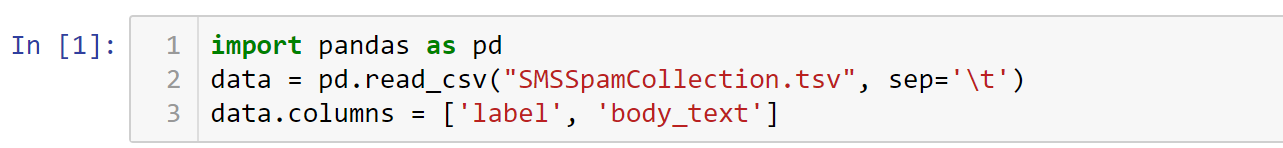
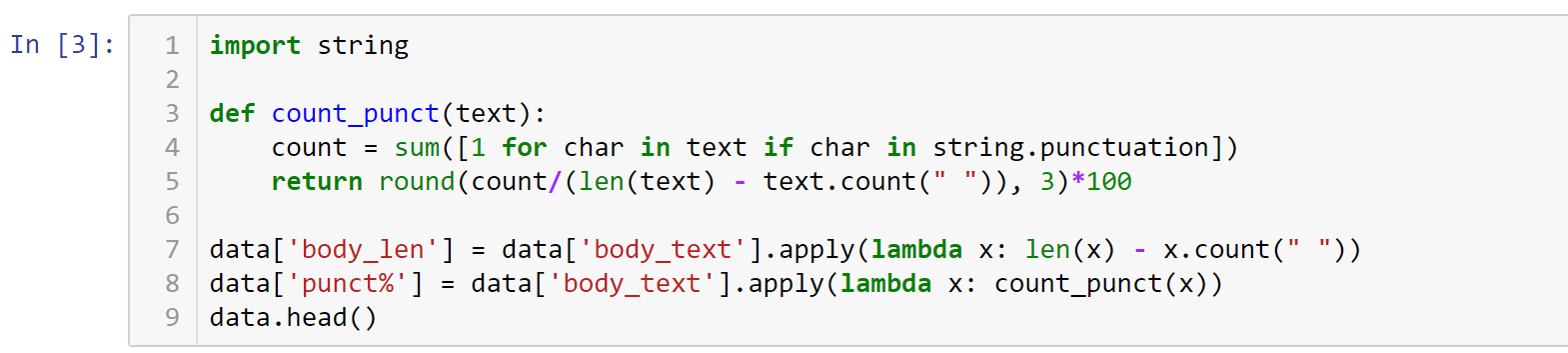
*Start a new Python project folder … do not reuse the previous workspace … As with any keyboard-driven console-like environment, developing muscle -memory for the common commands is also part of the learning curve.*

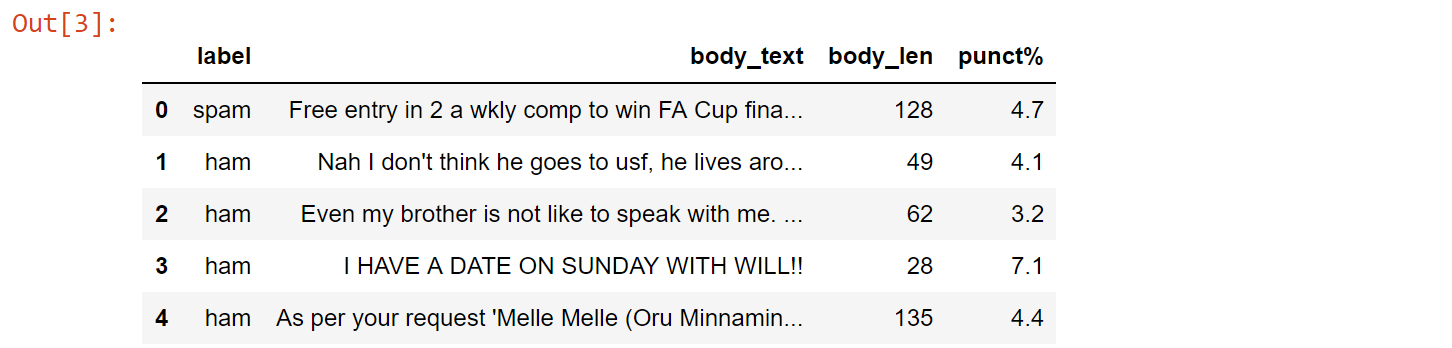
Feature Engineering: Transformations

This exercise extends the work done in 9.1. We are going to take a look at whether either of the two features created in the previous exercise might be a fit for a transformation. Again, this is a pretty large area with deep theoretical underpinnings and we are only going to hit the tip of the iceberg here.



