Refine spatial plots in R.

Learning Objectives

After completing this tutorial, you will be able to:

- Add a variable to the markdown chunk in your rmd report.

What you need

You will need a computer with internet access to complete this lesson and the data for week 6 of the course.

Download Week 6 Data (~500 MB){:data-proofer-ignore=" .btn }

```
all_landsat_bands <- list.files("data/week6/Landsat/LC80340322016189-SC20170128091153/crop",
           pattern=glob2rx("*band*.tif$"),
           full.names = T) # use the dollar sign at the end to get all files that END WITH

all_landsat_bands_st <- stack(all_landsat_bands)
```

Titles using plotRGB()

You can try to add a title but it won't plot

```
plotRGB(all_landsat_bands_st,
        r=4,b=3,g=2,
        stretch="hist",
        main="title here")
```

If we add the axes=T argument, our title plots but we also get the x and y axis. We don't need those. As a work around we can set the x and y axis labels to plot using "white. Also we turn off the tick marks.

```
# adjust the parameters so the axes colors are white. Also turn off tick marks.
par(col.axis="white", col.lab="white", tck=0)
# plot
plotRGB(all_landsat_bands_st,
        r=4,b=3,g=2,
        stretch="hist",
        main="title here",
        axes=T)
```

The final step is to turn off box which leaves a line on the left hand side of the plot.

```
# adjust the parameters so the axes colors are white. Also turn off tick marks.
par(col.axis="white", col.lab="white", tck=0)
# plot
plotRGB(all_landsat_bands_st,
        r=4,b=3,g=2,
        stretch="hist",
        main="title here",
        axes=T)
# set bounding box to white as well
box(col="white") # turn all of the lines to white
```

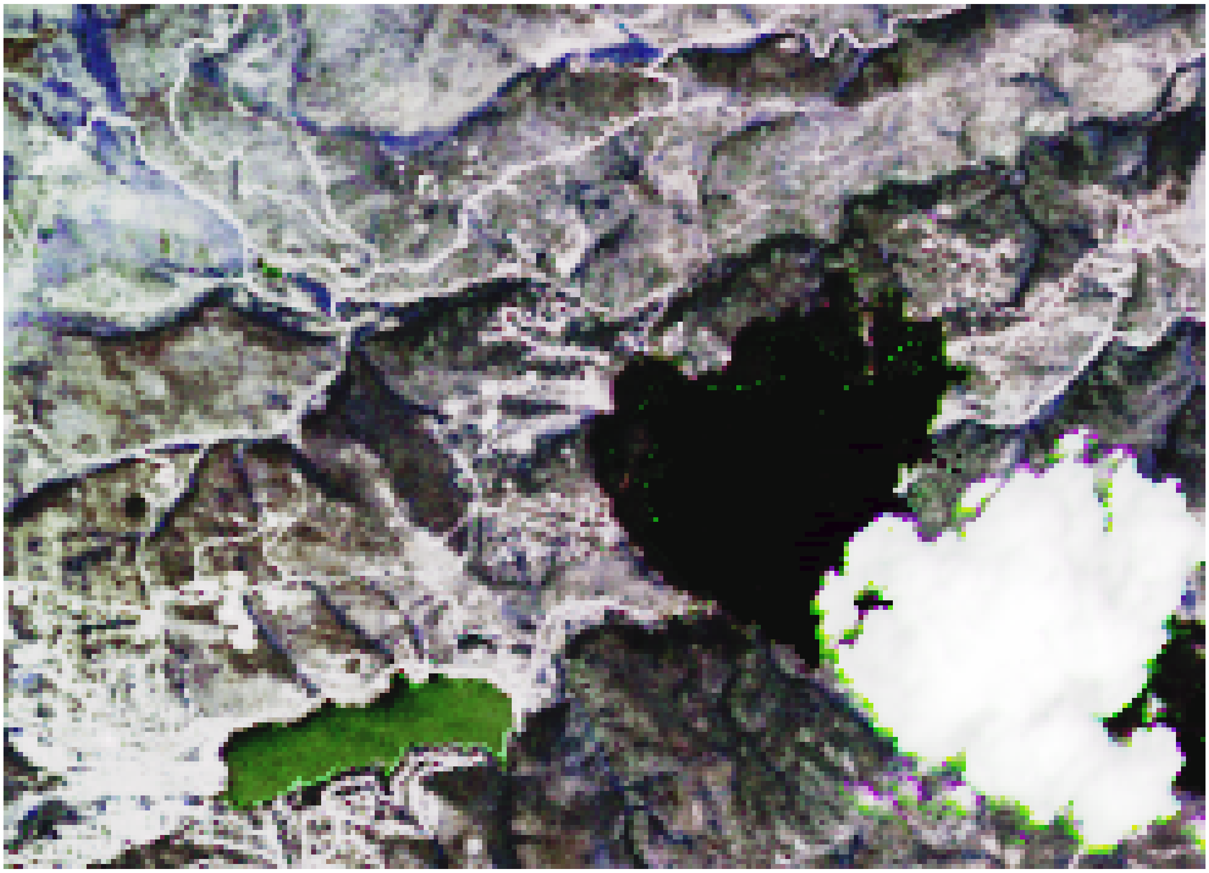

Figure 1: Remove axes labels.

