Controlling the plot scales

## Learning Objectives

* Scales affect how data is mapped to aesthetics
* Use variants of scale\_x\_\* and scale\_y\_\* to control axes
* Use variants of scale\_color\_\* to control color scheme

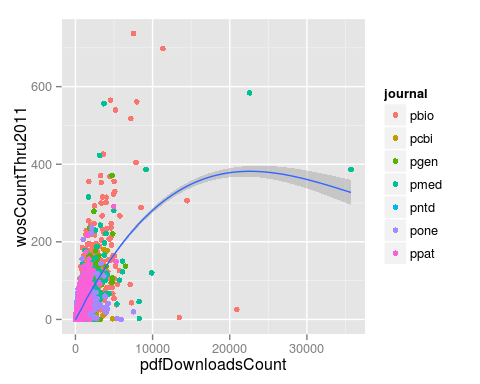
In the last lesson we made our first plots with ggplot2 by mapping columns of a data frame to the plot aesthetics. We can modify the transformation of the data to the plot using scales. These allow us to customize the display of the data.

### Scales that affect mapping to x and y

In the last lesson we plotted the number of PDF downloads versus the 2011 citation count.

p <- ggplot(research, aes(x = pdfDownloadsCount,  
 y = wosCountThru2011)) +  
 geom\_point(aes(color = journal)) +  
 geom\_smooth()  
p

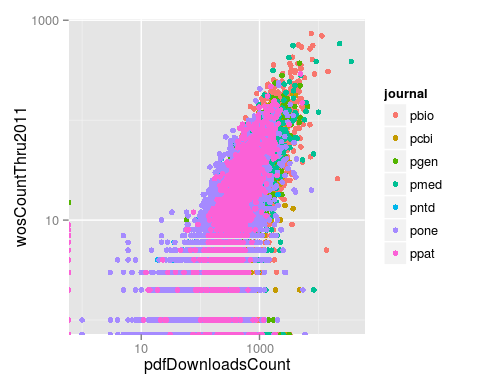
geom\_smooth: method="auto" and size of largest group is >=1000, so using gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the smoothing method.



This was not a very effective visualization because the majority of the points were clustered in the bottom left part of the graph. To improve it, we can log transform the axes. We do this by adding the ggplot2 functions scale\_x\_log10 and scale\_y\_log10.

p + scale\_x\_log10() + scale\_y\_log10()

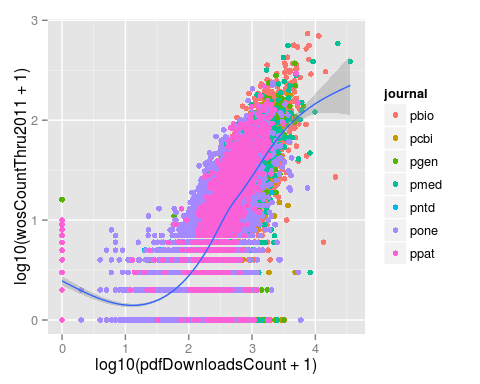
geom\_smooth: method="auto" and size of largest group is >=1000, so using gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the smoothing method.



However because of all the zeros in the data, the loess curve failed to work. Instead of log transforming using the scales, instead we can directly log transform the columns when specifying the aesthetic mappings. This allows us to add the pseudocount to avoid taking the log of zero.

p <- ggplot(research, aes(x = log10(pdfDownloadsCount + 1),  
 y = log10(wosCountThru2011 + 1))) +  
 geom\_point(aes(color = journal)) +  
 geom\_smooth()  
p

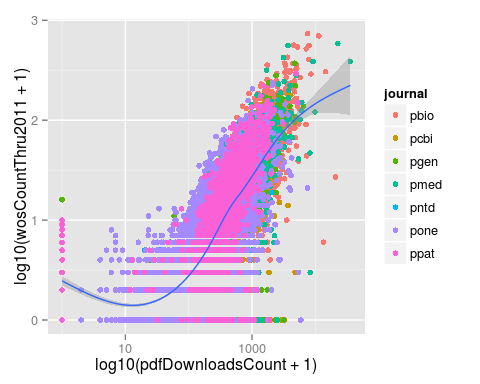
geom\_smooth: method="auto" and size of largest group is >=1000, so using gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the smoothing method.



This gives us the same result plus the loess curve. One big difference between the two approaches is the labeling of the axes. When we log transformed using the scale functions, the axis labels were automatically updated to reflect the transformation. We can manually update the axis labels using scale\_x\_continuous. To mimic the axes of the earlier plot, we create breaks 1 and 3 which correspond to data values of 10 (101) and 1000 (103).

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))  
p

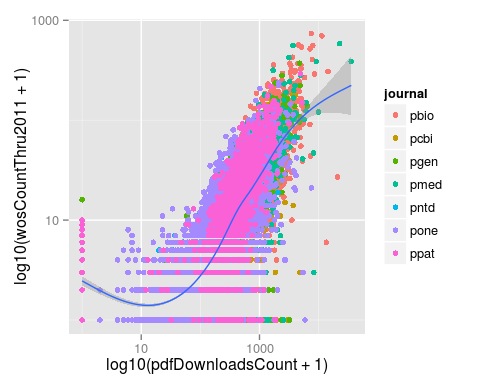
geom\_smooth: method="auto" and size of largest group is >=1000, so using gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the smoothing method.