Re-creating the two features … some repetitions in these exercises are intentional for learning purposes …

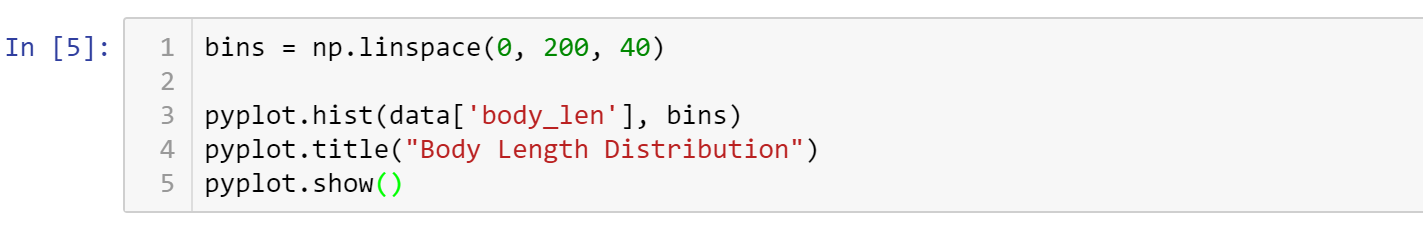


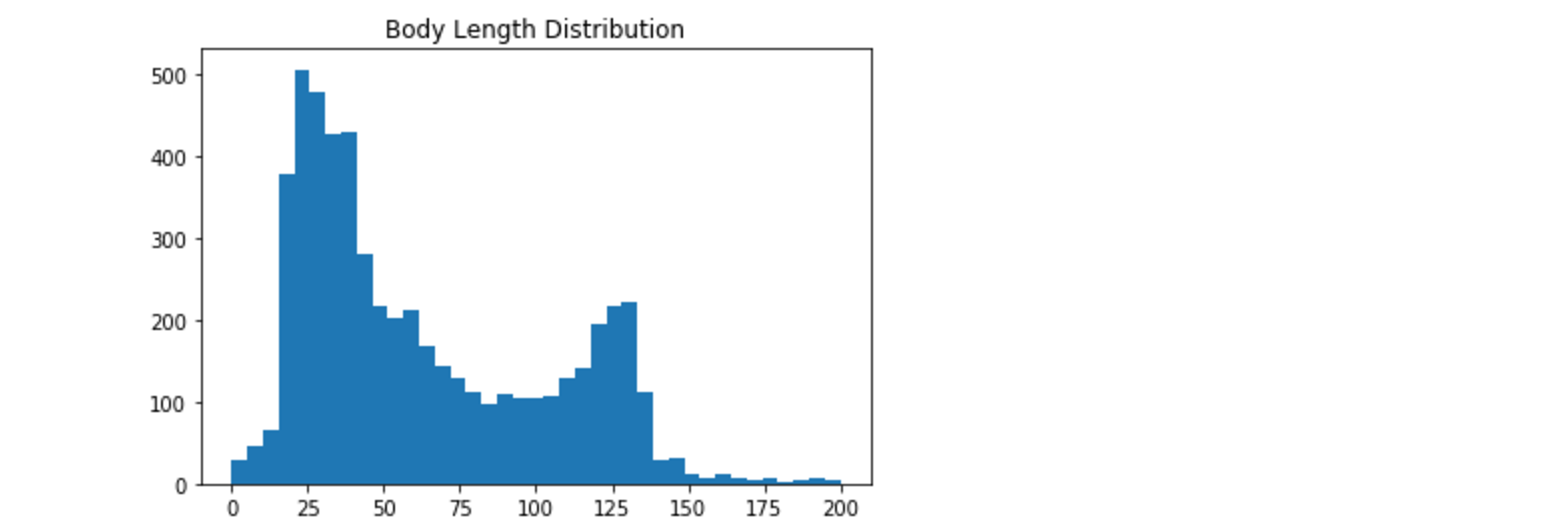


In order to determine whether transformation might be helpful, we can look at the distribution of our data using a histogram. On the last exercise, we looked at the normalized overlayed histograms, but we didn’t look at the full histogram so we are still not exactly sure what the full distribution looks like for these new features. We only know when it is split by label. The first thing we will do is look at those full distributions and then we can determine which one might be a fit for transformation. Now, what we are looking for here is a dramatic skew with a really long or maybe a few outliers. These are the scenarios that would make a feature a prime candidate for transformation.

