Lab 5 Oregon Fires

Lauren Ponisio

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.

a. 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.

```
terwilliger_rast <- rast("soil-burn-severity/2018_terwilliger_sbs/SoilSeverity.tif")
beachie_rast <- rast("soil-burn-severity/2020_beachiecreek_sbs/BeachieCreek_SBS_final.tif")
holiday_rast <-  rast("soil-burn-severity/2020_holidayfarm_sbs/HolidayFarm_SBS_final.tif")
```

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

```
summary(terwilliger_rast)
```

```
## Warning: [summary] used a sample
```

```
##       SoilBurnSe
##  Unburned: 8801
##  Low     :25507
##  Moderate: 4337
##  High    :  586
##  NA's    :61113
```

```
summary(beachie_rast)
```

```
## Warning: [summary] used a sample
```

```
##       Layer_1
##  3       :23275
##  2       :14608
##  4       : 4623
##  1       : 1969
##  0       :    0
##  (Other):    0
##  NA's    :55625
```

```
summary(holiday_rast)
```

```
## Warning: [summary] used a sample
```

```
##       Layer_1
##  3       :33163
##  2       :12950
##  4       : 4933
##  1       : 2305
##  0       :    0
##  (Other):    0
##  NA's    :46837
```

c. 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.

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

```
#turn into dataframe
df_holidayfarm <- as.data.frame(holiday_rast, xy = TRUE)

ggplot() +
    geom_raster(data = df_holidayfarm , aes(x = x, y = y, fill = Layer_1)) +
```

```
    scale_fill_brewer(palette = "Spectral", direction=-1,
  name = "Severity",
    labels = c("Very low/ unburned", "Low", "Moderate", "High")
  ) +
  ggtitle("Holiday plot with ggplot2 using the Spectral color scale") +
  xlab("latitude") +
  ylab("longitude")
```

Warning: Raster pixels are placed at uneven horizontal intervals and will be shifted
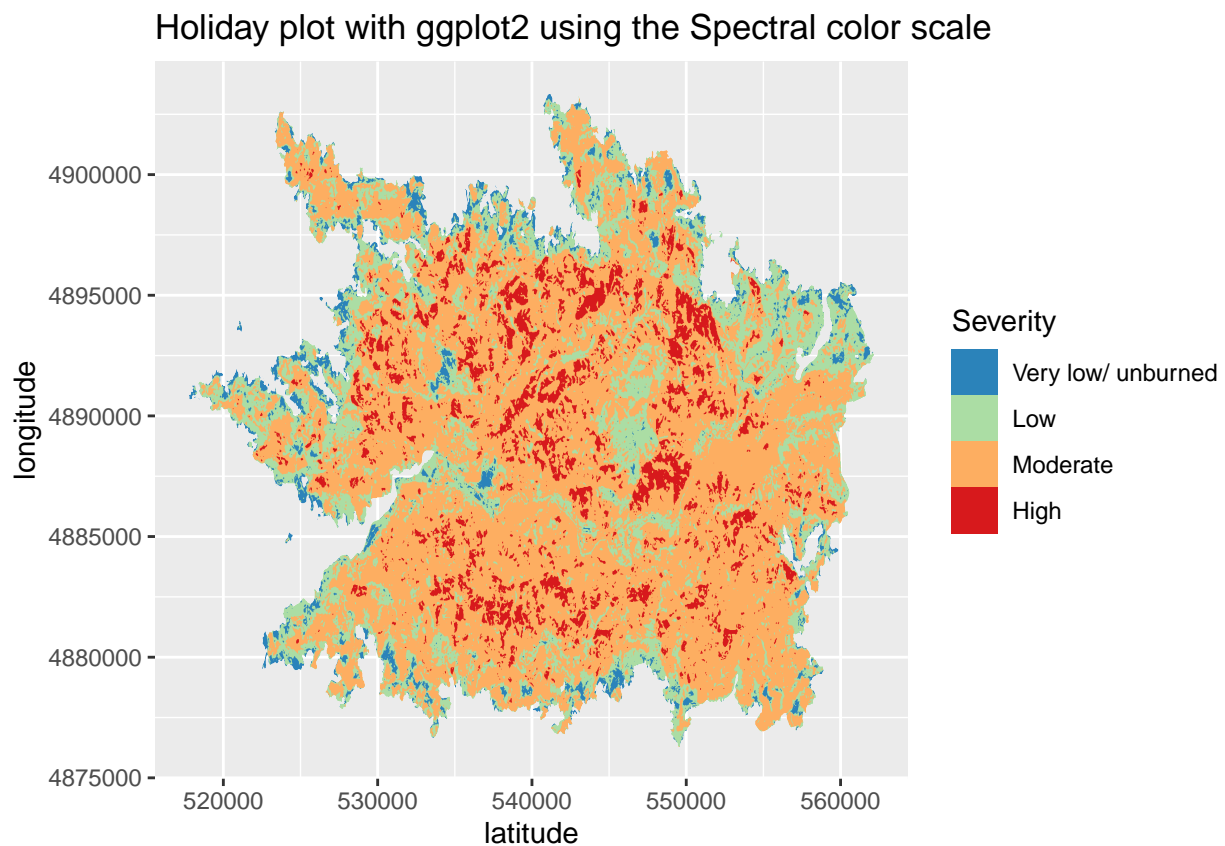## i Consider using `geom_tile()` instead.