And then we can do the same for the y-axis.

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.

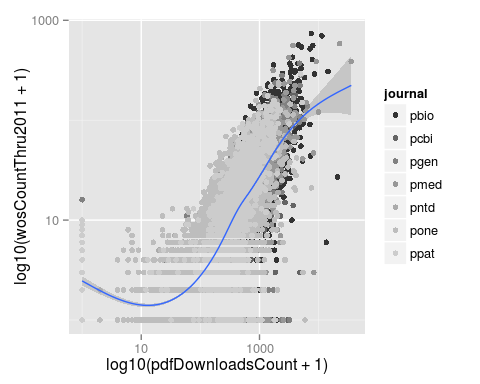


### Scales that affect mapping to color

When we mapped the column journal to the color of the points, ggplot2 assigned its default colors. Using scale functions, we can control the colors used in the plot. These all have the form scale\_color\_\*. For example, we could use grayscale.

p + scale\_color\_grey()

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.



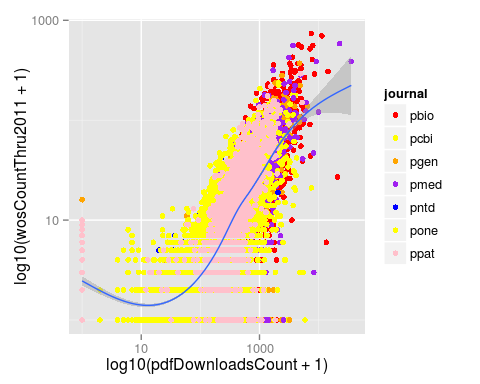
Or we could manually specify each of the colors for the seven journals. The colors are assigned to the factor levels in the order they are listed in the output from levels.

levels(research$journal)

[1] "pbio" "pcbi" "pgen" "pmed" "pntd" "pone" "ppat"

p + scale\_color\_manual(values = c("red", "yellow", "orange",  
 "purple", "blue", "yellow",  
 "pink"))

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.



But as that example demonstrates, choosing good colors to aid visualization requires some thought. Luckily others have already created nice color palettes for visualization. ggplot2 provides easy access to the ColorBrewer color palettes developed by Cynthia A. Brewer and colleagues at Pennsylvania State University. Go to [<http://colorbrewer2.org>](http://colorbrewer2.org) to view them. Alternatively you can view them directly in R if you install the package [RColorBrewer](https://cran.r-project.org/web/packages/RColorBrewer/index.html).

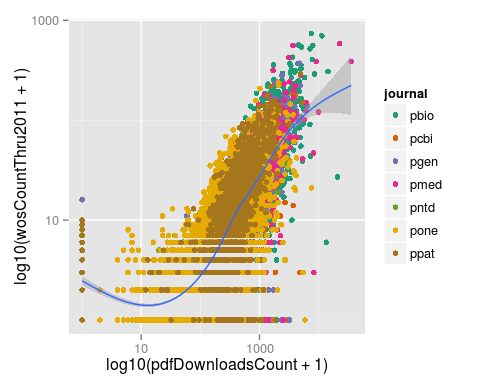
library("RColorBrewer")  
display.brewer.all(type = "qual")



These are the options available for color coding qualitative differences. Let's try the palette "Dark2".

p + scale\_color\_brewer(palette = "Dark2")

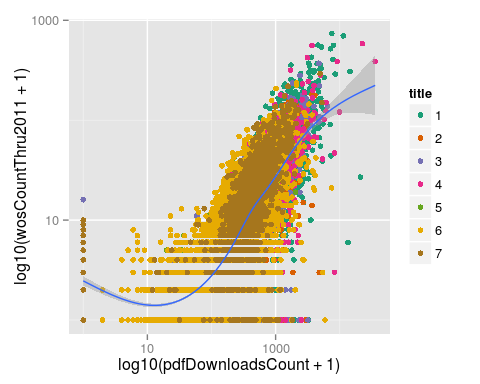
geom\_smooth: method="auto" and size of largest group is >=1000, so using gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the smoothing method.



Lastly, if we want the descriptions in the legend to be different than the shorthand we use for the raw data, we can do this via any of the scale\_color\_\* functions using the arguments labels and name. Here we'll just change the labels to the numbers 1 through 7 and the title to "title" to illustrate the point.

p + scale\_color\_brewer(palette = "Dark2", labels = 1:7, name = "title")

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.



### Challenge

## Modifying the scales

Update the plot to use a square root transformation instead of log10. Also color the points using the ColorBrewer palette "Accent".

p <- ggplot(research, aes(x = pdfDownloadsCount,  
 y = wosCountThru2011)) +  
 geom\_point(aes(color = journal)) +  
 geom\_smooth() +  
 scale\_x\_sqrt() +  
 scale\_y\_sqrt() +  
 scale\_color\_brewer(palette = "Accent")  
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.