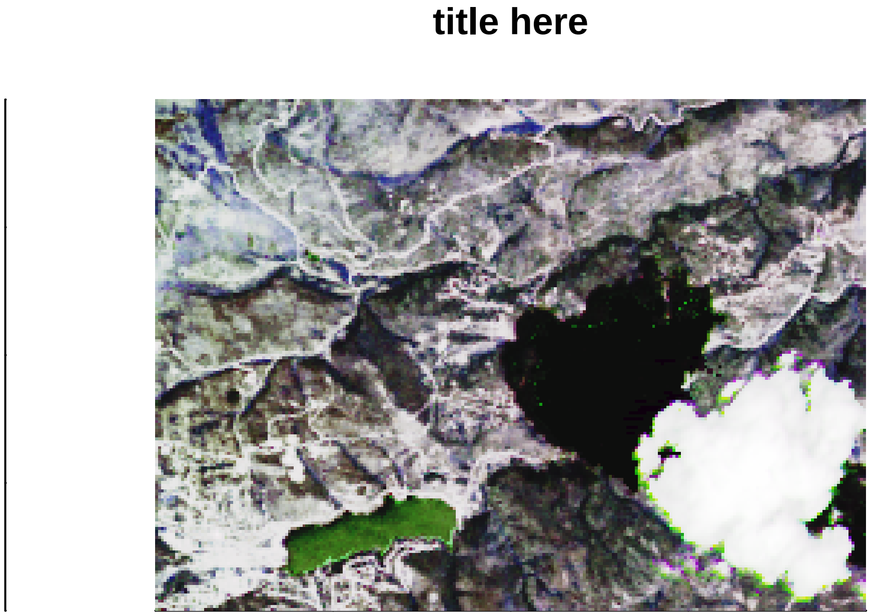

Figure 2: Remove axes labels.

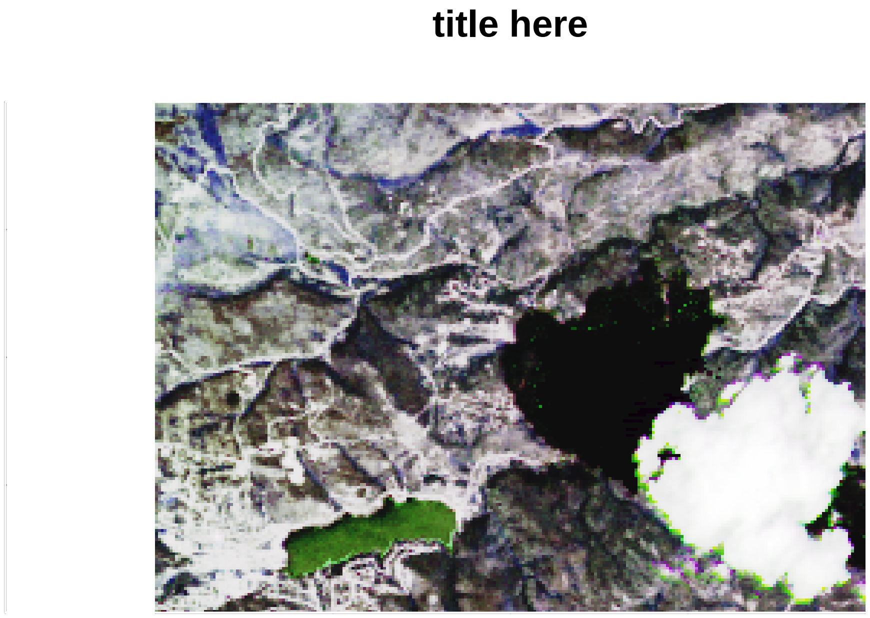

Figure 3: Remove axes labels.

Finally, if i adjust the fig.width and fig.height arguments for the plot, I can remove some of the extra white space that I see above the below the plot.

