Making heat maps with ggplot2

Augustus Pendleton

When would we use a heatmap?

Heat maps are a common way to visualize the "intensity" of a continuous variable between two categorical variables. This can help us see patterns in between groups, and potentially identify groups within our categorical variables that are similar to each other.

Load packages

```r
library(tidyverse)
```

Read in the data

First, let's read in the data.

```r
phylum_counts <- read_csv("data/phylum_counts.csv")

head(phylum_counts)
```

```
# A tibble: 6 x 5
  Phylum           Deep_May Deep_September Shallow_May Shallow_September
  <chr>               <dbl>          <dbl>       <dbl>             <dbl>
1 Acidobacteriota      9840           8156       7682.              6837
2 Actinobacteriota   336139         385255     834380.           1476968
3 Armatimonadota          0           3243          0              20961
```

```
4 Bacteroidota       166063        137116.    1012126.              570468
5 Bdellovibrionota    20443         19454.       9624.               19647
6 Chloroflexi        391859        335486      257714.               51769
```

These are absolute counts of amplicon sequence variants (ASVs) summed by Phylum, measured in Lake Ontario at two times (May vs September) and at two depths (Deep vs Shallow). The data is also in what we call a "wide" format - the values in one of our variables (Depth/Month group) are spread out across multiple columns as column names. While this format is often preferred when entering data, we'll need to convert our data to "long" format in order to use ggplot.

```r
long_counts <- pivot_longer(phylum_counts,
                 cols = Deep_May:Shallow_September,
                 names_to = "Depth_and_Month",
                 values_to = "Counts")

head(long_counts)
```

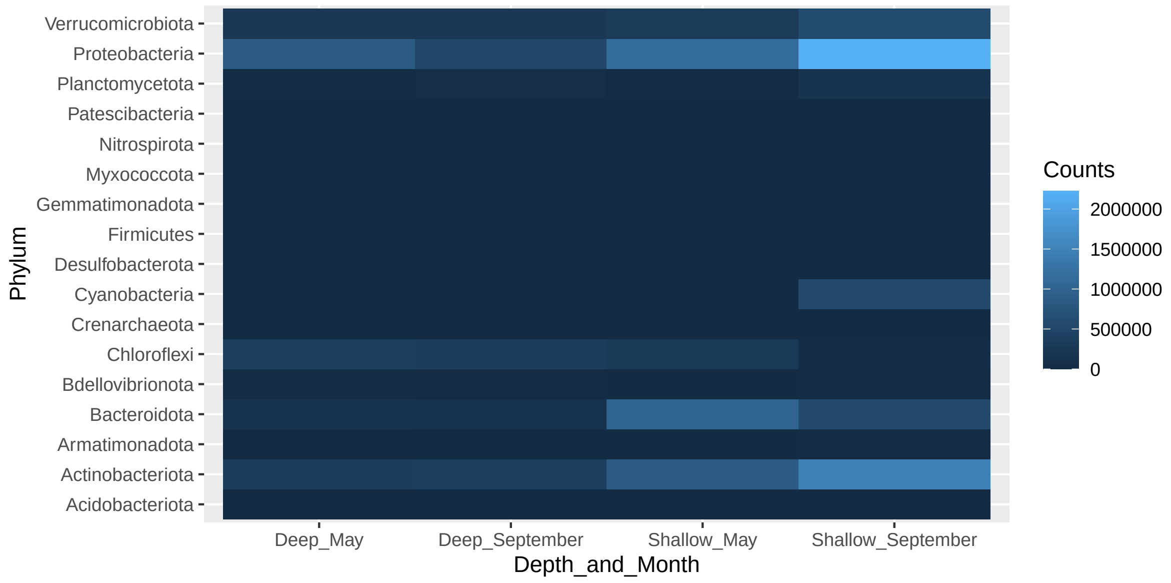
```
# A tibble: 6 x 3
  Phylum           Depth_and_Month    Counts
  <chr>            <chr>               <dbl>
1 Acidobacteriota  Deep_May             9840
2 Acidobacteriota  Deep_September       8156
3 Acidobacteriota  Shallow_May          7682.
4 Acidobacteriota  Shallow_September    6837
5 Actinobacteriota Deep_May           336139
6 Actinobacteriota Deep_September     385255
```

That's looking much better! Phylum and Depth_and_Month will be our two axes that define the grid of our heatmap, and Counts will be the variable that we map to the color in our heatmap.

