Creating different plots with geoms

## Learning Objectives

* Map a categorical (discrete) variable to an axis
* Create a boxplot and barplot
* Add error bars and text annotations

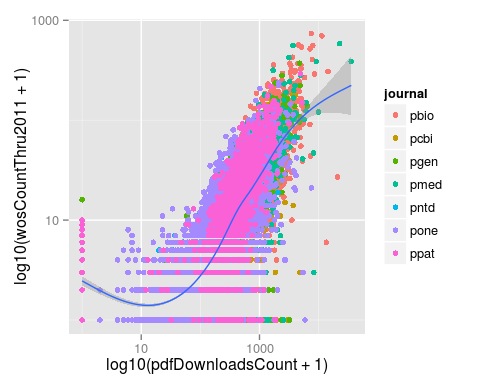
To this point we have mainly focused on a scatter plot created using geom\_point. ggplot2 offers many different geoms to create a wide variety of plots. In this lesson, we'll explore a few more geoms.

### Mapping categorical (discrete) variables

The scatter plot we made mapped continuous variables to the x and y axes.

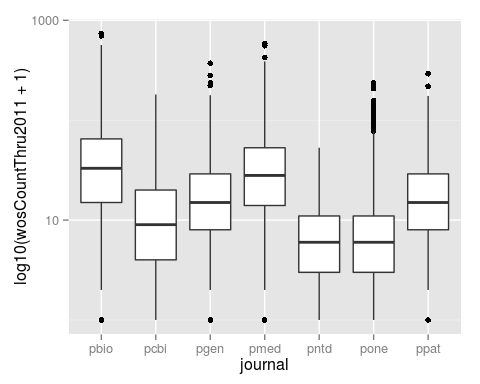
p <- ggplot(research, aes(x = log10(pdfDownloadsCount + 1),  
 y = log10(wosCountThru2011 + 1))) +  
 geom\_point(aes(color = journal)) +  
 geom\_smooth() +  
 scale\_x\_continuous(breaks = c(1, 3), labels = c(10, 1000)) +  
 scale\_y\_continuous(breaks = c(1, 3), labels = c(10, 1000))  
p

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.



Of course we can also map categorical (also known as discrete) variables.

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



### Creating a barplot

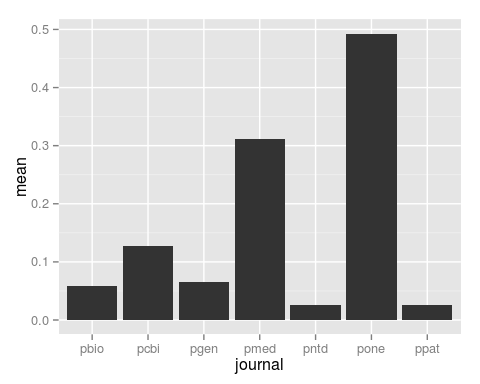
A common figure in scientific publications is the barplot, where the height of the bars represent the mean and the error bars represent the standard error of the mean (SEM). Recall from the challenge in the dplyr section that we calculated the mean and SEM for the number of article tweets per journal.

tweets\_per\_journal <- research %>%  
 group\_by(journal) %>%  
 summarize(num = n(),  
 mean = mean(backtweetsCount),  
 sem = sd(backtweetsCount) / sqrt(num))  
tweets\_per\_journal

Source: local data frame [7 x 4]  
  
 journal num mean sem  
1 pbio 1325 0.05811321 0.020153395  
2 pcbi 1351 0.12657291 0.052177184  
3 pgen 1619 0.06547251 0.020408525  
4 pmed 643 0.31104199 0.187868371  
5 pntd 621 0.02576490 0.009057697  
6 pone 14078 0.49303878 0.034484187  
7 ppat 1459 0.02604524 0.008807428

Let's create a barplot to display this result using geom\_bar.

