Stat 992 - Research Paper - Grammar of Graphics

Aaron Schram

Abstract

Introduction

The Grammar of Graphics (GoG) can be considered an orthogonal system of seven classes or data flow from which one can form a mathematical foundation for creating statistical graphics (Wilkinson, 2010). The core purpose of this data flow or class system is to design a grammar that can define any plot of any form. For further defining the GoG as a base language for forming statistical graphics, Wilkinson (2010) further describes this decomposition of the grammar on statistical graphics into seven strict categories which are the Variables, Algebra, Scales, Statistics, Geometry, Coordinates, and Aesthetics.

Then, we define Variables as the input of data from a file to a transformation of the data using table joins, filters, selects, and any other data pre-processing tools, which is denoted as the Algebra. From the Algebra, we have our the true data that we are interested in deriving a statistical graphic from, so we move onto the Scale class. Wilkinson (2010) denotes this the Scale class as the least observable or the most ignored class in with regards to the most common of charts. The core idea is that we a preferred scale or measurement for the data. These scales could consist of just an identity transformation on the algebra all the way up to non-linear transformations, but furthermore this class is also associated with the structure of the data. Here, the measurement or data structure just has to do with the class of the data itself, such as numeric, binary, interval, and so on. This truly marks the end of the data pre-processing phase as the remaining classes have to do with how the data will be mapped and plotted.

As for the last 4 classes, Statistics is the last class after all the data has been processed. It has to do with how we want to perceive the data. For example, do we want view every point as in a scatter plot, or do we want to view some kind of summary statistic as in histogram or bar chart. We even define this as output from regression lines or interactions plots from statistical tests and procedures. From the Statistics, the next class is the Geometry which is closely tied to the Statistics, since Geometry is all about how the data is being represented

on the plot itself. Geometry is the part where we specify if we are looking at lines, points, or bars, and how we perceive these on a graph. An easier way to examine this would be through the use of geom functions used in ggplot2 package (v3. 3.3; Wickham, 2016). These represent the visualizations of each layer and the representations of the Statistics (Wickham, 2009). In other words, the Geometry is the plot being created.

Finally, our last two classes are straightforward. Second to last, we have Coordinates, and this is just the coordinate grid system that we are plotting in. Then, we have the Aesthetic class which produces are final visual given all other classes. Wilkinson (2010) breaks the Aesthetic class into seven sub-functions based off of Bertin's *Visual Variables* (1967), and calls them position, size, shape, orientation, brightness, color, and granularity. Therefore, the Aesthetics class can be perceived as everything to do with the visualization of the graphic from the size each point to the color scheme of the graph itself.

Minimizing the Grammar

The Grammar of Graphics introduces a total of seven different orthogonal classes that are fundamental building blocks for constructing statistical graphics, however not all classes are needed to construct graphics, and the construction is not necessarily a straight data flow. It will be argued that the Variable, Algebra, and Scales classes are not necessarily needed. Additionally, classes can be reordered and combined to form alternate classes or even layers. One of the biggest points of my argument comes from Wickham's *A Layered Grammar of Graphics* (2010), where ggplot2 and layering are introduced. Wickham (2010) argues that the DATA, TRANS, and Algebra steps of the original grammar can be fully ignored by leveraging tools within **R** to preform these data processing steps.

Using ggplot2 and **R** as stepping stone for the first argument, the Variable class can be fully ignored. We do not need to graphical software to implement pulling data into the program, since there are several pre-made programs that will pull and store the data. Such as, Standard Query Language functions will make the data usable for later, and they can even perform transformations and data pre-processing operations. This seemly eliminates the Algebra class as well or at least combines the Variable class and Algebra class into one single step. The bottom line or at least on par with Wickham's (2010) reasoning is that both of these steps can be handled and performed by functions or libraries outside the graphics tool. Then, the graphics tool can call upon the data set created.

Now, the Scales is an interesting case as we can perform simple functions to transform our data very easily outside a graphics, but we can also easily implement these transformation within said tool. For instance, in **R**, we could use base functions to scale and transform the data, such as log, exp, or scale (R Core Team, 2023). However, in a function like ggplot2, we use scale arguments to align layers of plots, but these are done after calling the Statistics and the Geometry classes, and these consist of log scales, continuous scales for x and y axes, and color scaling for heat maps (Wickham, 2010). These individual uses of scaling on layers

allows us to realign our generated graphics before we overlay them. On the other hand, we can perform these basic transformations outside of the graphics tools, and the realigning of our layers can be seen as part of the layering step within the Coordinates class in this case or even as part of the Aesthetics class. This is due to the fact that Wickham's (2010) scaling in ggplot2 is placed out of order from the original use of the grammar as specified above.

To compound further on my logic for why Scales should not be a class, consider two different lines of thought, pre-sent data of interest and Generalized Linear Models. We often receive data in a preset form on a preset scale of interest. This by nature invalidates the need for the Algebra and Scales class. For the second case, suppose we have data that will generate statistics on as part of the Statistics class and such data will belong to non-Gaussian right-skewed data. Naturally, we would fit Generalized Linear Model to the data, but this would require transforming the data using a link function to our linear predictor space. This transformations would generate our statistics on the wrong scale of interest, and we would use the inverse link to back transform our data to the correct scale of interest. Here, we performed scaling and transformations within the Statistics class. Therefore, the Scales class has been shown to be within two other classes of interest, and we have shown it to be easy to perform outside of the graphics tool as well.

The Essential Grammar

Previously, we described that the Variable, Algebra, and Scales classes were not entirely needed, but we did not mention the remaining four classes of grammar. This leaves us with Statistics, Geometry, Coordinates, and Aesthetics. Statistics and Geometry are the foundations for forming the statistical graphic as Statistics dictates how we want to display our data and Geometry physically displays the statistics on the graphic. We can think of these two classes as the most rudimentary form of data and the graph type. Then, we have aesthetics which allows us to customize and visualize our statistical graphic. Clearly, we want to be able take our data in a specified form or summary statistic, pick a plot type, and then display it. The classes Statistics, Algebra, and Aesthetics will do this at the bare minimum while retaining the ability to customize our data points positioning, size, and color.

Now, this leaves us with the Coordinate class.

References