Customizing plot appearance with themes

## Learning Objectives

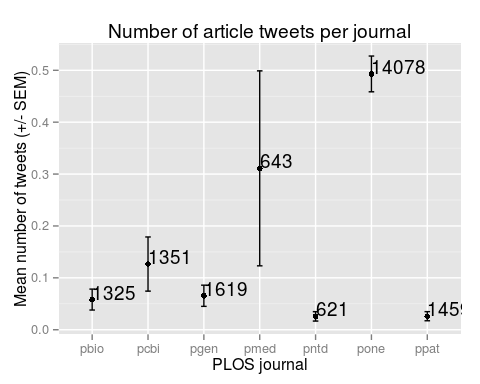
* Add title and axis labels to a plot
* Use themes to change plot appearance
* Modify the position of the legend

We have learned how to translate our data into a visualization using ggplot2. However, we have not spent much time adjusting the plot appearance. Not surprisingly, ggplot2 offers these capabilities as well.

### Labeling a plot

We can add a title and axis labels using the ggplot2 function labs (for labels). Let's modify the plot we created in the last lesson that showed the mean number of article tweets per journal.

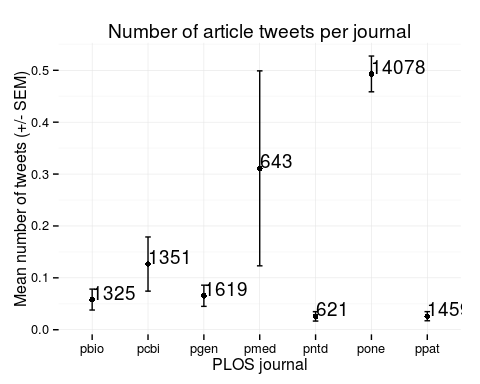
tweets\_point <- ggplot(tweets\_per\_journal, aes(x = journal, y = mean)) +  
 geom\_point() +   
 geom\_errorbar(aes(ymin = mean - sem, ymax = mean + sem), width = 0.1) +  
 geom\_text(aes(label = num), hjust = 0, vjust = 0) +  
 labs(title = "Number of article tweets per journal",  
 x = "PLOS journal", y = "Mean number of tweets (+/- SEM)")  
tweets\_point



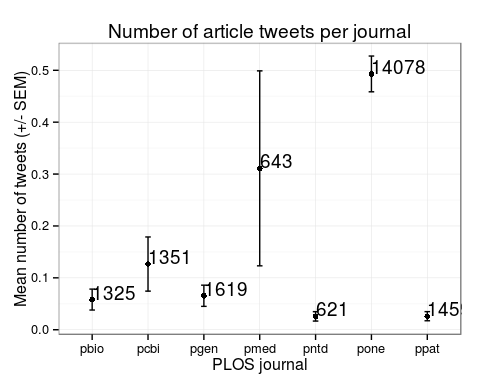
### Using built-in themes

By default all ggplot2 graphics have a gray background with white gridlines. This is known as theme\_gray(), but we can easily change this if we have a different preference. There are multiple other built-in themes that can be applied by adding them to the plot.

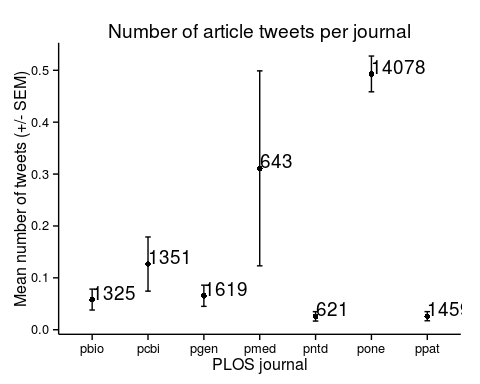
tweets\_point + theme\_minimal() # Minimal theme



tweets\_point + theme\_bw() # Black and white theme

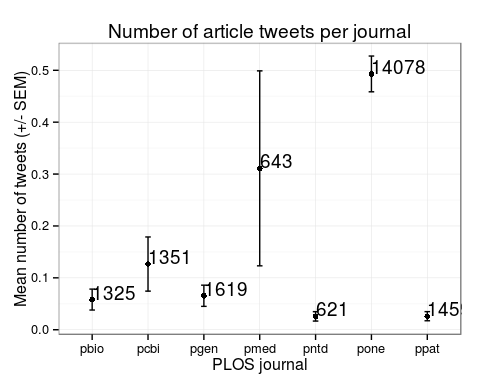


tweets\_point + theme\_classic() # Classic theme



If we plan to make multiple plots that all need the same theme, we can use theme\_set so that they are automatically applied to all plots.

theme\_set(theme\_bw())  
tweets\_point



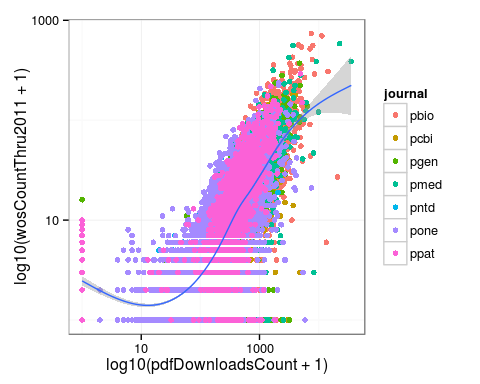
### Modifying the legend

In theory, every aspect of the plot can be modified. But memorizing how to control every feature of the plot is not worth the effort. It can be quite complex, and you are likely only going to need to make a few changes. Because ggplot2 is so popular, there are many online resources explaining how to accomplish specific tasks.