Make our first heatmap

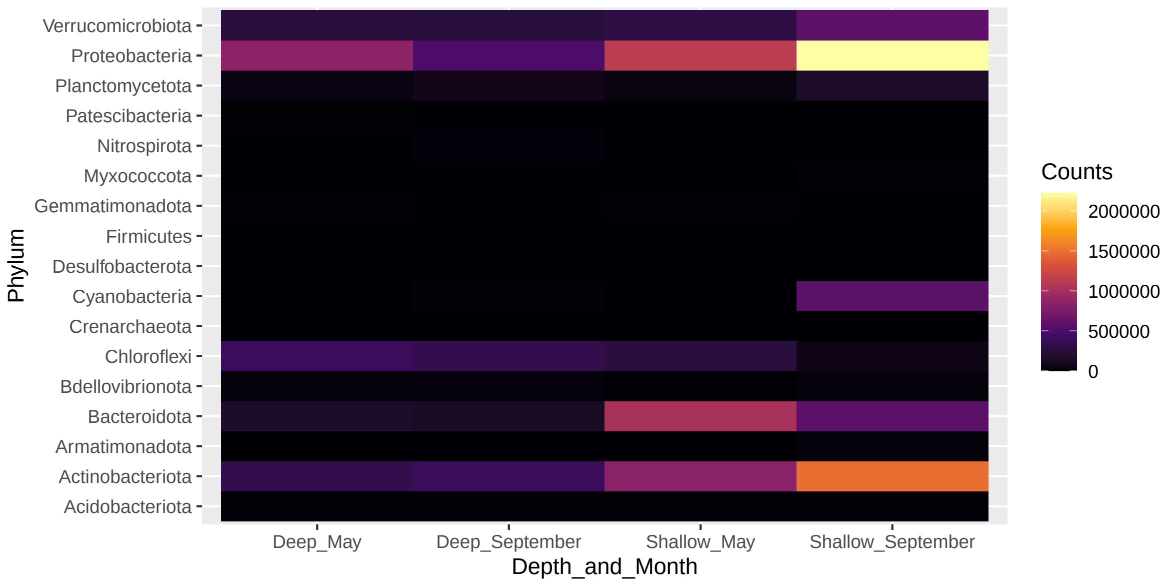
To make a heatmap, we'll use our two categorical variables as our x and y axes, and our continuous variable ("Counts") to define the how "intense" of a color each cell is filled with. We'll use `geom_tile` to construct the map.

```
long_counts %>%
  ggplot(aes(x = Depth_and_Month, y = Phylum, fill = Counts)) +
  geom_tile()
```



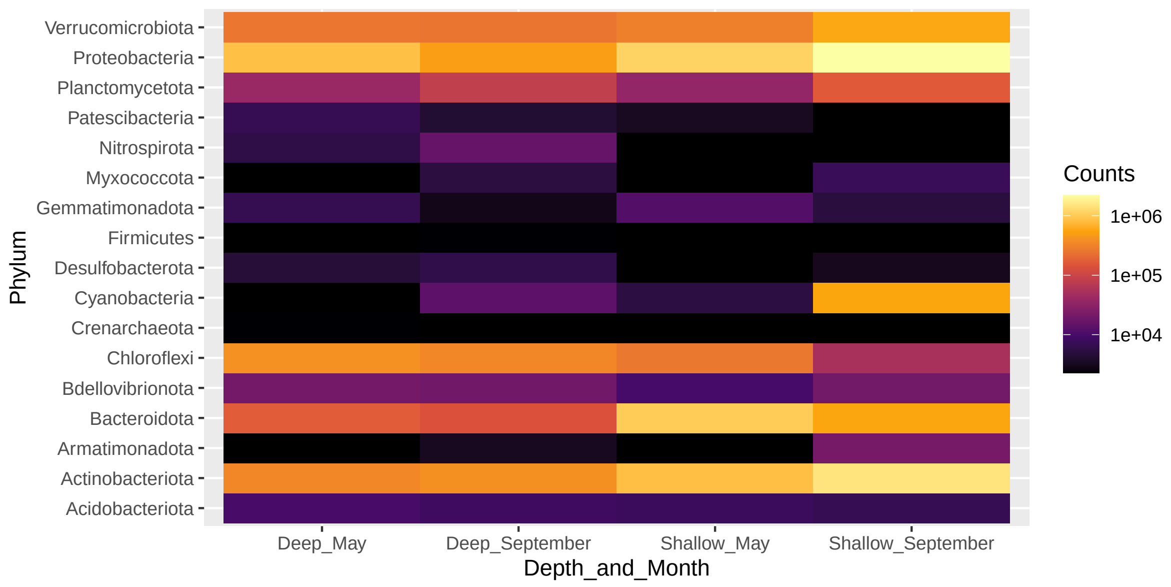
That's a good start! Lighted areas have higher counts compared to darker areas. Personally, I do not love this palette - I think it is harder to see contrast. Let's adjust which colors we fill in.

```
long_counts %>%
  ggplot(aes(x = Depth_and_Month, y = Phylum, fill = Counts)) +
  geom_tile() +
    scale_fill_viridis_c(option = "inferno")
```