Plotting the two new features …

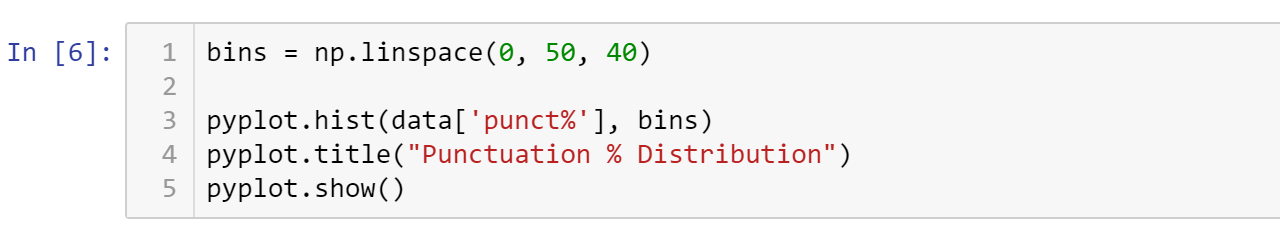


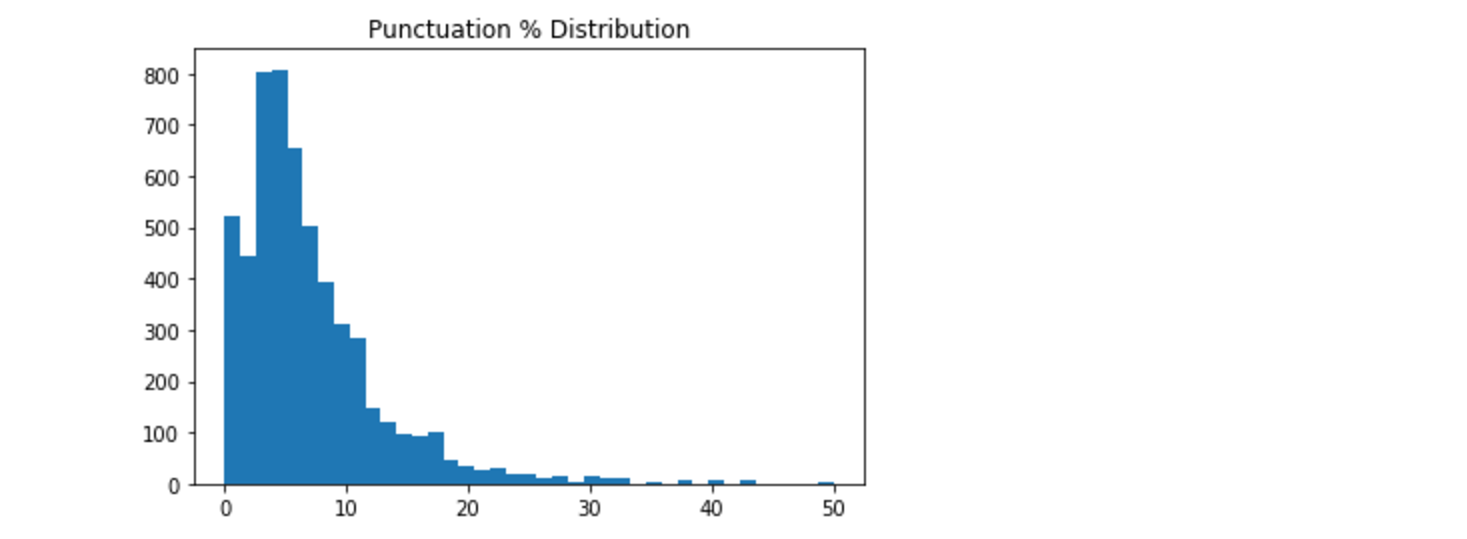




This distribution makes sense. We saw that spam were mostly long so those are the ones concentrated down at spike on the right side. And real text are mostly short so those are the ones kind of concentrated on the left taller spike near zero. So we see this bimodal distribution here with two different spikes.