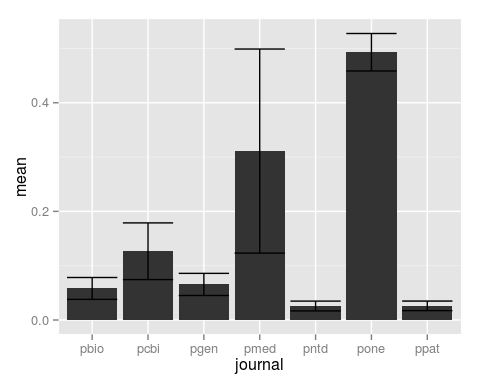
tweets\_bar <- ggplot(tweets\_per\_journal, aes(x = journal, y = mean)) +  
 geom\_bar(stat = "identity")  
tweets\_bar



We have to set the argument stat = "identity" because by default geom\_bar tabulates the number of occurrences of the variable mapped to x and maps this frequency to y.

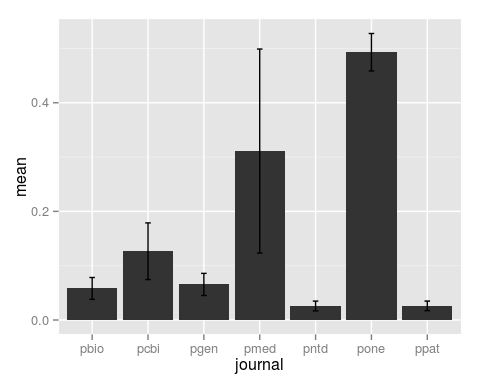
Now let's add the error bars using geom\_errorbar. We pass it values for ymin and ymax, which define the range of the error bars.

tweets\_bar <- ggplot(tweets\_per\_journal, aes(x = journal, y = mean)) +  
 geom\_bar(stat = "identity") +  
 geom\_errorbar(aes(ymin = mean - sem, ymax = mean + sem))  
tweets\_bar



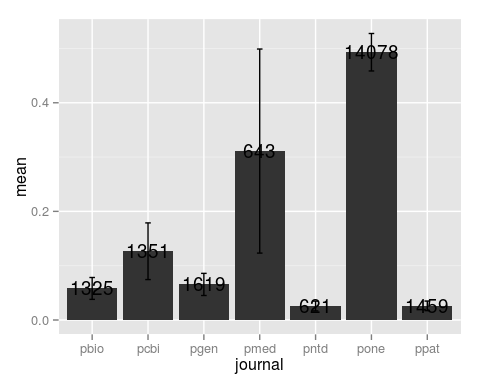
By default the error bars are as wide as the bars. Let's reduce their size using the argument width.

tweets\_bar <- ggplot(tweets\_per\_journal, aes(x = journal, y = mean)) +  
 geom\_bar(stat = "identity") +  
 geom\_errorbar(aes(ymin = mean - sem, ymax = mean + sem), width = 0.1)  
tweets\_bar



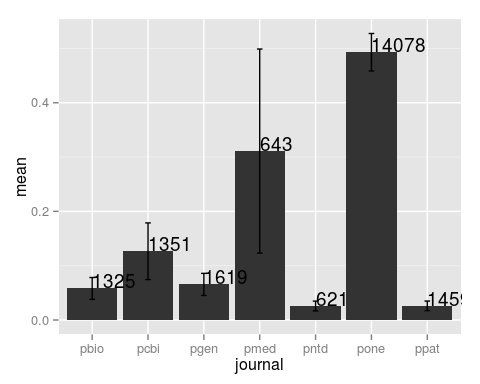
Since we also calculated the sample size per journal, let's add the number of articles to the plot using geom\_text.

tweets\_bar <- ggplot(tweets\_per\_journal, aes(x = journal, y = mean)) +  
 geom\_bar(stat = "identity") +  
 geom\_errorbar(aes(ymin = mean - sem, ymax = mean + sem), width = 0.1) +  
 geom\_text(aes(label = num))  
tweets\_bar



