GGPlot Class Reflections

**Lesson 10- Multipanel Plots**

**What was your level of comfort with the lesson/application?**

I was fairly comfortable with this lesson as it allowed us to revisit code (i.e. how to make a histogram, how to annotate a plot, how to add a line, pick a fill color). I learned new code to perform on vector of indexes (setdiff, union, intersect) that was easy to conceptualize. The overview of grid.arrange was also easy to follow, except the part about resizing plots.

**What areas of the lesson/application confused or still confuses you?**

Resizing plots: what do the numbers mean?? 1,1,2? Then 1,1,NA. Then 3,NA,NA.

Which stand for the plot # and which stand for size?

*grid.arrange(snowy, sfoggy, breezy,*

*layout\_matrix = rbind(*

*c(1,1,2),*

*c(1,1,NA),*

*c(3,NA,NA)));*

Size really refers how many grid spaces the plot takes. In the above example there is a 3x3 grid (so, 9 grid spaces). Plot 1 takes up four grid spaces on the 3x3 grid. Plots 2 and 3 each take 1 grid space, and three of the grid spaces have nothing in them (NA).

Two caveats:

* A plot is put on a grid only once
* A plot will always fill a rectangular spot on a grid

So this would produce the exact same results -- plot 1 fills the rectangle between (1,1) and (2,2).

*grid.arrange(snowy, sfoggy, breezy,*

*layout\_matrix = rbind(*

*c(1,NA,2),*

*c(NA,1,NA),*

*c(3,NA,NA)));*

This would also produce the same results:

*grid.arrange(snowy, sfoggy, breezy,*

*layout\_matrix = rbind(*

*c(NA,1,2),*

*c(1,NA,NA),*

*c(3,NA,NA)));*

Plot 1 here is 2 rows by columns (6 grid spaces) and will overlap with plot 2

*grid.arrange(snowy, sfoggy, breezy,*

*layout\_matrix = rbind(*

*c(1,NA,2),*

*c(NA,NA,1),*

*c(3,NA,NA)));*

difference between arrangeGrob and grid.arrange

I have never used **arrangeGrob** but it seems that it does the same thing, but the number of rows and columns is defined using **nrow** and **ncol** – and you have a few more controls like padding between the plots. However, I wonder what happens if your **layout\_matrix** is not consistent with **nrow** and **ncol**?

**What is a way you can apply the material in this lesson towards your research or area of study?**

The operations on indexes (intersect, setdiff, etc) is very useful, as I am often subsetting data for different conditions and I get tired of using mydata[(mydata$varB==”app1”) & (mydata$varB==”noapp”),] or some variation of this.

The grid.arrange is very useful because I usually do plot more than 1 plot together for my manuscripts (and yes, sometimes I do want some of the plots to be bigger than the other and I had not been able to figure out how to do that).

**What are some things you would like to learn related to, but not covered in, this lesson?**

save the legend key of a plot and then display it as its own plot

for example, if you had 5 plots, the 6th spot would be the legend key (large, and easy to read!)

A tricky problem because the legend is based on the data from the current plot – and each of the 6 areas holds a separate plot. I guess if all your plots have the same legend then you could create a plot where everything is invisible except for the legend. Or, you could use ***annotate()*** to manually create a legend.

**Lesson 9- Textplots**

**What was your level of comfort with the lesson/application?**

This lesson was a bit challenging because there were a lot of things going on in ggplot (the text, the color of the text, the actual text that displayed, the legend, the legend breaks, the legend values). I understood everything from the lesson, except everything under scale\_color\_gradientn.

**What areas of the lesson/application confused or still confuses you?**

Everything under scale color gradientn (more examples would help!)

