It’s been a long time since I’ve posted a Statistics Sunday post! Now that I’m moved out of my apartment and into my house, I have a bit more time on my hands, but work has been quite busy. Today, I’m preparing for 2 upcoming [standard-setting studies](http://www.deeplytrivial.com/2018/09/statistics-sunday-what-is-standard.html) by drawing a sample of items from 2 of our exams. So I thought I’d share what I’m up to in order to pass on some of these new psychometric tricks I’ve learned to help me with this project.

Because I can’t share data from our item banks, I’ll generate a fake dataset to use in my demonstration. For the exams I’m using for my upcoming standard setting, I want to draw a large sample of items, stratified by both item difficulty (so that I have a range of items across the [Rasch](http://www.deeplytrivial.com/2017/09/great-minds-in-statistics-georg-rasch.html) difficulties) and item domain (the topic from the exam outline that is assessed by that item). Let’s pretend I have an exam with 3 domains, and a bank of 600 items. I can generate that data like this:

domain1 <- data.frame(domain = 1, b = sort(rnorm(200)))  
domain2 <- data.frame(domain = 2, b = sort(rnorm(200)))  
domain3 <- data.frame(domain = 3, b = sort(rnorm(200)))

The variable domain is the domain label, and b is the item difficulty. I decided to sort that varible within each dataset so I can easily see that it goes across a range of difficulties, both positive and negative.

head(domain1)

## domain b  
## 1 1 -2.599194  
## 2 1 -2.130286  
## 3 1 -2.041127  
## 4 1 -1.990036  
## 5 1 -1.811251  
## 6 1 -1.745899

tail(domain1)

## domain b  
## 195 1 1.934733  
## 196 1 1.953235  
## 197 1 2.108284  
## 198 1 2.357364  
## 199 1 2.384353  
## 200 1 2.699168

If I desire, I can easily combine these 3 datasets into 1:

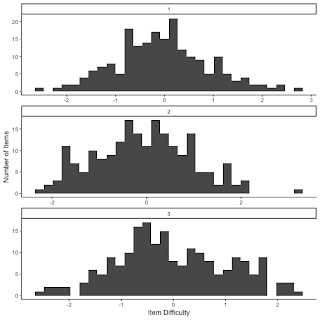
item\_difficulties <- rbind(domain1, domain2, domain3)

I can also easily visualize my item difficulties, by domain, as a group of histograms using ggplot2:

library(tidyverse)

item\_difficulties %>%  
 ggplot(aes(b)) +  
 geom\_histogram(show.legend = FALSE) +  
 labs(x = "Item Difficulty", y = "Number of Items") +  
 facet\_wrap(~domain, ncol = 1, scales = "free") +  
 theme\_classic()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

[](https://i2.wp.com/4.bp.blogspot.com/-hRfW802ko8Y/W8OpSzHrnDI/AAAAAAAAOCc/DpiGUSp0TjsgAQdqzbfn0njsaIxdkpeUgCLcBGAs/s1600/unnamed-chunk-4-1.png?ssl=1)

Now, let’s say I want to draw 100 items from my item bank, and I want them to be stratified by difficulty and by domain. I’d like my sample to range across the potential item difficulties fairly equally, but I want my sample of items to be weighted by the percentages from the exam outline. That is, let’s say I have an outline that says for each exam: 24% of items should come from domain 1, 48% from domain 2, and 28% from domain 3. So I want to draw 24 from domain1, 48 from domain2, and 28 from domain3. Drawing such a random sample is pretty easy, but I also want to make sure I get items that are very easy, very hard, and all the levels in between.

I’ll be honest: I had trouble figuring out the best way to do this with a continuous variable. Instead, I decided to classify items by quartile, then drew an equal number of items from each quartile.

To categorize by quartile, I used the following code:

domain1 <- within(domain1, quartile <- as.integer(cut(b, quantile(b, probs = 0:4/4), include.lowest = TRUE)))

The code uses the quantile command, which you may remember from my post on [quantile regression](http://www.deeplytrivial.com/2018/02/statistics-sunday-quantile-regression.html). The nice thing about using quantiles is that I can define that however I wish. So I didn’t have to divide my items into quartiles (groups of 4); I could have divided them up into more or fewer groups as I saw fit. To aid in drawing samples across domains of varying percentages, I’d probably want to pick a quantile that is a common multiple of the domain percentages. In this case, I purposefully designed the outline so that 4 was a common multiple.

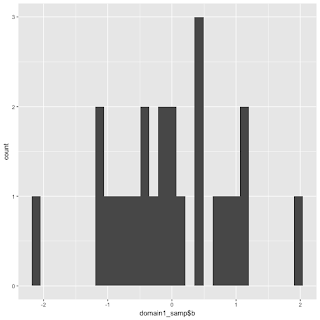
To draw my sample, I’ll use the sampling library (which you’ll want to install with install.packages(“sampling”) if you’ve never done so before), and the strata function.

library(sampling)  
domain1\_samp <- strata(domain1, "quartile", size = rep(6, 4), method = "srswor")

The resulting data frame has 4 variables – the quartile value (since that was used for stratification), the ID\_unit (row number from the original dataset), probability of being selected (in this case equal, since I requested equally-sized strata), and stratum number. So I would want to merge my item difficulties into this dataset, as well as any identifiers I have so that I can pull the correct items. (For the time being, we’ll just pretend row number is the identifier, though this is likely not the case for large item banks.)

domain1$ID\_unit <- as.numeric(row.names(domain1))  
domain1\_samp <- domain1\_samp %>%  
 left\_join(domain1, by = "ID\_unit")  
qplot(domain1\_samp$b)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

[](https://i0.wp.com/1.bp.blogspot.com/-1LT4akxAXBM/W8OpYRS5qDI/AAAAAAAAOCg/_pVYqUvt3IkH045VxHUIaifvWahmqd8CgCLcBGAs/s1600/unnamed-chunk-7-1.png?ssl=1)

For my upcoming study, my sampling technique is a bit more nuanced, but this gives a nice starting point and introduction to what I’m doing.