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

```
#turn into dataframe
df_beachiecreek <- as.data.frame(beachie_rast, xy = TRUE)

ggplot() +
    geom_raster(data = df_beachiecreek , aes(x = x, y = y, fill = Layer_1)) +
    scale_fill_brewer(palette = "Spectral", direction=-1,
  name = "Severity",
    labels = c("Very low/ unburned", "Low", "Moderate", "High")
  ) +
  ggtitle("Beachie plot with ggplot2 using the Spectral color scale") +
```

3

```
    xlab("latitude") +
    ylab("longitude")
```

```
## Warning: Raster pixels are placed at uneven horizontal intervals and will be shifted
## i Consider using 'geom_tile()' instead.
```
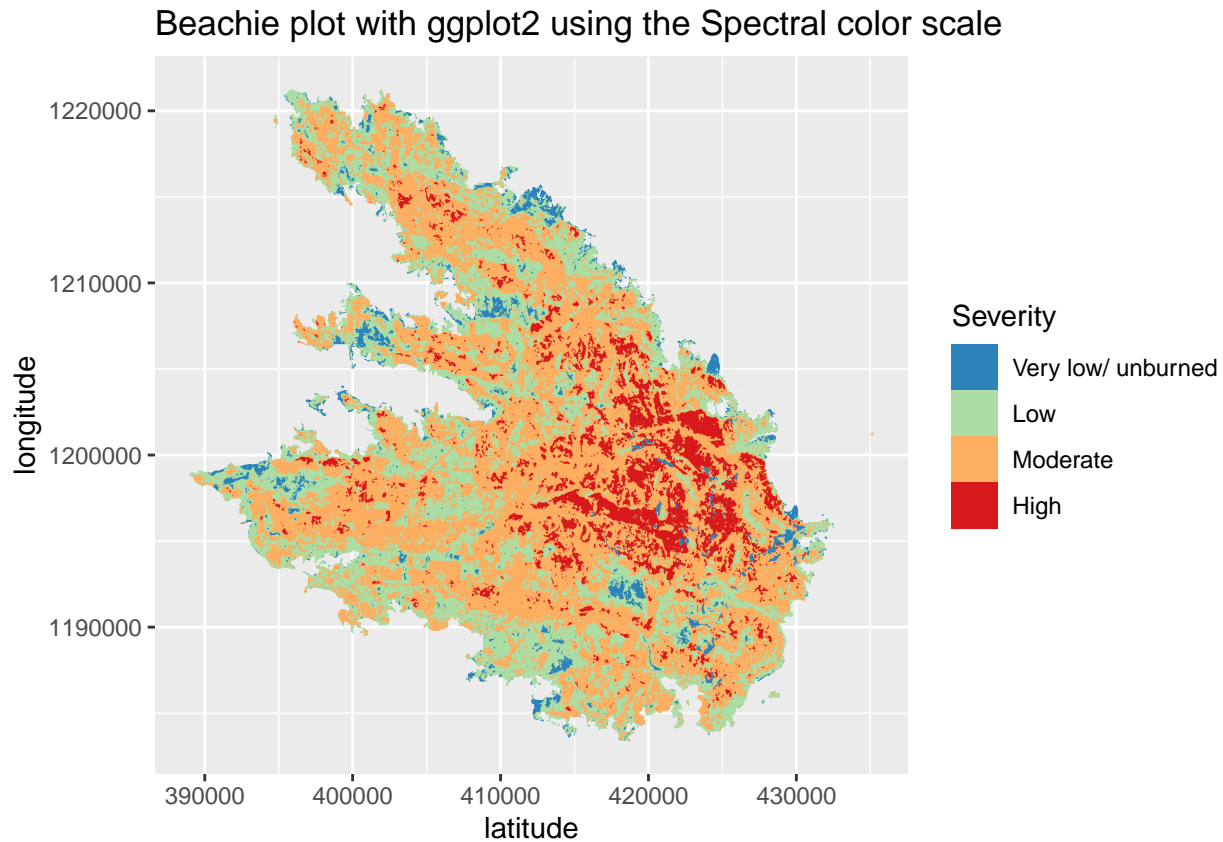

Figure 2: Beachie plot with ggplot2 using the Spectral color scale

```
#turn into dataframe
df_terwilliger <- as.data.frame(terwilliger_rast, xy = TRUE)

#plot
ggplot() +
    geom_raster(data = df_terwilliger , aes(x = x, y = y, fill = SoilBurnSe)) +
    scale_fill_brewer(palette = "Spectral", direction=-1,
                      name = "Severity",
    labels = c("Very low/ unburned", "Low", "Moderate", "High")
  ) +
    ggtitle("Terwilliger plot with ggplot2 using the Spectral color scale") +
    xlab("latitude") +
    ylab("longitude")
```