Particularly, colors, values, breaks, and labels

It is hard to wrap my head around all of these and what they mean depending on what goes under *geom\_text*

preciphum = ggplot(data=weatherData) +

geom\_text(mapping=aes(x=relHum, y=precipNum,

color=avgTemp,

label=avgTemp),

size=4) + # size is NOT part of the mapping

scale\_y\_continuous(trans="log10")+

scale\_color\_gradientn(colors=c("red","darkgreen","blue")) +

***label=avgTemp*** means that every **precipNum** vs. **relHum** point on scatterplot will be replaced by the **avgTemp** value for that day (or, more generally, the **avgTemp** value on the same row)

***color=avgTemp*** means that every **precipNum** vs. **relHum** point on scatterplot will be colored based on a gradient that is determined by the ***avgTemp*** value for that day (or, more generally, the ***avgTemp*** on the same row). A legend will be added that displays the gradients colors for all ***avgTemp*** values.

The gradient color can be modified in ***scale\_color\_gradientn()*** -- otherwise, GGPlot picks colors.

It only really makes sense to set ***color*** and ***label*** to the same value.

In geom\_text, do we need to set color to =1:nrow(weatherData)? If we instead make gradient of 25 colors, do we need to change scale\_color\_gradientn too?

If you wanted a gradient that goes between 25 colors, then you need to put 25 colors in the colors subcomponent of ***scale\_color\_gradientsn***.

If you wanted exactly 25 colors – then you cannot use gradients.

**What is a way you can apply the material in this lesson towards your research or area of study?**

I will definitely use the code for modifying the legend title, position, height, and width!

In my research, I also regularly use the color parameter, but under geom\_point.

I will use geom\_text, but more in the context of annotating the plot.

**What are some things you would like to learn related to, but not covered in, this lesson?**

In the last application, I wanted to fix the breaks and values on the legend (which displayed average temperature) to make them more meaningful but got stuck.

I cannot see what you were trying to do – if you add some code or give me something more specific, I can help you out.

**Lesson 8- Barplots**

**What was your level of comfort with the lesson/application?**

I am starting to understand the basics of for loops, scale x discrete, geom\_hline, and annotate. I was able to complete the lesson without issue.

**What areas of the lesson/application confused or still confuses you?**

geom\_col vs. geom\_bar (I’ve only used geom bar in the past and glad I know I can use geom\_col now too!)

I am still philosophically confused by what ***geom\_bar()*** is supposed to do! I have new examples of both in ***application\_answers/App08-Alt…R*** and ***examples/geom\_col vs geom\_bar.R***. I will be going over the App08 Alt answer in class Monday.

color. I get confused where to specify the color of something. Sometimes it’s under geom\_bar/boxplot/etc. Sometimes its in scale\_fill\_discrete. Sometimes its under fill=. This is the type of code I usually have to look up or look back at previous applications.

There are three scenarios here:

1. You just want to set a color that is not related to the data (turns the whole bar red regardless of data)

geom\_col(mapping=aes(x=month, y=heatDays),

fill = "red"

width=0.6) + …

1. You want to color code the bars based on some data (uses the factors in wtype to set parts of the bar to different colors – GGPlot picks the colors)

geom\_col(mapping=aes(x=month, y=heatDays, fill=wtype),

width=0.6) + …

1. You want to ***manually*** color code the bars based on some data (uses the factors in wtype to set parts of the bar to different colors – scale\_fill\_manual() adjusts the colors)

geom\_col(mapping=aes(x=month, y=heatDays, fill=wtype),

width=0.6) +

scale\_fill\_manual(values=c("red", "blue"…)…

Yeah, it's a bit confusing…

I had to look at the app answers to plot two bars in one plot (cooling days AND heating days). The method that I would have come up with would have been convoluted and involved reorganizing the entire data frame (dplyr: melt). I did not know it was as simple as adding TWO geom\_col parameters.

Well, it was not that simple. The answer is actually wrong but it did not properly stack the bars. There is a bug in GGPlot that does not allow one to both stack and nudge bars. My app08Alt.R answer has the fix and I will be going over it in class.

**What is a way you can apply the material in this lesson towards your research or area of study?**

Yes, I frequently use barplots in my research that have to be filled with another variable’s values. I also need to add lines or annotate the plot, which I did not know how to do before.