This is not a great candidate for transformation because it is not really heavily skewed and there is not really any clear outliers. Now let us explore the punctuation percentage feature.

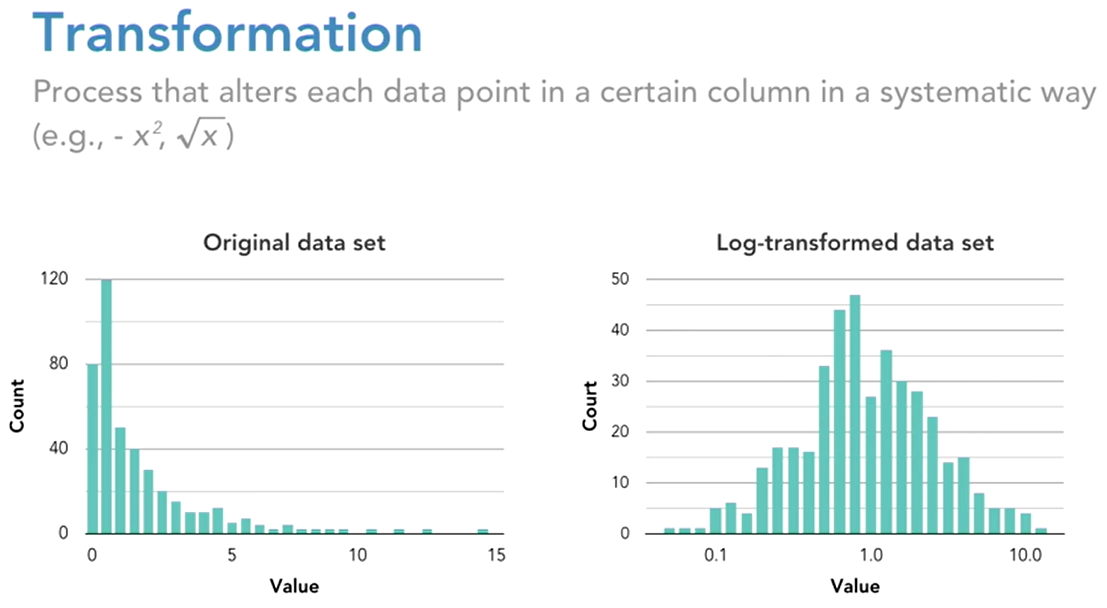




This one could very well be a nice distribution for a transformation. It is fairly skewed in 5 near zero where we see a lot close to zero, and we see a tail extending all the way up to 40 with some of these outliers out in the 50. We are going to focus on this feature for our transformations.