d. Compare these visualizations what is something you notice? -ANSWER: I notice that the Terwilliger plot has muxh lower severity burns than the Beachie Creek or Holiday farm sites. The Holiday farm
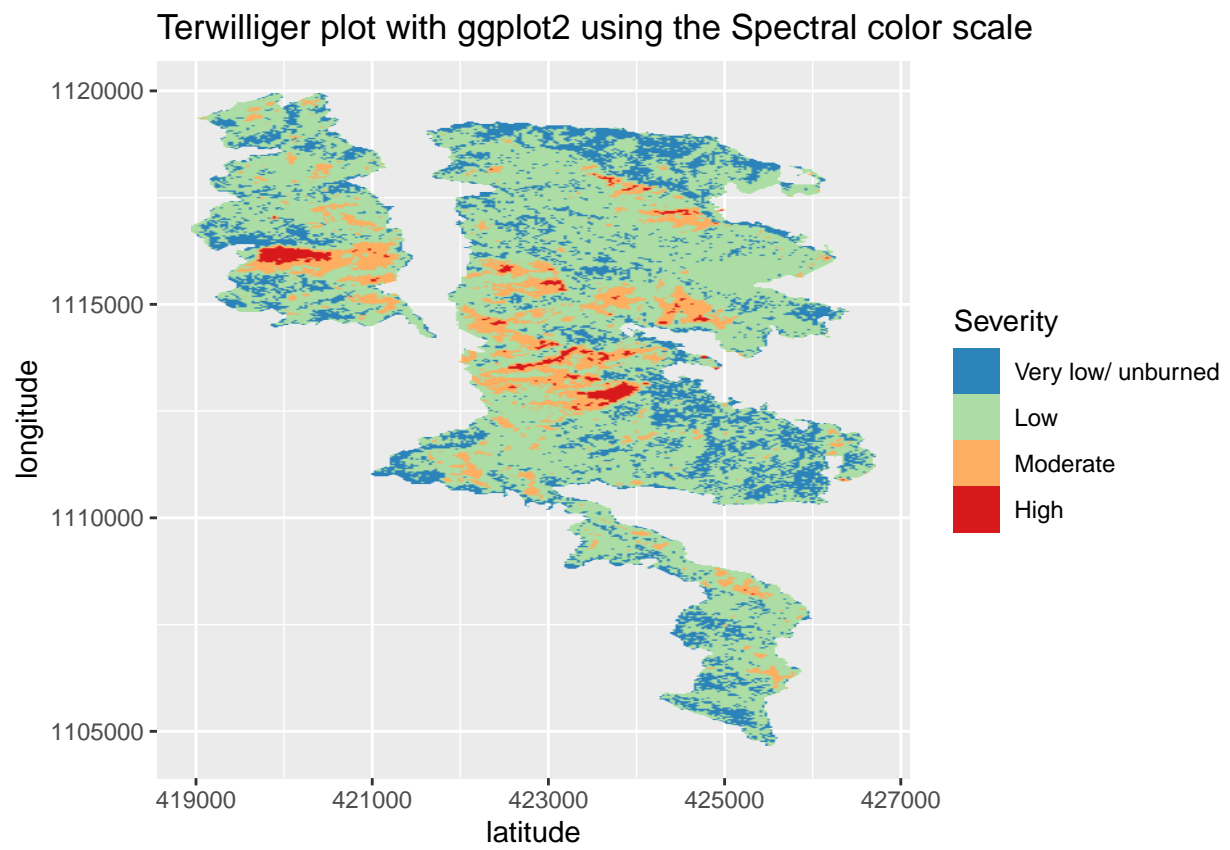
Figure 3: Terwilliger plot with ggplot2 using the Spectral color scale

fire and the beachie creek fire seem to both have high severity burns across their plot. I also notice that the shapes of the three fires are all very different.

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

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

```
crs(terwilliger_rast, proj = TRUE)
```

```
## [1] "+proj=aea +lat_0=34 +lon_0=-120 +lat_1=43 +lat_2=48 +x_0=600000 +y_0=0 +datum=NAD83 +units=m +n
```

```
crs(beachie_rast, proj = TRUE)
```

```
## [1] "+proj=aea +lat_0=34 +lon_0=-120 +lat_1=43 +lat_2=48 +x_0=600000 +y_0=0 +datum=NAD83 +units=m +n
```

```
crs(holiday_rast, proj = TRUE)
```

```
## [1] "+proj=utm +zone=10 +datum=NAD83 +units=m +no_defs"
```

- ANSWER crs: Holiday:NAD83, utm Beachie:NAD83, aea Terwilliger:NAD83, aea
- ANSWER units: Holiday:meters Beachie:meters Terwilliger:meters
- ANSWER the same? : units and datum are the same, projections are different

b. What about the resolution of each raster?

```
res(terwilliger_rast)
```

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

```
res(beachie_rast)
```

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

```
res(holiday_rast)
```

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

- ANSWER resolution: Holiday: 20x20 meters Beachie: 20 x 20 meters Terwilliger: 30 x 30 meters
- ANSWER the same? : Not the same

c. Calculate the min and max values of each raster. Are they all the same?

```
minmax(holiday_rast)
```

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

```
minmax(beachie_rast)
```

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

```
minmax(terwilliger_rast)
```

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

- ANSWER minmax: Holiday: 1/127 Beachie: 1/127 Terwilliger: 1/4
- ANSWER the same? : Not the same

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.

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

```
##     Layer_1
##  Min.   :1.0
##  1st Qu.:2.0
##  Median :3.0
##  Mean   :2.8
##  3rd Qu.:3.0
##  Max.   :4.0
##  NA's   :1536190
```

That's better :)

d. Do the same conversion for Beachie.