In my plot below, i used `fig.width=7, fig.height=6` in my code chunk arguments. The units are in inches.

```
# adjust the parameters so the axes colors are white. Also turn off tick marks.
par(col.axis="white", col.lab="white", tck=0)
# plot
plotRGB(all_landsat_bands_st,
        r=4,b=3,g=2,
        stretch="hist",
        main="title here",
        axes=T)
# set bounding box to white as well
box(col="white") # turn all of the lines to white
```

Any time you modify the parameters, you should consider resetting the dev() space. In the case above we set the axes to white. This setting will become a global setting until you clear your plot space! To clear the plot space programmatically, use `dev.off()`.

```
# reset dev (space where plots are rendered in RStudio)
dev.off()
## null device
##           1
```

legends

```
# calculate NDVI
ndvi <- (all_landsat_bands_st[[5]] - all_landsat_bands_st[[4]]) / (all_landsat_bands_st[[5]] + all_lands
```

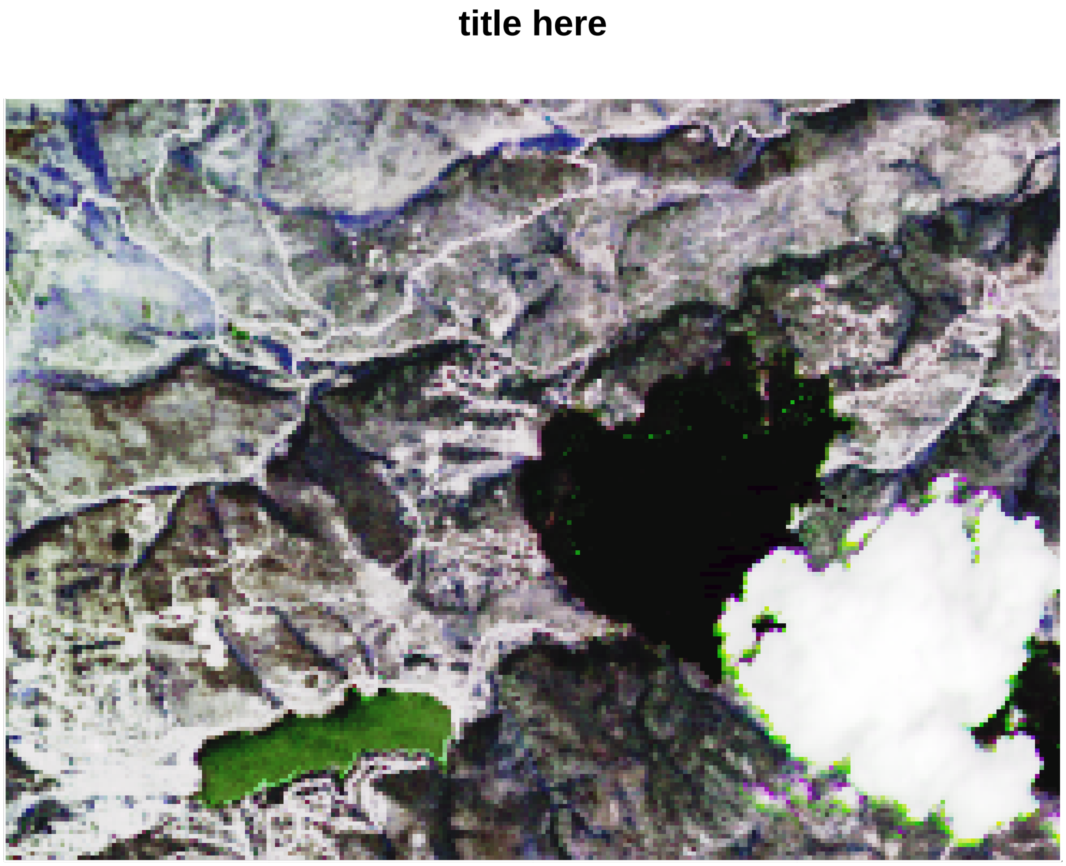
title here

Figure 4: Remove axes labels.