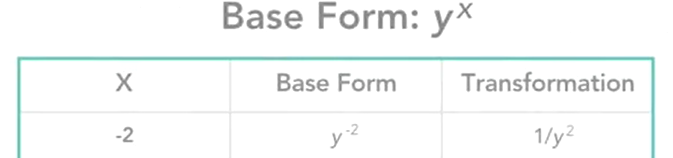
**Transformation**

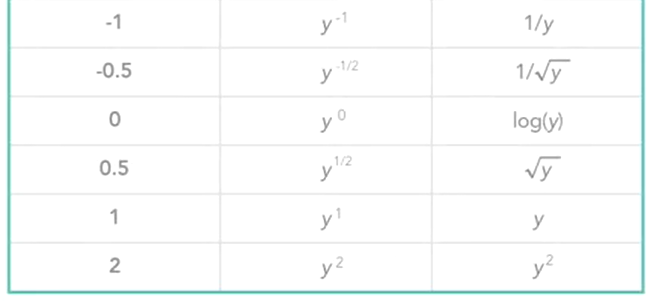
What is a transformation? A transformation is ***a process that alters each data point in a certain column in a systematic way*** (e.g., -x^2, sqr(x)) that makes it cleaner for a model to use. For instance, that could mean squaring each value, or maybe taking the square root of each value in a given column. So let’s say a distribution for a certain feature has a long right tail like this one does in this image. Then the transformation would aim to pull that tail in to make it a more compact distribution.



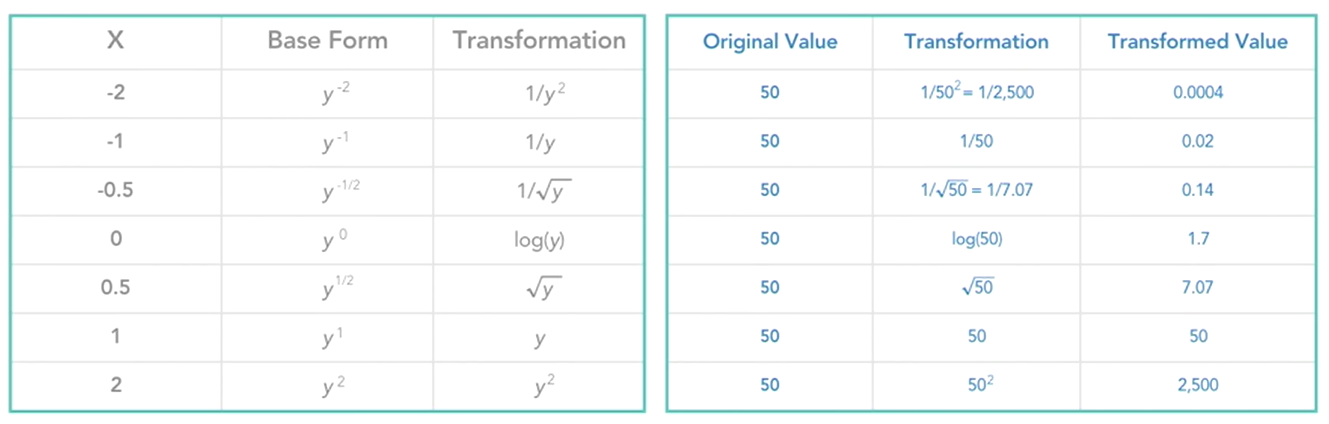
We do this so that the model does not get distracted trying to chase down outliers in a tail. The series of transformation that we will be working with are call the Box-Cox Power Transformations. This is a very common type of transformation. The **base form** **of this type of transformation is y to the x power** (see below), where y is the value in an individual cell, and x is the exponent of the power transformation you are applying. You will notice that this table shows some common power transformations using exponents from negative two up to positive two.





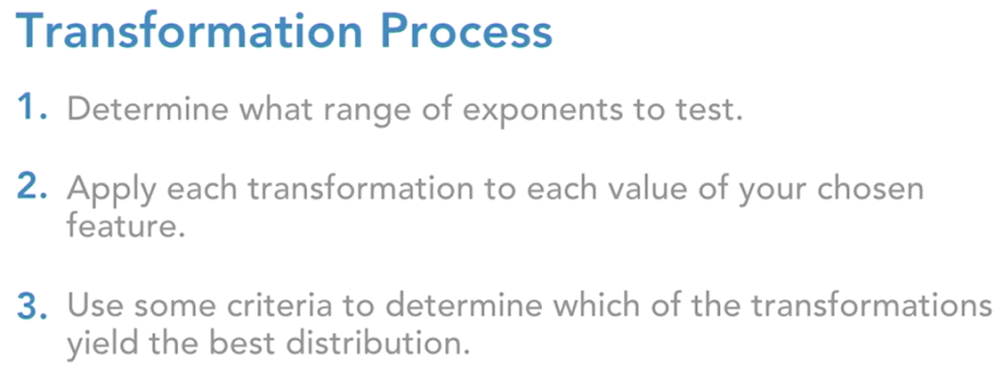


For the first line in the table with an exponent of negative two that translates to y to the negative two which is the same as one over y squared. For example, let’s say that 50% of the characters in a given text message are punctuation. So the value in that cell will be 50.