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

```
##     Layer_1
##  Min.   :1.0
##  1st Qu.:2.0
##  Median :3.0
##  Mean   :2.7
##  3rd Qu.:3.0
##  Max.   :4.0
##  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)`

    a. 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,
                        crs(holiday_rast))

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

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

    b. 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(terwilliger_rast,
                            crs(holiday_rast), res = res(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
```

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

```r
# convert to data.frames
df_terwilliger <- as.data.frame(terwilliger_rast, xy = TRUE)
df_holiday <- as.data.frame(holiday_rast, xy = TRUE)
df_beachie <- as.data.frame(beachie_rast, xy = TRUE)


#create plot

ggplot() +
    geom_raster(data = df_terwilliger,
                aes(x = x, y = y,
                    fill = SoilBurnSe)
```

```
                ) +
    geom_raster(data = df_holiday,
            aes(x = x, y = y,
                fill = Layer_1)
            ) +
  geom_raster(data = df_beachie,
            aes(x = x, y = y,
                fill = Layer_1)
            ) +
    scale_fill_brewer(palette = "Spectral", direction=-1) +
  ggtitle("Beachie Creek, Terwilliger, and Holiday Farm Fire plot with ggplot2 using \n the Spectral c
```
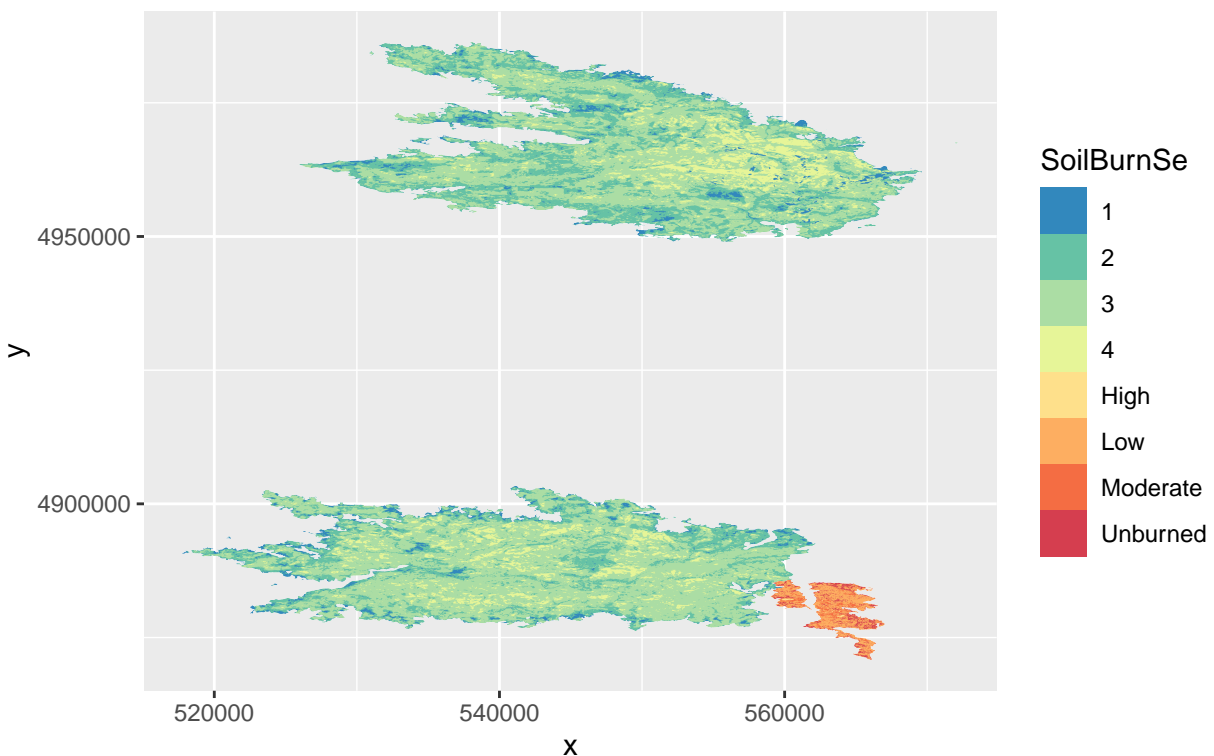
```
## Warning: Raster pixels are placed at uneven horizontal intervals and will be shifted
## i Consider using 'geom_tile()' instead.
## Raster pixels are placed at uneven horizontal intervals and will be shifted
## i Consider using 'geom_tile()' instead.
```


Beachie Creek, Terwilliger, and Holiday Farm Fire plot with ggplot2 usin
the Spectral color scale

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.

```
minmax(terwilliger_rast$SoilBurnSe)
```

```
##      SoilBurnSe
```

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

```
minmax(terwilliger_rast)
```

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

d. 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()).

```
#convert terwilliger from factor to characters
df_terwilliger$SoilBurnSe <- as.character(df_terwilliger$SoilBurnSe)

#modify the dataframe

df_terwilliger$SoilBurnSe[df_terwilliger$SoilBurnSe == "High"] <- "4"
df_terwilliger$SoilBurnSe[df_terwilliger$SoilBurnSe == "Moderate"] <- "3"
df_terwilliger$SoilBurnSe[df_terwilliger$SoilBurnSe == "Low"] <- "2"
df_terwilliger$SoilBurnSe[df_terwilliger$SoilBurnSe == "Unburned"] <- "1"

#convert back to factors
df_terwilliger$SoilBurnSe <- as.factor(df_terwilliger$SoilBurnSe)
```