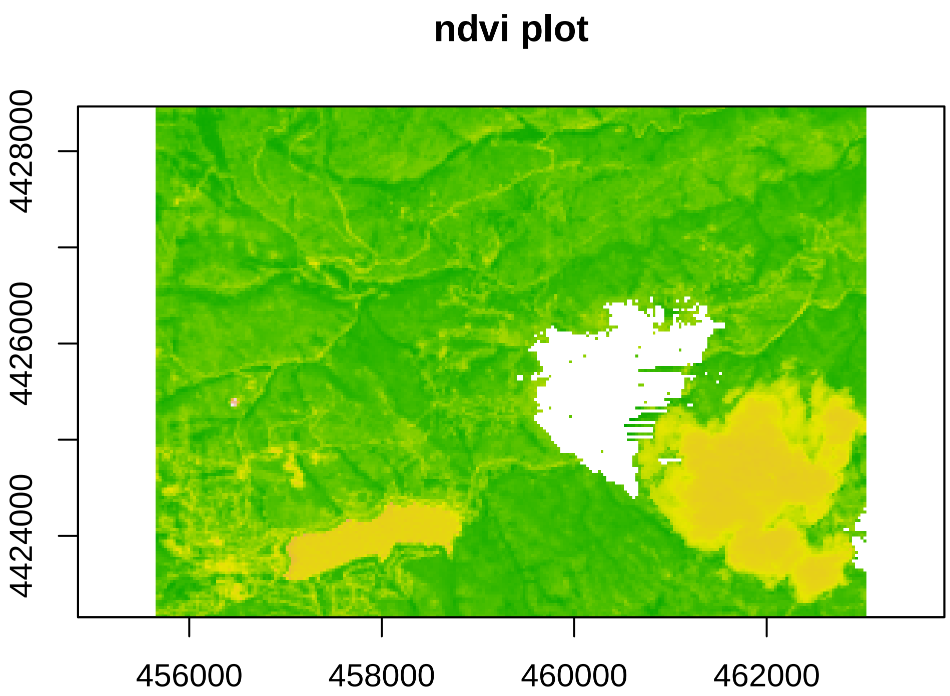

Figure 5: ndvi plot - no legend

```
# plot ndvi

plot(ndvi,
     legend=F,
     main="ndvi plot")
```

```
# reclassify ndvi
# create classification matrix
reclass <- c(-1, .3, 1,
             .3, .5, 2,
             .5, 1, 3)
# reshape the object into a matrix with columns and rows
reclass_m <- matrix(reclass,
                    ncol=3,
                    byrow=TRUE)

ndvi_classified <- reclassify(ndvi,
                         reclass_m)
the_colors <- c("grey", "yellow", "springgreen")
# plot ndvi with legend
plot(ndvi_classified,
     legend=F,
     main="ndvi plot",
     col=the_colors)
legend("topright",
       legend=c("Healthy vegetation", "Less healthy vegetation", "No vegetation"),
       fill= the_colors)
```

There are two problems with our plot:

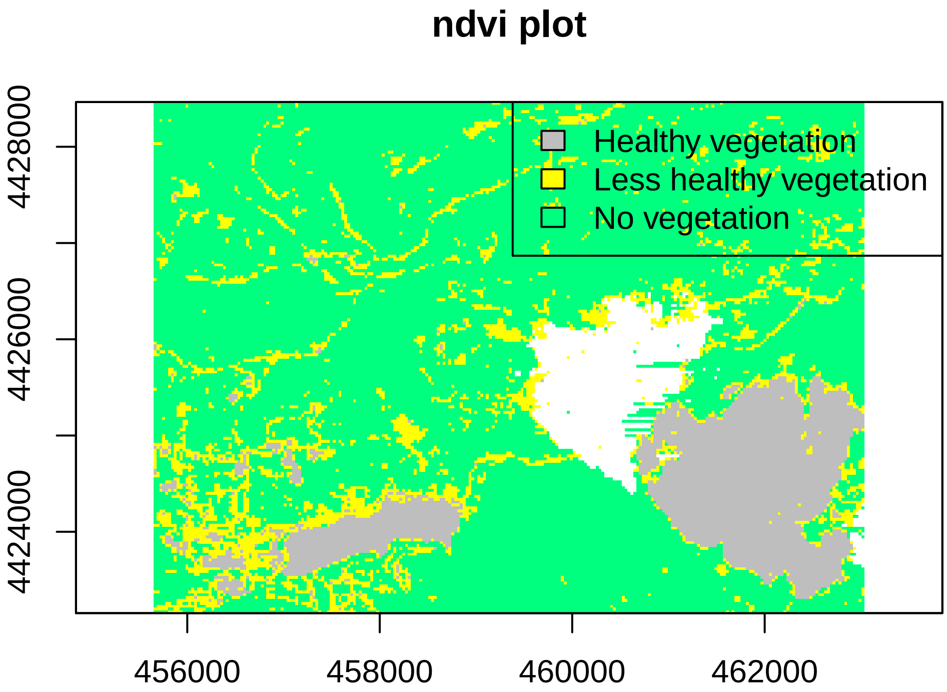

Figure 6: ndvi plot - no legend

1. *The order of our colors is wrong:* we can use `rev()` on our list of colors to reverse the order of colors drawn on the legend.
2. *We'd like the legend to be drawn outside of the plot.* We can set the x and y location of the legend using `@extent@xmax` for the x location and `@extent@ymax`. This tells R to draw the legend along the upper RIGHT hand corner of the extent of our NDVI dataset.

```
# plot ndvi with legend
plot(ndvi_classified,
     legend=F,
     main="ndvi plot",
     col=the_colors)
# set xpd to T to allow the legend to plot OUTSIDE of the plot area
par(xpd=T)
legend(x = ndvi_classified@extent@xmax, y=ndvi_classified@extent@ymax,
       legend=c("Healthy vegetation", "Less healthy vegetation", "No vegetation"),
       fill= rev(the_colors)) # use rev to reverse the order of colors for the legend
```