So let us go through these different transformations and see how that would impact the transformed value. Starting with the first line, one over y squared. So in this example that would be one over 50 squared, or one over 2,500, and that will give you 0.0004.

Then the next transformation is just one over 50 and then it’s one over square root of 50 and so on. So this kind of gives you an idea of how different power transformations alter the original values. In practice, what this process looks likes would be as follows. First you determine what range of exponents you want to test out. So in our example we had a range from negative two to positive two. And that’s a commonly used range. Then you’d apply these transformations to each value in the feature you’d like to transform.

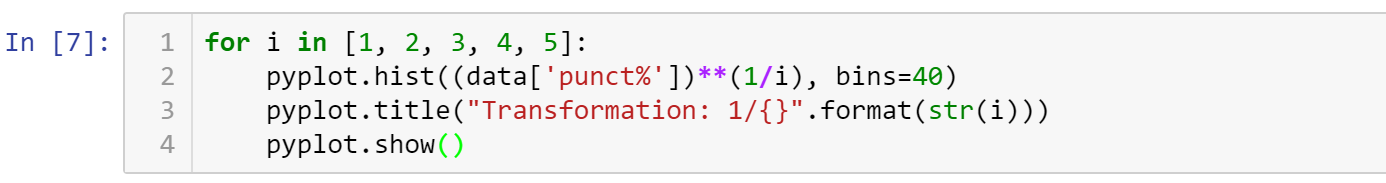


Using positive exponents …

Power 1 means the value itself, value raised to 1 is itself

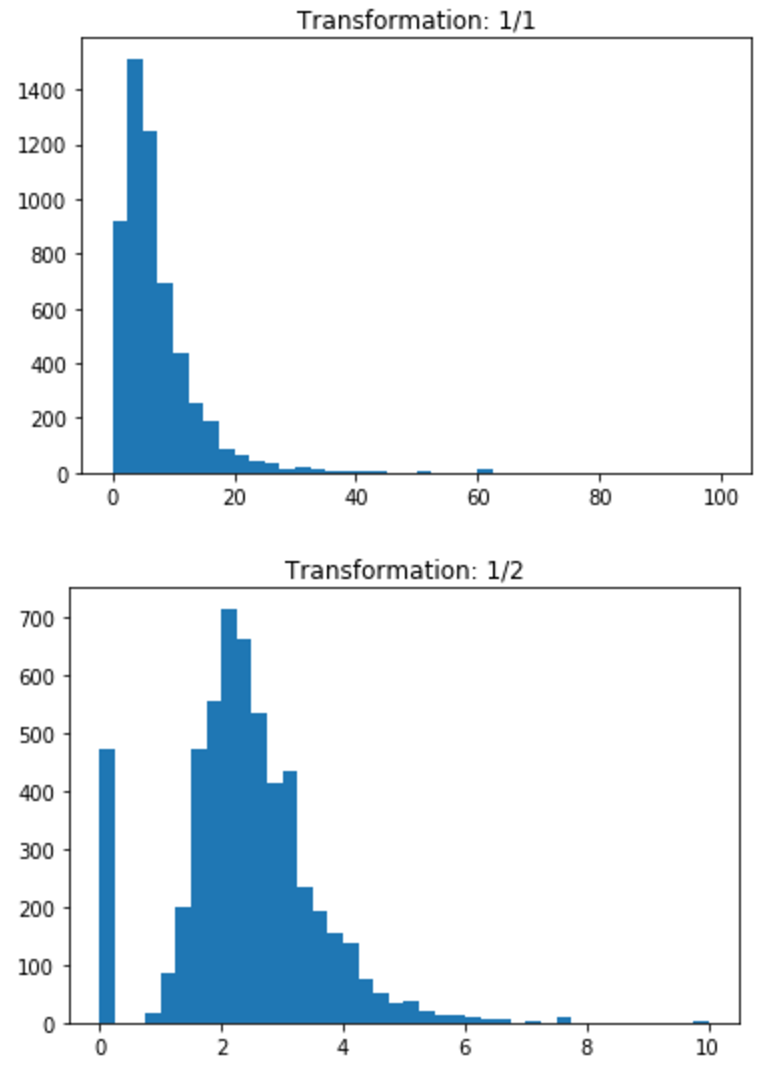
Power 2 means value raised by ½ means square root of it