e. Try plotting again.
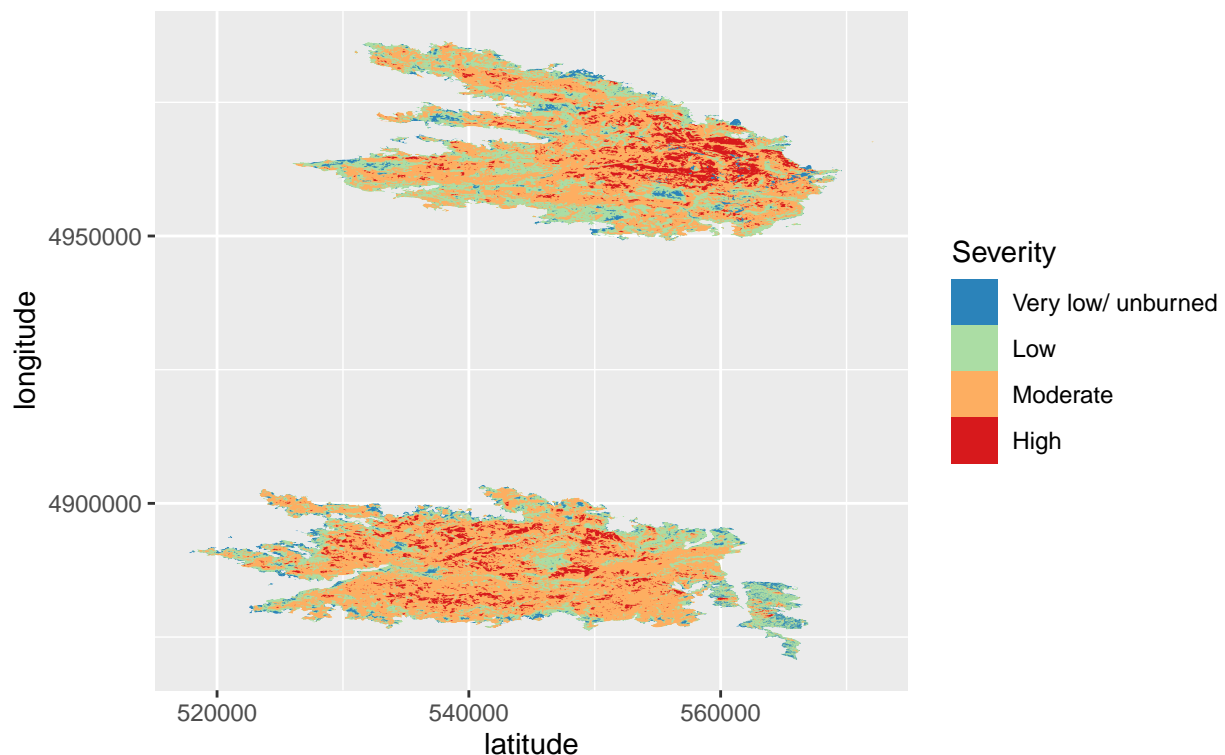
```
#create plot

ggplot() +
    geom_raster(data = df_terwilliger,
                aes(x = x, y = y,
                   fill = SoilBurnSe)
                ) +
    geom_raster(data = df_holiday,
              aes(x = x, y = y,
                  fill = Layer_1)
              ) +
  geom_raster(data = df_beachie,
              aes(x = x, y = y,
                  fill = Layer_1)
              ) +
    scale_fill_brewer(palette = "Spectral", direction=-1,
                      name = "Severity",
    labels = c("Very low/ unburned", "Low", "Moderate", "High")
```

```
  ) +
  ggtitle("Beachie Creek, Terwilliger, and Holiday Farm Fire plot with ggplot2 using \n the Spectral co
  xlab("latitude") +
  ylab("longitude")
```

```
## Warning: Raster pixels are placed at uneven horizontal intervals and will be shifted
## i Consider using `geom_tile()` instead.
## Raster pixels are placed at uneven horizontal intervals and will be shifted
## i Consider using `geom_tile()` instead.
```


Beachie Creek, Terwilliger, and Holiday Farm Fire plot with ggplot2 usin
the Spectral color scale

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

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

```
ecoregions_shp <- st_read("OR-ecoregions/Ecoregions_OregonConservationStrategy.shp")
```

```
## Reading layer 'Ecoregions_OregonConservationStrategy' from data source
##    '/Users/holly/Desktop/DataSciEcol/sumblime_repos/ds-environ-HA/5-OR-fires/OR-ecoregions/Ecoregions_
```

```
##    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)
```