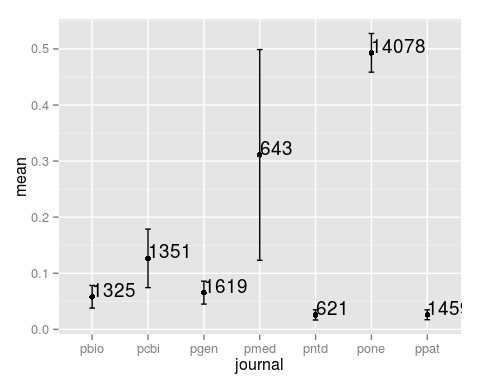
These are difficult to read because the text boxes are centered on the coordinate that marks the top of each bar. The positioning of the text box is controlled by the arguments hjust and vjust, for horizontal and vertical justification. By default these are both set to 0.5. We can image the text box being defined from 0 to 1 from left to right for hjust and from 0 to 1 from bottom to top for vjust. The values we set for hjust and vjust adjust where the text box is situated relative to that coordinate at the top of each bar. We'll set them both to 0 to place the bottom left corner of the text box at that point, and thus the text is above and to the right of the bar.

tweets\_bar <- ggplot(tweets\_per\_journal, aes(x = journal, y = mean)) +  
 geom\_bar(stat = "identity") +  
 geom\_errorbar(aes(ymin = mean - sem, ymax = mean + sem), width = 0.1) +  
 geom\_text(aes(label = num), hjust = 0, vjust = 0)  
tweets\_bar



To make it easier to see this central point, and to show an alternative to a boxplot, we can easily switch from a barplot to using points.

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)  
tweets\_point



Using multiple geoms, we were able to created a plot with multiple layers to display multiple sources of information, the mean, SEM, and sample size.

### Challenges

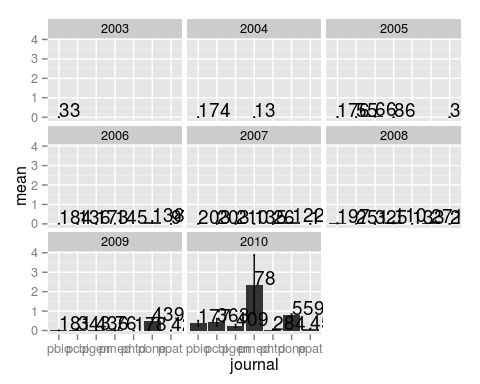
## Mean number of tweets per journal per year

Modify the dplyr code above to calculate the mean, SEM, and sample size of the number of article tweets per journal *and* per year. Use facet\_wrap to make a separate subplot per year.

tweets\_per\_journal\_year <- research %>%  
 group\_by(journal, year) %>%  
 summarize(num = n(),  
 mean = mean(backtweetsCount),  
 sem = sd(backtweetsCount) / sqrt(num))  
tweets\_per\_journal\_year

Source: local data frame [42 x 5]  
Groups: journal  
  
 journal year num mean sem  
1 pbio 2003 33 0.000000000 0.000000000  
2 pbio 2004 174 0.000000000 0.000000000  
3 pbio 2005 176 0.011363636 0.008012313  
4 pbio 2006 184 0.010869565 0.007664915  
5 pbio 2007 203 0.004926108 0.004926108  
6 pbio 2008 197 0.030456853 0.018916893  
7 pbio 2009 181 0.005524862 0.005524862  
8 pbio 2010 177 0.367231638 0.146996329  
9 pcbi 2005 55 0.000000000 0.000000000  
10 pcbi 2006 136 0.000000000 0.000000000  
.. ... ... ... ... ...

ggplot(tweets\_per\_journal\_year, aes(x = journal, y = mean)) +  
 geom\_bar(stat = "identity") +  
 geom\_errorbar(aes(ymin = mean - sem, ymax = mean + sem), width = 0.1) +  
 geom\_text(aes(label = num), hjust = 0, vjust = 0) +  
 facet\_wrap(~year)