On to the pesky white space on either side of the plot. There are several ways to handle this - one is to turn off the axes which we don't need anyway. We can also turn off the BOX that bounds the plot.

```
# plot ndvi with legend
plot(ndvi_classified,
     legend=F,
     main="ndvi plot with axes & box turned off",
     col=the_colors,
     axes=F,
     box=F)
# set xpd to T to allow the legend to plot OUTSIDE of the plot area
par(xpd=T)
```

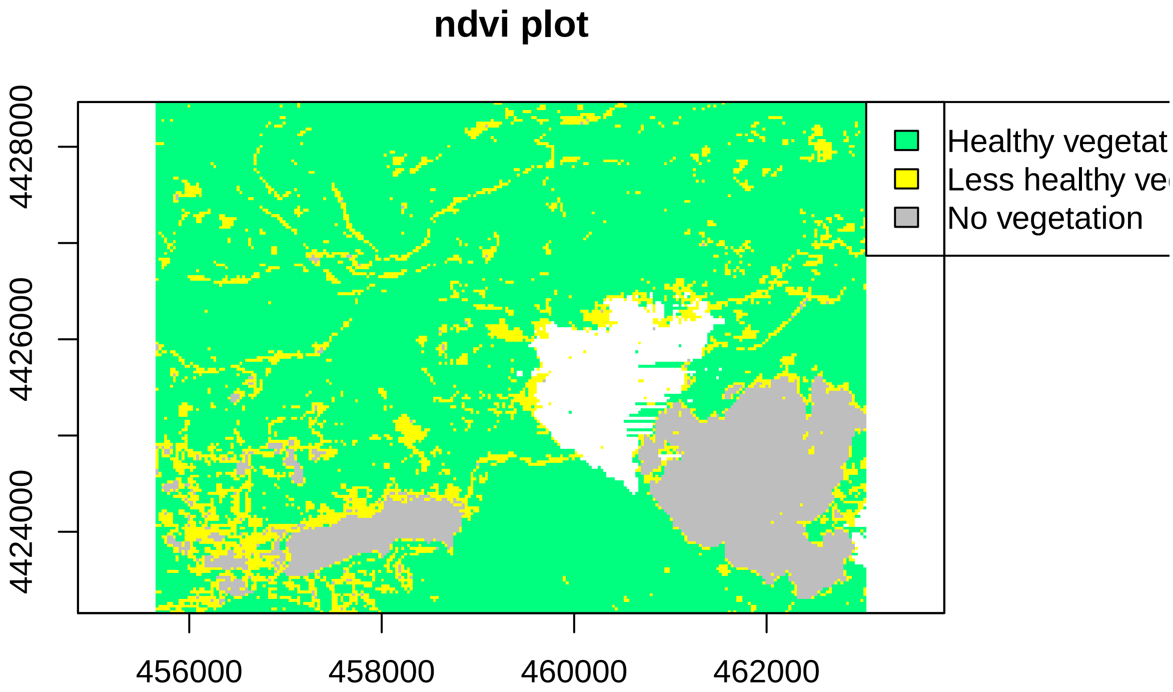

Figure 7: plot with legend in the upper right.

```
legend(x = ndvi_classified@extent@xmax, y=ndvi_classified@extent@ymax,
       legend=c("Healthy vegetation", "Less healthy vegetation", "No vegetation"),
       fill= rev(the_colors)) # use rev to reverse the order of colors for the legend
```

Finally - let's work on figure location. Notice in the plot above that there is too much white space above and below the plot. In the case of my plot, the aspect ratio of width:height is not right. We want the height to be SMALLER to remove some of the white space. The white space is there because R is trying to plot our figure using a 1:1 aspect ration. We can use the `mar` argument to adjust the white space on the bottom, left, top, or right sides of the plot. This makes room for our legend.