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

```
crs(ecoregions_shp, proj=TRUE)
```

```
## [1] "+proj=lcc +lat_0=41.75 +lon_0=-120.5 +lat_1=43 +lat_2=45.5 +x_0=399999.9999984 +y_0=0 +datum=NA
```

```
ecoregions_shp <- st_transform(ecoregions_shp,
                               crs(holiday_rast))

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

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

c. 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.

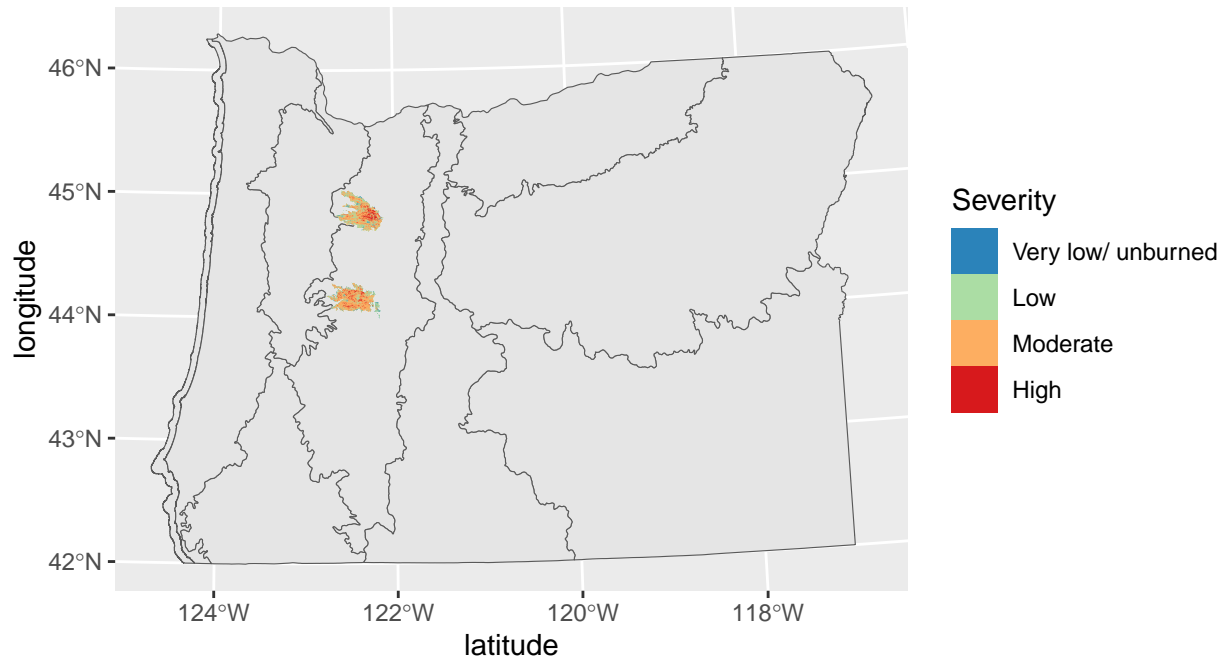```
ggplot() +
  #shapefile
   geom_sf(data = ecoregions_shp) +

  #rasters
    geom_raster(data = df_terwilliger,
                aes(x = x, y = y,
                    fill = SoilBurnSe)) +
    geom_raster(data = df_holiday,
            aes(x = x, y = y,
                    fill = Layer_1)) +
  geom_raster(data = df_beachie,
            aes(x = x, y = y,
                    fill = Layer_1)) +

  #aesthetics
    scale_fill_brewer(palette = "Spectral", direction=-1,
                        name = "Severity",
    labels = c("Very low/ unburned", "Low", "Moderate", "High")
) +
   ggtitle("Beachie Creek, Terwilliger, and Holiday Farm Fire plot with ggplot2 using \n the Spectral c
  xlab("latitude") +
  ylab("longitude")
```

```
## Warning: Raster pixels are placed at uneven horizontal intervals and will be shifted
## i Consider using 'geom_tile()' instead.
## Raster pixels are placed at uneven horizontal intervals and will be shifted
## i Consider using 'geom_tile()' instead.
```

Beachie Creek, Terwilliger, and Holiday Farm Fire plot with ggplot2 using the Spectral color scale