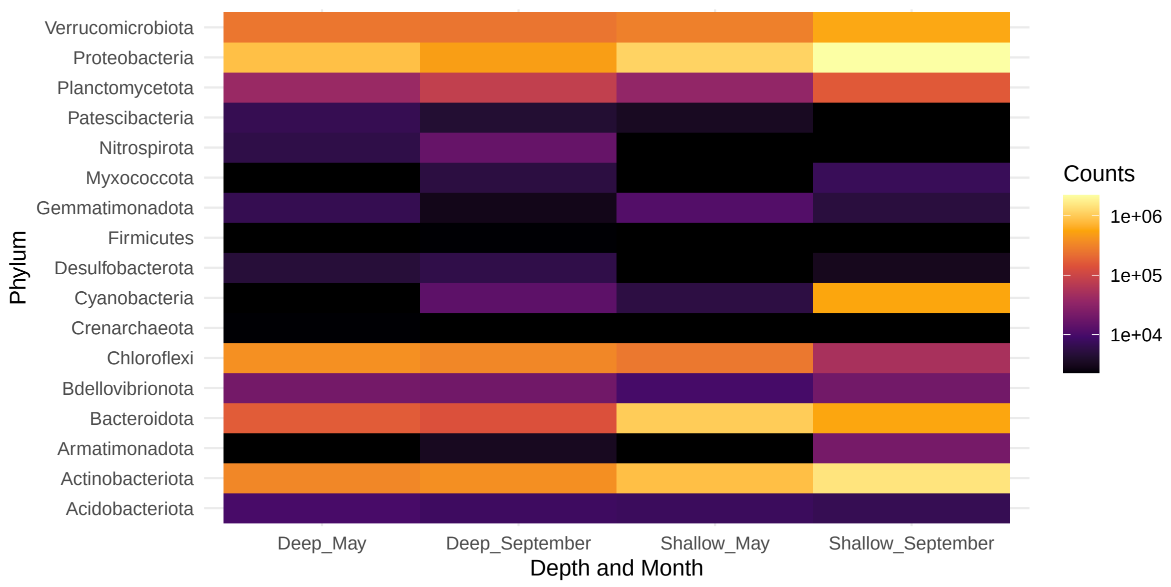
Okay, getting better! We see that variations in a highly abundant phylum (Proteobacteria) is obscuring changes in other, less abundant taxa. Perhaps a log scale would be better. I also change the zero values to blank, instead of the default grey.

```
long_counts %>%
  ggplot(aes(x = Depth_and_Month, y = Phylum, fill = Counts)) +
  geom_tile() +
  scale_fill_viridis_c(option = "inferno",
                       transform = "log10",
                       na.value = "black")
```



Looking good! Let's do some final aesthetic tune-ups.

```
long_counts %>%
  ggplot(aes(x = Depth_and_Month, y = Phylum, fill = Counts)) +
  geom_tile() +
  scale_fill_viridis_c(option = "inferno",
                       transform = "log10",
                       na.value = "black") +
  labs(x = "Depth and Month") +
  theme_minimal()
```



That looks pretty good! But wait… our goal is to look for patterns and groupings. We can start to figure out some of the main findings by slowly scanning row by row, but it's hard to seem immediate patterns between groupings.