Below, we adjusted the top margin to 2 and the right margin to 5. This makes space for our legend but also makes a bit of space for our plot title.

```
# set a margin for our figure
par(xpd=F, mar=c(0,0,2,5))
# plot ndvi with legend
plot(ndvi_classified,
     legend=F,
     main="ndvi plot with axes & box turned off",
     col=the_colors,
     axes=F,
     box=F)
# set xpd to T to allow the legend to plot OUTSIDE of the plot area
par(xpd=T)
legend(x = ndvi_classified@extent@xmax, y=ndvi_classified@extent@ymax,
       legend=c("Healthy vegetation", "Less healthy vegetation", "No vegetation"),
       bty=F, # turn off legend border
       fill= rev(the_colors)) # use rev to reverse the order of colors for the legend
```

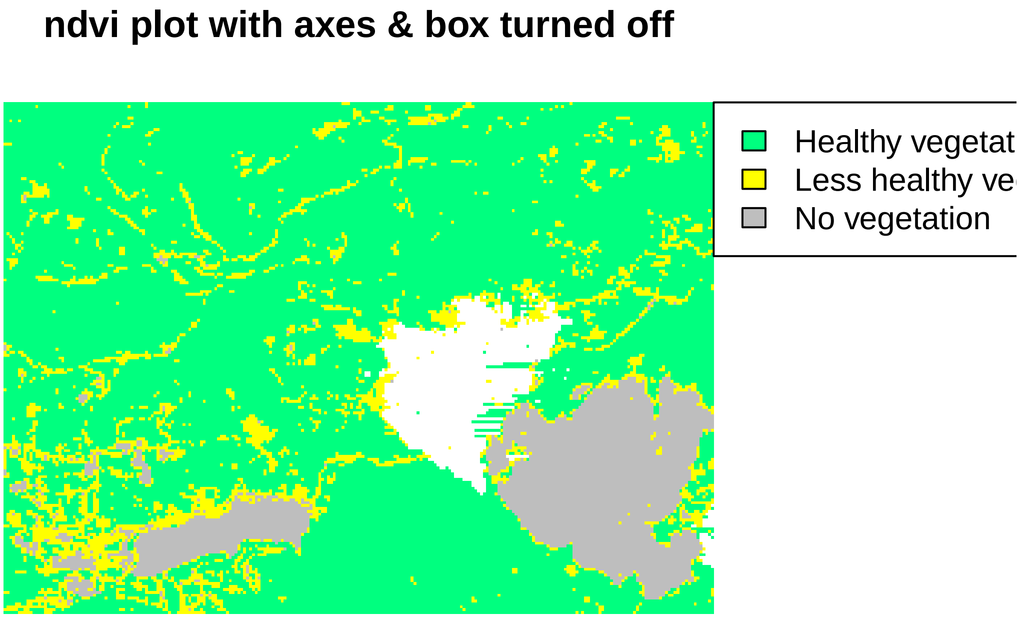

Figure 8: plot with legend in the upper right.