**What are some things you would like to learn related to, but not covered in, this lesson?**

How to change the color of the outline of the entire bar (not each individual piece of the bar). For example, in the last application, each bar was blue or red. I wanted to add a black outline to each bar (one bar for each month).

This is not possible (as far as I know) unless you supply GGplot with the summed values for the months instead of asking GGPlot to stack the values. I will be doing this example in class on Monday.

In the past application, we used fill to color code the bars by weather type. The first weather type was “NA” but in the plot it showed as a blank title. I know I can use the labeller=as\_labeller(mylegendlabels) to add the “NA” to the legend but I do not know where this code would go.

Easiest way: add it to your column (then you do not need to mess with GGPlot)

weatherData$wtype[which(weatherData$wtype=="")] = "--";

Or, harder, put it in you scale\_fill\_manual

scale\_fill\_manual(labels=c("--", "", "", "", "", "", "", ""),

values=c('#b3e2cd','#fdcdac','#cbd5e8','#f4cae4','#e6f5c9',

'#fff2ae','#f1e2cc','#cccccc'))+

But, for that, you need to manually put all 8 values in (I, lazily, did not do this) – I do not know if there is a way to replace only one value in the legend.

In this case, ***scale\_fill\_manual()*** modifies the styles of your ***fill*** in the ***mapping***. And, ***fill=wtype***, means the fill colors of the bars are mapped to the ***wtype*** column. That's a lot to get one's head around!

**Lesson 7-Boxplots part II**

**What was your level of comfort with the lesson/application?**

I was very comfortable with this lesson and application as it mostly covered material that we had seen before (how to calculate quantiles, convert values to categories/levels, make boxplots). I appreciated this because it gave me the opportunity to reinforce what I had learned and highlighted the areas I had to go back and read. There was some new material (facet grid, using factor (levels=) in ggplot, the labeller function, and scale\_x\_discrete(limits=) which was great to learn because it is handy.

**What areas of the lesson/application confused or still confuses you?**

Why didn’t my for loop (see my code: method 1) and my non for loop (see my code: method 2) not give the same values when categorizing relative humidity into low, medium, and high?

Copy/paste error – check what your if else statement is saying (hint: it is not comparing humidity!)

RGB colors (red=1, green=0.2, blue=0)—it is hard to wrap my head around the exact quantities to get the color I want. I would much rather just specify the actual colors (i.e. “aquamarine” or the HEX code “#8dd3c7”).

I am agnostic about which method you use. The hardest part about using rgb() is that you are mixing light colors – not pigment colors. So blue and yellow does not make green. HEX codes are using the same mixing-light method – just harder because the colors are represented using hexadecimal values instead of numbers between 0 and 1.

HEX code “#8dd3c7" means:

* red: 8d
* green: d3
* blue: c7

Can you use labeller(as\_labeller) under geom\_boxplot as well, in addition to under facet\_grid?

What labels do you specifically want to change in the geom\_boxplot? There is probably an easier way…

**What is a way you can apply the material in this lesson towards your research or area of study?**

I am constantly battling ggplot because it treats variables that I want as factors not as factors, and because it does not order them and label them the way I want when plotting. Material from this application has solved all of those issues for me! Some plots require very specific coloring of boxplots and knowing that I can supply the vector of colors (in order that the boxplots appear) is helpful. For example, coloring one boxplot white and all of the rest green.

**What are some things you would like to learn related to, but not covered in, this lesson?**

Other ways boxplots could look

How to change the shading of the facet grid titles (default is grey shading)

add to theme() -- you can use a color name instead of "transparent"

theme(strip.text = element\_text(size=9, color="darkgreen"), # this is just text changes

strip.background = element\_rect(colour="gray", fill="transparent"))

**Lesson 6-Boxplots I**

**What was your level of comfort with the lesson/application?**

In the prior class reflection, I asked for more for loops, and I got that in this lesson! I was the most comfortable with the plotting (boxplots, violin plots) and plotting components (adding whiskers, annotating). I was also comfortable with subsetting and extracting values from a data frame. I am getting slightly more experienced with for loops, but they still intimidate me a little bit. I think I just need to do lots and lots of for loops.