Etc.

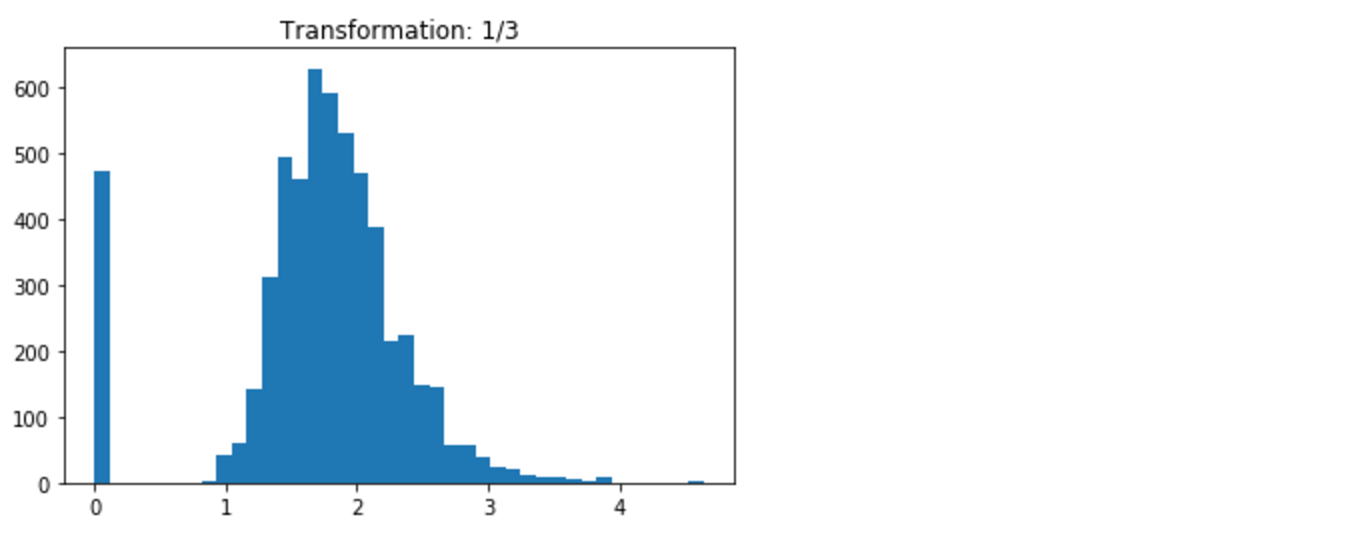


Note that the first graph below is exactly our old graph (“as is,” 1/1, no change)