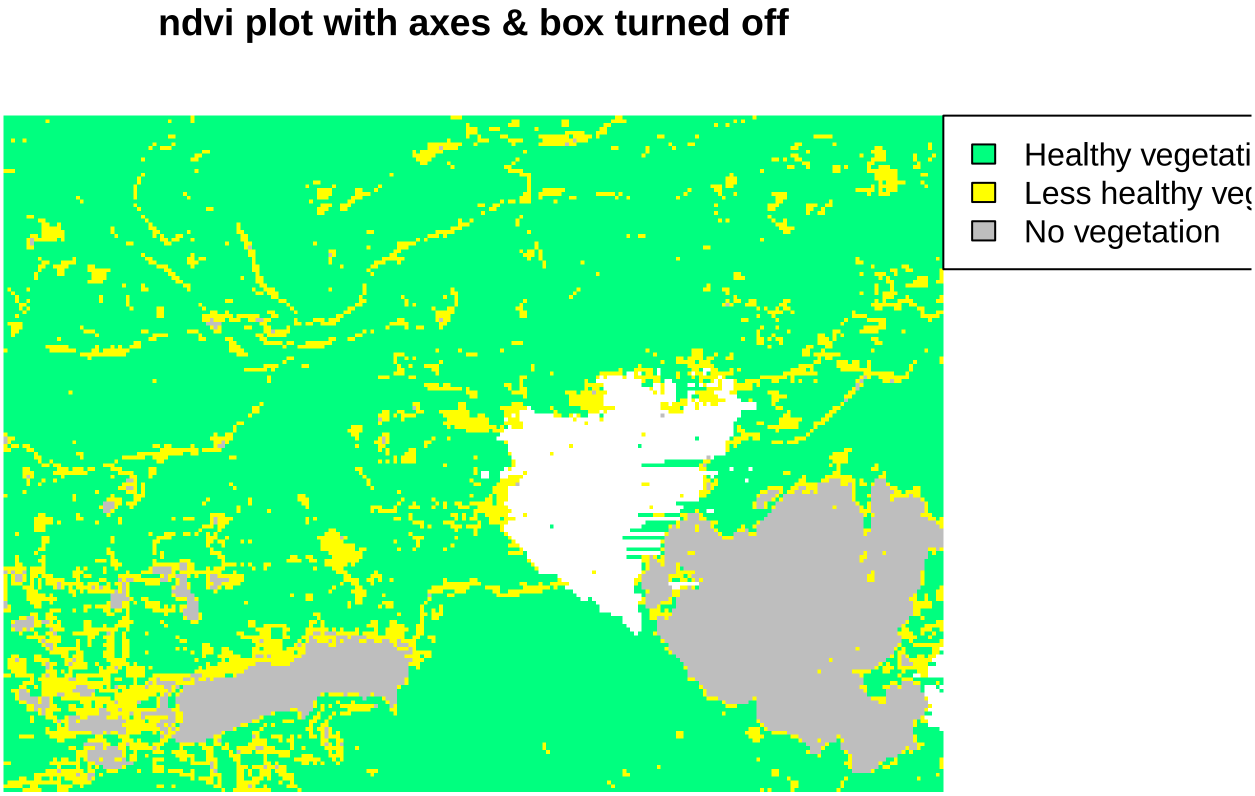

Figure 9: plot with legend in the upper right.

```
dev.off()
## null device
##           1
```

If things are still not looking right, we can adjust the size of our output figure. Try adding the arguments fig.width and fig.height to your code chunk. `{r fix-plot-legend3, fig.cap="plot with legend in the upper right.", fig.width=6, fig.height=4}`

I use these values because i want the HEIGHT of the figure to be smaller. I want the WIDTH of the figure to be “landscape” or wider so i’ll leave that value alone.

Below, I set `fig.width=6`, `fig.height=4` in my code chunk. setting a smaller height pulled the title down closer to my plot. I’ve also decreased the font size of my legend using the `cex` legend argument. Note that you may have to play with the figure size a bit to get it just right.

HINT: use ‘dev.size()“ to figure out the size of your plot dev space in RStudio.

```
# set a margin for our figure
par(xpd=F, mar=c(0,0,2,6))
# plot ndvi with legend
plot(ndvi_classified,
     legend=F,
     main="NDVI plot with axes & box turned off & custom margins\n to make room for the legend",
     col=the_colors,
     axes=F,
     box=F)
# set xpd to T to allow the legend to plot OUTSIDE of the plot area
par(xpd=T)
legend(x = ndvi_classified@extent@xmax, y=ndvi_classified@extent@ymax,
       legend=c("Healthy vegetation", "Less healthy vegetation", "No vegetation"),
       bty="n", # turn off legend border
       cex=.8, # make the legend font a bit smaller
       fill= rev(the_colors)) # use rev to reverse the order of colors for the legend
```

While we are at it, let’s add a border using a crop extent shapefile that was used to clip these data

```
# import crop extent
crop_ext <- readOGR("data/week6/vector_layers/fire_crop_box_2000m.shp")
# set a margin for our figure
par(xpd=F, mar=c(0,0,2,6))
# plot ndvi with legend
plot(ndvi_classified,
     legend=F,
     main="NDVI plot with axes & box turned off & custom margins\n to make room for the legend",
     col=the_colors,
     axes=F,
     box=F)

plot(crop_ext,
     lwd=2,
     add=T)
# set xpd to T to allow the legend to plot OUTSIDE of the plot area
par(xpd=T)
legend(x = ndvi_classified@extent@xmax, y=ndvi_classified@extent@ymax,
       legend=c("Healthy vegetation", "Less healthy vegetation", "No vegetation"),
       bty="n", # turn off legend border
       cex=.8, # make the legend font a bit smaller
```