Here we'll show some common modifications of the legend. We'll use our plot of 2011 citations versus PDF downloads.

p <- ggplot(research, aes(x = log10(pdfDownloadsCount + 1),  
 y = log10(wosCountThru2011 + 1))) +  
 geom\_point(aes(color = journal)) +  
 geom\_smooth() +  
 scale\_x\_continuous(breaks = c(1, 3), labels = c(10, 1000)) +  
 scale\_y\_continuous(breaks = c(1, 3), labels = c(10, 1000))  
p

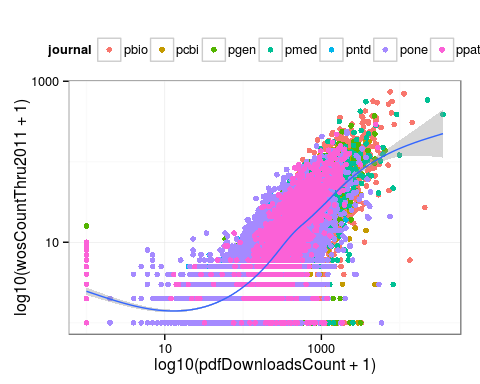
geom\_smooth: method="auto" and size of largest group is >=1000, so using gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the smoothing method.



By default the legend is placed to the right of the plot. We can control this using the legend.position argument to theme. We can use a key word like "left", "right", "top", or "bottom".

p + theme(legend.position = "top")

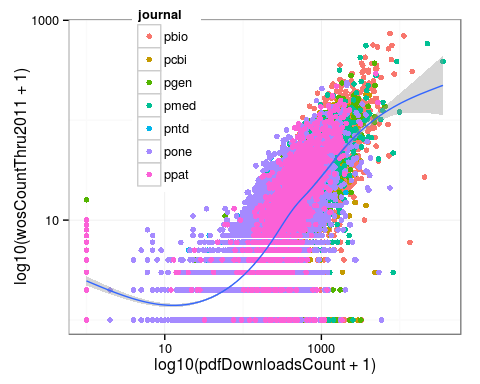
geom\_smooth: method="auto" and size of largest group is >=1000, so using gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the smoothing method.



We can specify the position of the center of the legend relative to the entire plot, where (0, 0) is the bottom left and (1, 1) is the top right.

p + theme(legend.position = c(0.25, 0.75))

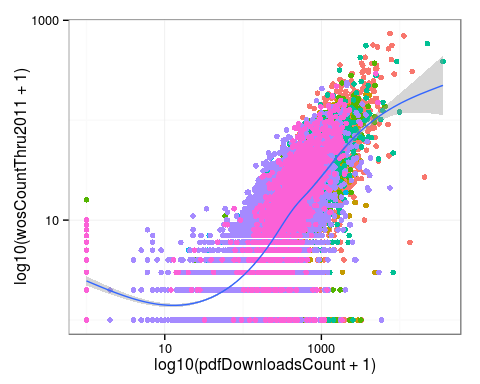
geom\_smooth: method="auto" and size of largest group is >=1000, so using gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the smoothing method.



Or we can remove the legend.

p + theme(legend.position = "none")

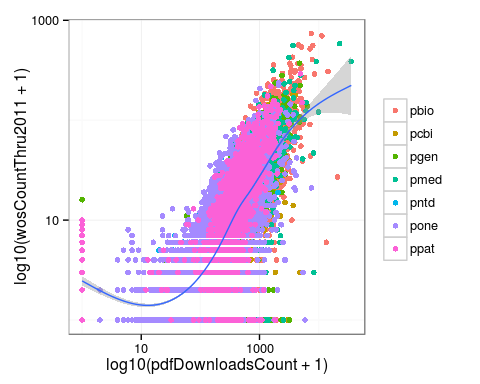
geom\_smooth: method="auto" and size of largest group is >=1000, so using gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the smoothing method.



Lastly, we can maintain the legend but remove only the legend title by setting the argument legend.title to element\_blank.

p + theme(legend.title = element\_blank())

geom\_smooth: method="auto" and size of largest group is >=1000, so using gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the smoothing method.



The theme is composed of theme elements, e.g. theme\_text controls the text appearance, theme\_line controls the line appearance, etc. This can be complicated, so it is best to use one of the built-in themes when first beginning to use ggplot2.