**What areas of the lesson/application confused or still confuses you?**

When else will we use stat\_boxplot

I have not been able to come up with another use. Adding ***identity = "stat"*** in ***geom\_boxplot()*** essentially invokes the functionality of ***stat\_boxplot()***. This is in the last example I went over in class Monday and will continue with next Monday.

Are annotate (geom=”text”) and geom\_text(aes()) equivalent? Reason for using one over the other?

annotate() is designed for manually adding text to a plot that is not directly linked to data (e.g., descriptive text) whereas geom\_text() is designed for mapping data on to a plot using text instead of points. These can obviously overlap.

In lesson 6, you had code to calculate the median values for North and South directions. In my own research, I conduct similar operations but the code is more condensed. For example, for the median value:

A=median(weatherData$changeMaxTemp[weatherData$windDir=="North"])

Is the code I have written I have bad practice? Should I be aiming to structure my code like yours:

northVals=which(weatherData$windDir == "North");

weatherData[northVals, "changeMaxTemp"]

northMed = median(weatherData[northVals,"changeMaxTemp"], na.rm=TRUE);

I would argue that you should use the expanded code. It contains more information, and it is easier to understand and far easier to debug.

the extension: condensed for loop (many for loops in one). I followed the code and understand superficially.

If you have three different for loops that all have use the same variable and iteration:

for(day in 1:nrow(weatherData))

then you can combine the for loops. A subtle, but not vital, point.

**What is a way you can apply the material in this lesson towards your research or area of study?**

Similar as last week, the for loops, data indexing, and data binning/categorizing was useful and I am going to use this code frequently for my own research. I do plot lots of boxplots for my manuscripts, and this week, I learned how to modify the appearance of outlier values and how to annotate the plot, which I have wanted to learn how to do.

**What are some things you would like to learn related to, but not covered in, this lesson?**

for loops to conduct the same mathematical operation over multiple columns (e.g. converting 3 columns of data from F to C)

I would do it as a function. I added an example of this in the **scripts\_extension** folder of the Class Material in GitHub. The file is called **ConvertMultipleColumns.R**

This is where some would use a 2-D for loop.

how to modify the appearance of certain outlier values, not all outlier values in a plot

You would have to remove all outliers from the plot (outlier.shape = NA) and then manually add the outlier point. We will be talking next week how to get the outlier points and plot them manually next class. The next challenge for you would be to selectively pick out outlier point. I can help with that when it comes time.

**Lesson 5-Histograms**

**What was your level of comfort with the lesson/application?**

Of all of the lessons, I was the least comfortable with this one, but the content was still familiar enough that I knew I would be able to complete the lesson and application. I had worked with stacked plots before, and knew about moving around legends but I had not worked with for loops or the which function before, so that was new.

**What areas of the lesson/application confused or still confuses you?**

When to use & vs. &&

I added a script file to your ***scripts*** folder that talks about this

facet\_grid vs. facet\_wrap

Will need to look into this… (still have not looked into it….)

for loops: I understand them at their most basic, but need more practice constructing for loops in different ways to feel comfortable using them for my own research. Perhaps more examples and more applications of the loops.

We will get to an example in class

**What is a way you can apply the material in this lesson towards your research or area of study?**

The for loops will be very handy because often, I have meta data on my biological samples that I need to categorize/bin in different ways. Most of the tasks I do in R involve data indexing and sub-setting, so having learned the which function will also be useful. I do not really use histograms for my research but anticipate using stacked histograms to visualize microbial community data.