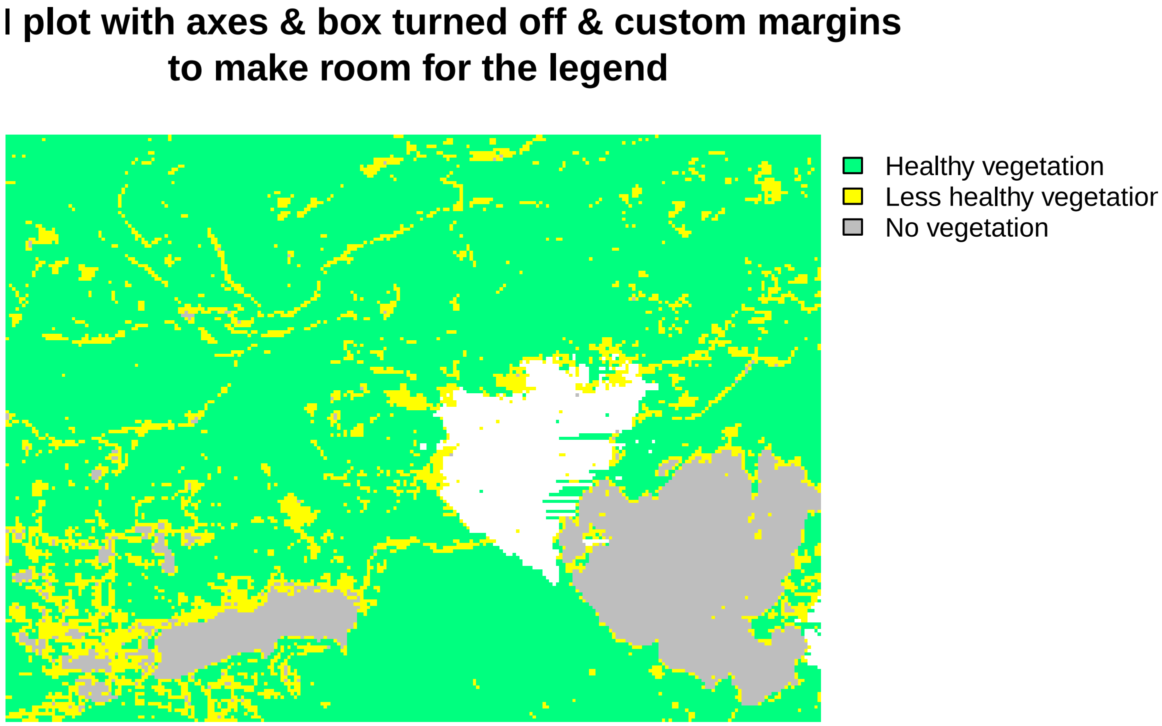
l plot with axes & box turned off & custom margins
to make room for the legend
🟩 Healthy vegetation
🟨 Less healthy vegetatior
⬜ No vegetation

Figure 10: plot with legend in the upper right.

```
       fill= rev(the_colors)) # use rev to reverse the order of colors for the legend
```

Again - always reset dev when you've been adjusting `par()` values for plots!

```
dev.off()
```

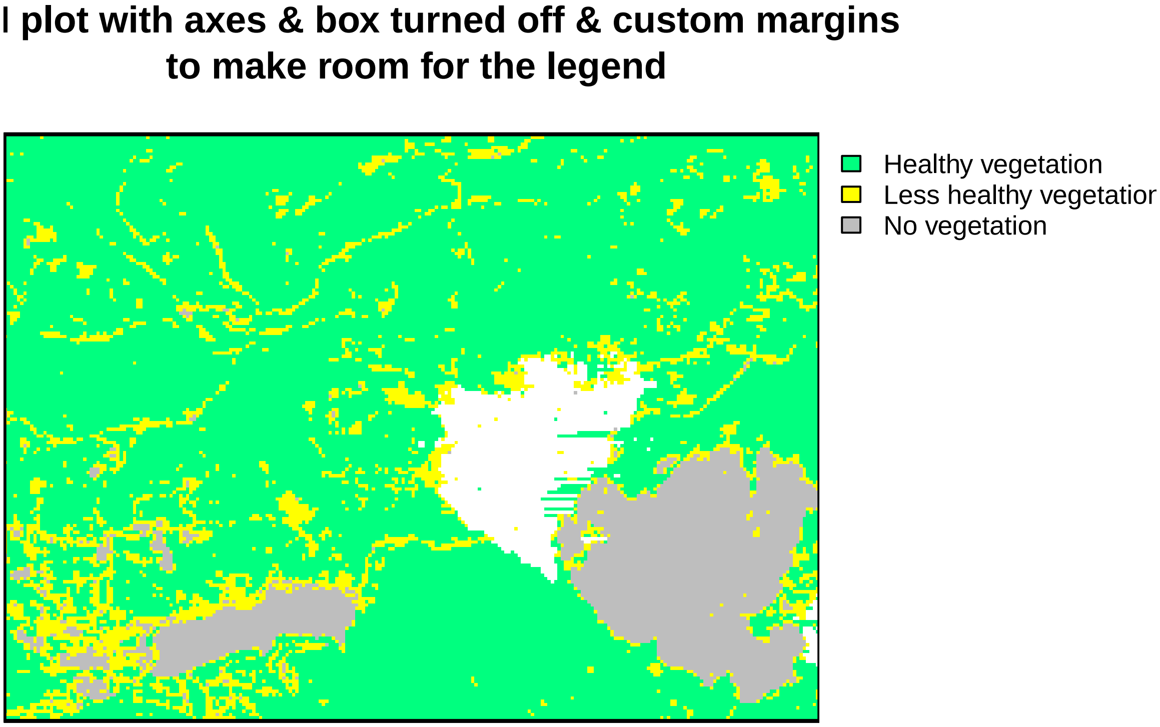
l plot with axes & box turned off & custom margins
to make room for the legend
Healthy vegetation
Less healthy vegetatior
No vegetation

Figure 11: add crop box