) +
# ecoregions
geom_sf(
  data = reprojected_oreco,
  fill = NA,
  color = "blue"
) +
  scale_fill_brewer(palette = "Spectral", direction = -1) +
 coord_sf() +
  theme_classic() +
  labs(
    title = "fire and ecoregions",
    fill = "Severity Class"
  )
```



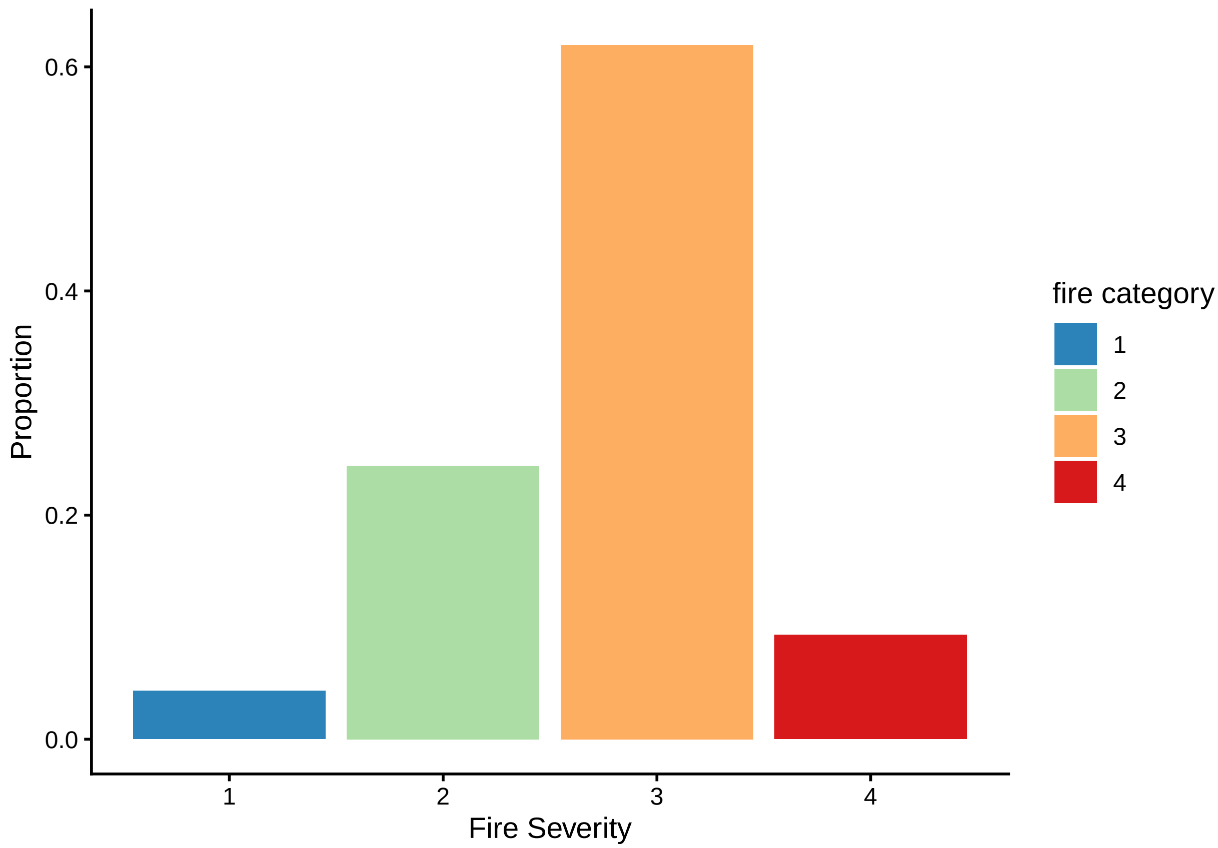
We could get fancy and zoom into the correct region using extent, which we will cover next week. For now, this looks pretty good.

Lab part 5: Exploring patterns of fire severity

- 5a. Create a barplot with the count of each fire severity category.
- Use scale_fill_brewer(palette = "Spectral", direction=-1) to get the bars to match the maps.
- Plot the proportion on the y. To do this, in geom_bar, include y = (..count..)/sum(..count..). EX: aes(x= Layer_1, y = (..count..)/sum(..count..)

HINT: Rather annoyingly, you will need to convert the layer values to factors again to get fill to recognize them. EX: fill=as.factor(Layer_1)

```
ggplot(all_fires_df, aes(
  x = (Layer_1),
  y = (..count..) / sum(..count..),
  fill = as.factor(Layer_1)
)) +
  geom_bar() +
  scale_fill_brewer(palette = "Spectral", direction = -1) +
  ylab("Proportion") +
  xlab("Fire Severity") +
  labs(fill = "fire category") +
  theme_classic()
```



-5b. What do you notice about the frequency of different severity classes when you compare these barplots. How does this relate to the Haldofsky reading? ANSWER: It shows that the proportion of cat. 3 fires is significantly higher that any of the fire categories, cat. 2 fires are up there as well, but the positive part is that there is a lower amount of cat.4 fires. The concern as well is that cat. 4 fires are higher cat. 1. This relates to Haldofsky's reading because the concern in haldofsky is how climate change will increase fire severity, and coverage. I feel llike looking at this graph we see an increase in fire severity. this is a concern in my opinion, that fire cat.1 is the lowest out of the four. I feel like looking at this graph we can see an increase in fire severity due to climate change.

Also, if the legend label bothers you (as it does for me) Check out this tutorial: https://www.datanovia.com/en/blog/ggplot-legend-title-position-and-labels/