When R processes the code and generates output, the R functions format the result based on its type (via the toWordprocessingML() function). The person running the document through R can chose to override some of the methods for handling different types. But it would also be useful for the author of the document to be able to specify the desired formatting of the results for a particular code segment. There are various approaches to support this, each with pros and cons.

One approach is that the author inserts a “template” result and formats that. In many cases, this is desirable as the author will want to see the actual results from the computation they did, not what will be done in the future when the results are re-generated. Then, when we generate a new result, we try to fit the result to this format. This is easier said than done if the format is “free form”. But for matters such as tables where the number of rows and columns remains the same, we can reuse the table and cell formatting. For example, let’s suppose we produce a summary of a data frame that we generate randomly. We want to display this in a table and color the alternating cells.

The only difficulty is that we have to instruct the R processor to use adjacent tables. This can be done with an argument to wordDynDoc() but we would also like it to be feasible to do this within the document. We want the author to indicate that a particular element is a template.

1. Perhaps the simplest, most deterministic convention is that if a paragraph of style R code or R plot (or its descendants) is followed by a paragraph descending from Routput or R output table, then we use the settings from that following R output paragraph as a “template” for the output from R. Specifically, we endeavor to “squeeze” the R object into this template where it makes sense. So we try to make it make sense.
   1. We interpret following pictures as providing the desired format (e.g. PDF, JPEG, PNG, …), width and height, etc. for the plot we produce.
   2. If the trailing element is a table, we try to put the elements of the R object into the table, using the recycling rules of R.
   3. And however else we can insert an R object into an element.
2. We might do this with a comment on the R code to indicate the template. The precise format or style of the comment needs to be determined, but we can have R code within it,
3. We might assume that if a table immediately follows an R code segment, that we insert into that. Similarly for a plot.
4. We could put an empty run with a particular style just after the R code or before the next element. This is a bit less desirable as we cannot see it and also if it is after the R code, it would have to be a paragraph.