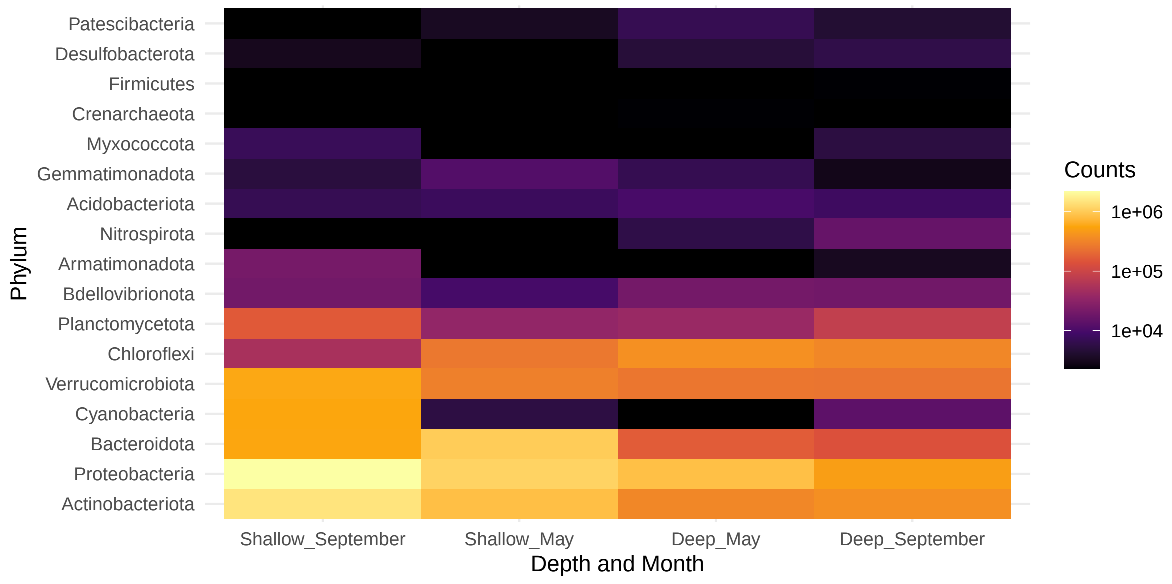
One of the best things we can do to make our heat maps more approachable is to cluster or arrange our rows and columns so that more similar (rows) and more similar (columns) are next to each other. Obviously, there are caveats here - if one axis is ordinal (like time, or dosage) you'll probably avoid scrambling the rows. For our data, however, I'd like to maximize keeping similar groups together.

There are a few ways we can do this. `pheatmap` is another plotting library you can check out, which can do this automatically. Or, you can do it by hand. In today's lesson, we'll split the difference. We'll use a function that Gus wrote which helps us rearrange those columns. To load this function, we'll use a `source` command.

```r
source("data/reorder_variable_by_clustering.R")
```

Now, we'll use the function to first rearrange the Phylum, and then to rearrange the Depth and Month, and then we'll pass that into ggplot.[1]

```
long_counts %>%
  reorder_variable_by_clustering("Phylum","Depth_and_Month","Counts") %>%
  reorder_variable_by_clustering("Depth_and_Month","Phylum","Counts") %>%
  ggplot(aes(x = Depth_and_Month, y = Phylum, fill = Counts)) +
  geom_tile() +
  scale_fill_viridis_c(option = "inferno",
                       transform = "log10",
                       na.value = "black") +
  labs(x = "Depth and Month") +
  theme_minimal()
```



That's looking much better!

[1]Under the hood, this function is converting our dataframe into a matrix, using a hierarchical clustering method to create a dendrogram, of values in our given variable. Then, we're converting that variable to a factor and setting the levels of that factor to be the same as the order of variables in our dendrogram, so that more similar samples are close together.