**What are some things you would like to learn related to, but not covered in, this lesson?**

How to add text to the inside of the plot (e.g. within the plot borders)

annotate(geom="text") – more here: <https://ggplot2.tidyverse.org/reference/annotate.html>

for loops to conduct the same mathematical operation over multiple columns (e.g. converting 3 columns of data from F to C)

I added a script file to your ***scripts*** folder that talks about this

**Lesson 4-Data objects and canvas styles**

**What was your level of comfort with the lesson/application?**

I was comfortable with both the lesson and application, although there were definitely things I had to read multiple times to make sure I understood, particularly the different ways to format Dates in R and what the different plot/panel parameters mean in ggplot.

**What areas of the lesson/application confused or still confuses you?**

What is a *tibble* exactly and when are they more appropriate to use over data frames

This is a philosophical discussion! They supposedly are modern versions of the data frame. I have not used them and people whom I work with have not found any convincing reason to switch to a tibble – unless you are already fully in the tidyverse.

**What is a way you can apply the material in this lesson towards your research or area of study?**

This is very useful for my research because I have to produce many visually appealing, colorful, publication-quality figures. Usually that means I have to make the axes and axis tick marks large, the plot border thick, and the plot elements colorful. My research deals with formatting Dates as well, and in the past, this has caused me many headaches but I believe I finally understand. I do a lot of data wrangling so knowing how to access different data frame values in different ways was helpful. I did not know that theDate = weatherData[["date"]] was the same as weatherData$theDate.

**What are some things you would like to learn related to, but not covered in, this lesson?**

How to add text to the inside of the plot (e.g. within the plot borders)

for loops to conduct a mathematical operation over multiple columns (e.g. converting 3 columns from F to C)

**Lesson 3-Plot Styles**

**What was your level of comfort with the lesson/application?**

I felt slightly more overwhelmed during this lesson compared to the previous, but it was very fun. The ggplot code is getting more busy, but the different components are making more sense now! I was able to follow along and complete the lesson and the application just fine.

**What areas of the lesson/application confused or still confuses you?**

In axis.title.x=element\_text(size=14, color="black",face="bold"); are there other parts instead of element\_text? Couldn’t ggplot have been designed so that we wouldn’t need to add element\_text to every single axis title/label component?

**What is a way you can apply the material in this lesson towards your research or area of study?**

Yes, for all of the plots I make, I have to set the shape, colors, and sizes of all of the text in the plot. I mostly always set the color of the x and y axis to black, but I do bold them and make the font large. I also have done scatter plots with regression lines but did not know how to change the color or shading of line. Now, I know and will use that code. Something important I learned is to set theme\_bw() before modifying the theme later; I have not done this in the past and it has caused much frustration.

**What are some things you would like to learn related to, but not covered in, this lesson?**

Which other ggplot components need to be in a certain order? For example, theme\_bw() needs to be set before theme(). Are there other ggplot components that need to be set in a certain order?

Center Plot title

Adding confidence intervals or standard errors to the regression line

Set dashed lines to denote upper and lower limits on y-axis

**Lesson 2-Components**

**What was your level of comfort with the lesson/application?**

I was very comfortable with the commands and skills covered in this lesson, from reading/saving as .CSV file, to plotting a scatterplot in ggplot. I knew how to set custom x and y tick marks and how to specify a plot title. I am sure as the course advances, plot components will get more challenging.

**What areas of the lesson/application confused or still confuses you?**

Not many in this lesson

**What is a way you can apply the material in this lesson towards your research or area of study?**

Absolutely. For my data, I create a lot of boxplots and a few scatterplots to showcase, for example, microbial community Shannon diversity vs. individual’s age, and using the code we created will come in handy. The code we created is more clean, structured, and easier to modify/navigate, which will save me many headaches down the road. Having the code written this way will also help me diagnose and solve R errors more quickly.

**What are some things you would like to learn related to, but not covered in, this lesson?**

What are the other components of mapping, besides aes (aesthetics)

Why certain mapping aesthetics go under geom\_point vs in ggplot

plotData = **ggplot**( data=deaths ) +

**geom\_point**(mapping=aes(x=time, y=accdeaths))

How to access a data file (.csv or .txt) directly from a URL