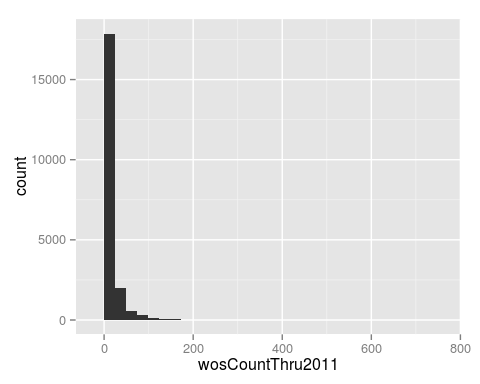


## Visualizing a single distribution

The geoms geom\_histogram and geom\_density can be used to create histograms and density plots, respectively. Using these geoms, visualize the distribution of 2011 citations (wosCountThru2011). Compare the raw distribution to log10 and square root transformations.

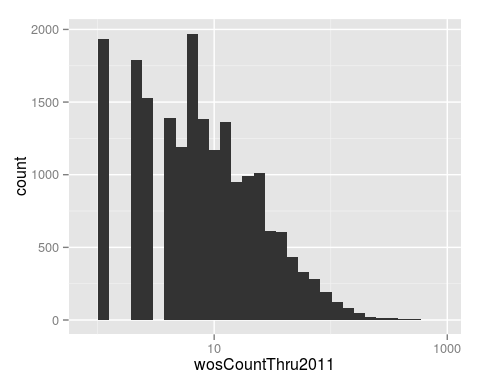
p <- ggplot(research, aes(x = wosCountThru2011))  
p + geom\_histogram()

stat\_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.



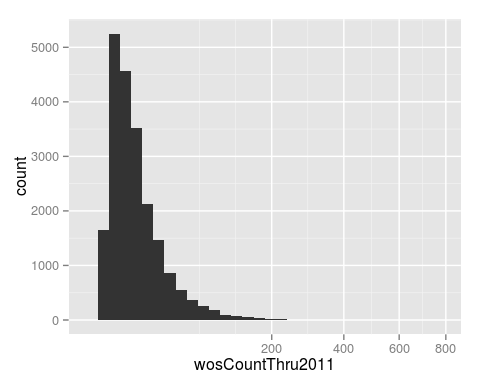
p + geom\_histogram() + scale\_x\_log10()

stat\_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.

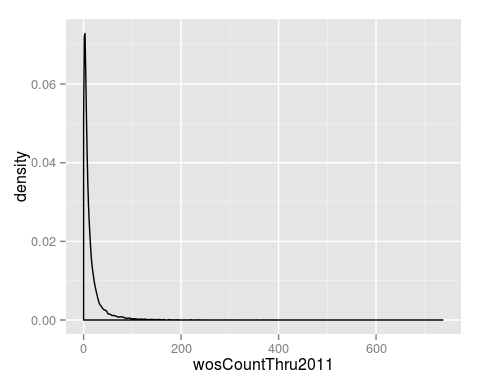


p + geom\_histogram() + scale\_x\_sqrt()

stat\_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.

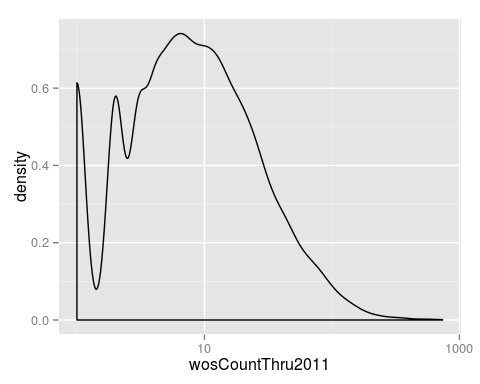


p + geom\_density()



p + geom\_density() + scale\_x\_log10()

Warning: Removed 1654 rows containing non-finite values (stat\_density).



p + geom\_density() + scale\_x\_sqrt()