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

a. 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)

```
#terwilliger
ggplot(data = df_terwilliger, aes(x = as.factor(SoilBurnSe), y = (..count..)/sum(..count..), fill = as.
  geom_bar(stat = "count") +

  # color scale to match maps
  scale_fill_brewer(palette = "Spectral", direction = -1,
                    name = "Severity",
                    labels = c("Very low/unburned", "Low", "Moderate", "High")) +

  # labels and title
  ylab("Proportion") +
```
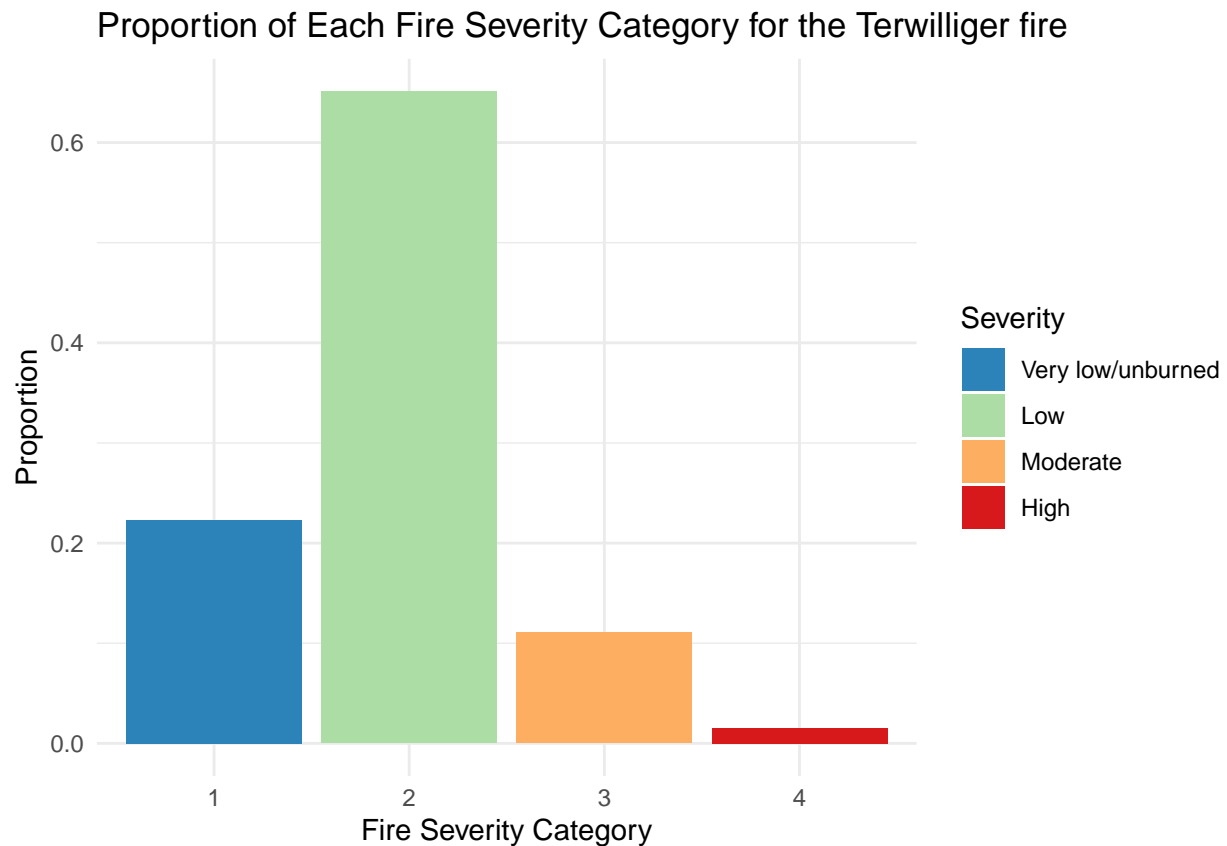
```
  xlab("Fire Severity Category") +
  ggtitle("Proportion of Each Fire Severity Category for the Terwilliger fire") +

  theme_minimal()
```

```
## Warning: The dot-dot notation ('..count..') was deprecated in ggplot2 3.4.0.
## i Please use 'after_stat(count)' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```
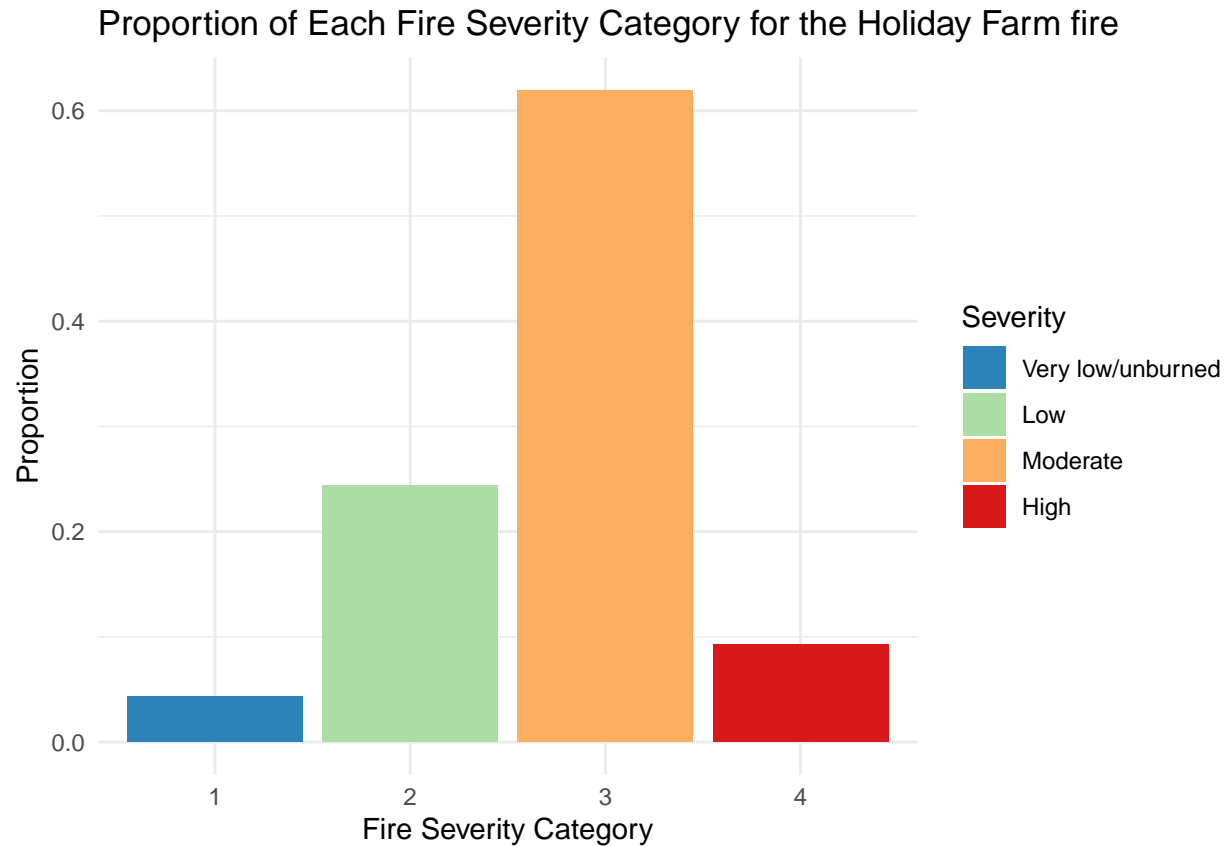

Proportion of Each Fire Severity Category for the Terwilliger fire