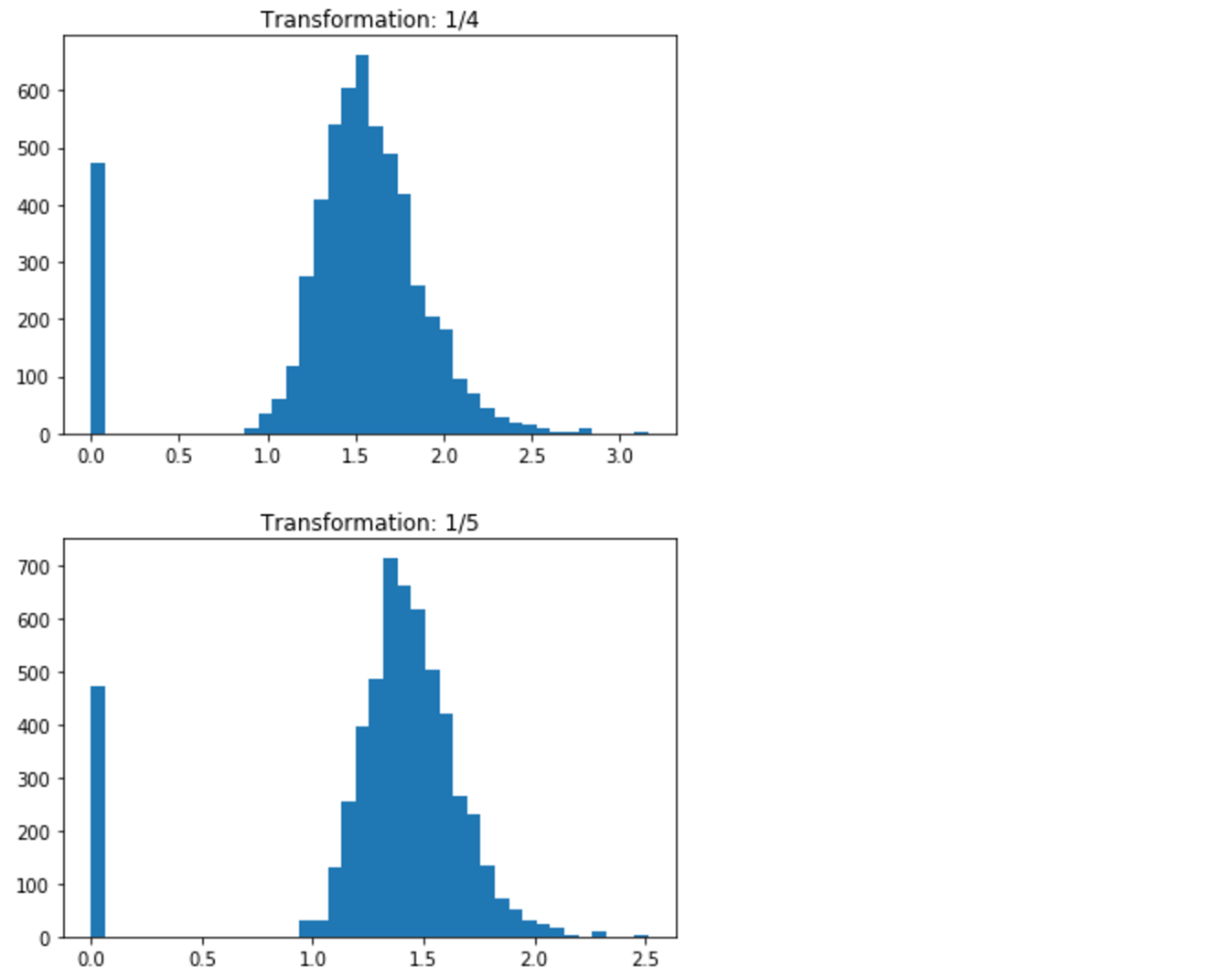
The second graph, ½, started to pull the tail as part of the histogram



The third graph, ⅓, is becoming like a pronounced histogram, more compact, more normal.



The fourth graph, ¼, is even more better, and ⅕, as well.



Given these five distributions, ¼ and ⅕ are the better transformation because it is more of a compact and normal distribution. Both look pretty good. Also, note that you do have this stack to the left and all these are just zeros, so this means that there is no punctuation.

Any power transformation of zero is just going to keep it at zero so we’ll maintain that stack on the left. What we are mostly concerned about is the rest of the distribution seeing how the transformation effects that. These power transformations are a commonly used method to transformation skewed data or data that is not behaving particularly well. It helps your model key in on the data and leverage it to make predictions in a cleaner way.

● All submissions should be separate from other exercises and quests. Please do not lump all your answers into one document and re-using that same workspace to gain multiple points. Thanks.

● Place your name at the bottom of your code, download your Python program in html format, and submit your work in Canvas.