```
#holiday farm
ggplot(data = df_holiday, aes(x = as.factor(Layer_1), y = (..count..)/sum(..count..), fill = as.factor(
  geom_bar(stat = "count") +

  # color scale to match maps
  scale_fill_brewer(palette = "Spectral", direction = -1,
                    name = "Severity",
                    labels = c("Very low/unburned", "Low", "Moderate", "High")) +

  # labels and title
  ylab("Proportion") +
  xlab("Fire Severity Category") +
  ggtitle("Proportion of Each Fire Severity Category for the Holiday Farm fire") +
```

```
theme_minimal()
```

Proportion of Each Fire Severity Category for the Holiday Farm fire


```
#beachie creek
ggplot(data = df_beachie, aes(x = as.factor(Layer_1), y = (..count..)/sum(..count..), fill = as.factor(
  geom_bar(stat = "count") +

  # color scale to match maps
  scale_fill_brewer(palette = "Spectral", direction = -1,
                    name = "Severity",
                    labels = c("Very low/unburned", "Low", "Moderate", "High")) +

  # labels and title
  ylab("Proportion") +
  xlab("Fire Severity Category") +
  ggtitle("Proportion of Each Fire Severity Category for the Beachie Creek fire") +

  theme_minimal()
```

Proportion of Each Fire Severity Category for the Beachie Creek fire
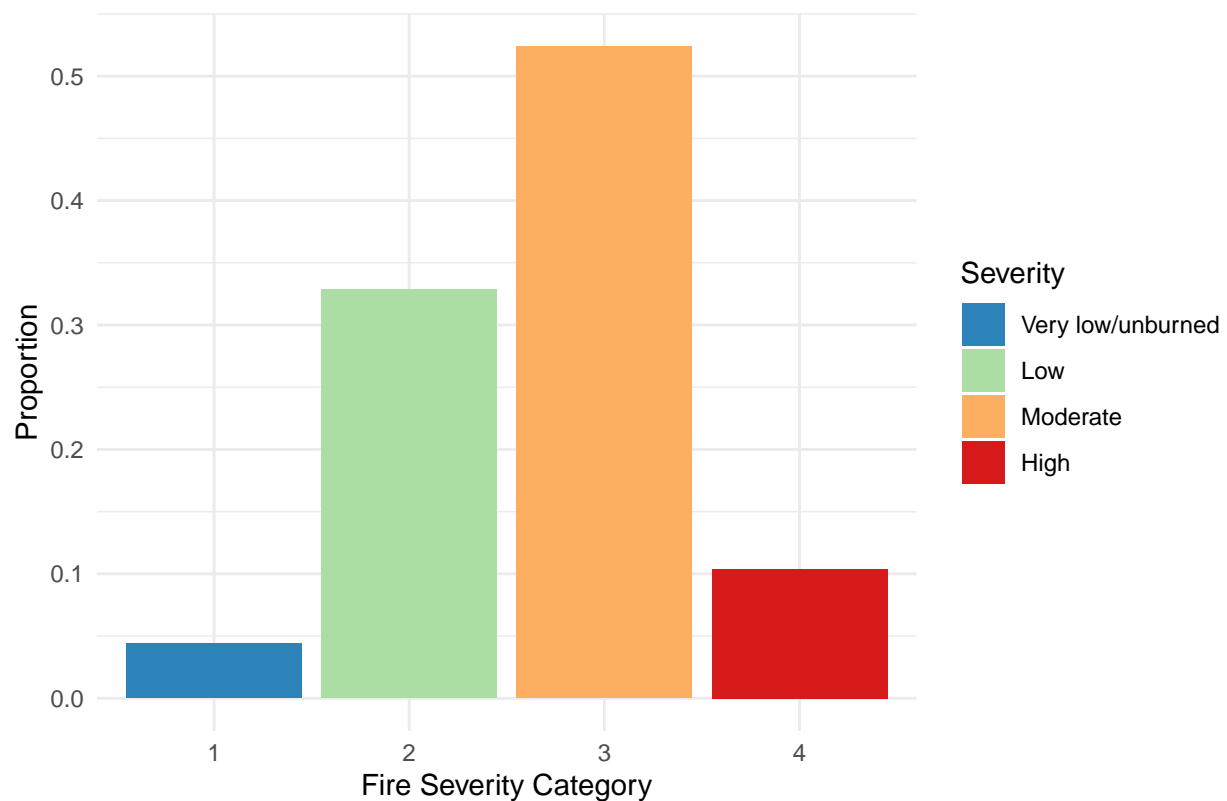
b. What do you notice about the frequency of different severity classes when you compare these barplots. How does this relate to the Haldofsky reading? ANSWER: I notice that the Terwilliger fire had low burn severity as it's highest proportion, and the holiday farm and beachie creek fires had moderate severity as the highest proportion. This reminds me of the Haldofsky reading because the author focuses on the different characteristics a fire or area can have depending on it's location, forest type, and other factors that can impact how the area experiences fire. This is an example of different areas responding to fire differently.

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/