Duncan Temple Lang, University of California at Davis

Table of Contents

Using RWordXML for Authoring Dynamic Documents with R Code

We have defined several styles that you use to markup R and shell code as you are authoring a document in Word. To access these styles, you can either open our template (RStyles.dotm) that is installed with the package, or you can chose to use that template as the default for all Word documents (or merge its styles into the default template). Either works fine.

The simple idea is that you write your document, cut-and-paste or write R code and mark it up with the appropriate styles that identify the role of the code, e.g. an R plot, regular R code, shell code.

When you want to run the code and insert the results, you use the RWordXML package and specifically the *wordDynDoc()* to process the document, run the code and insert the relevant output into a *new* Word document (i.e. we don't overwrite the original one!).

Installing the Templates with Styles

We'll illustrate this for a Mac.

There are two ways we can use the R styles as defined in inst/WordStyles/RStyles.dotm. One is to simply open that file and then write our document. The second is to define that template file as the global/default template document.

The first approach involves simply opening the RStyles.dotm file in Word. On a Mac, this is

```
open inst/WordStyles/RStyles.dotm
```

We can use the template with R styles as the global template by copying it to the location that Word finds the default template. On a Mac, this is /Users/duncan/Library/Application Support/Microsoft/Office/User Templates/Normal.dotm. Before we copy our new version there, you might make a copy of what is currently in that file. This is important if the settings there are different from the default settings. So we might do the following:

```
cp '/Users/duncan/Library/Application Support/Microsoft/Office/User Templates/Norm
   '/Users/duncan/Library/Application Support/Microsoft/Office/User Templates/Origi
```

Now we can copy our template into the default location:

```
cp inst/WordStyles/RStyles.dotm '/Users/duncan/Library/Application Support/Microso
```

If you want to reverse this, you can delete the newly created file and Word will create a new default based on the standard settings. If the original .dotm file was different from these settings, you should copy it back

```
cp '/Users/duncan/Library/Application Support/Microsoft/Office/User Templates/Orig
    '/Users/duncan/Library/Application Support/Microsoft/Office/User Templates/Nor
```

On a Mac, The Templates and Add-ins menu item under Tools can be used to manage the default template and its contents (under Organize).

The Styles

Firstly, the RStyles.dotm is a work in progress. I welcome any enhancements to it and people should define their own styles for code-related content. They can share them by "publishing" their .dotm file.

There are several styles defined in this template. They are described in the following table.

Table 1.

Style name	inline or paragraph	Purpose	Short cut
R options	paragraph		
Routput	paragraph		
R output (no border)	paragraph		
R output table	paragraph		
R plot output	paragraph		
Rcode	paragraph	a paragraph	
Rexpr	inline	an R expression in the middle of text, e.g. `length(x)`	
Rfunc	inline	a reference to an R function by name	
Rplot	paragraph	R code that creates a new plot. This is a paragraph style and text should be split across lines using Shift-Return rather than starting a new	Command+P
R Session Info	paragraph	a style for presenting information describing an R session via *sessionInfo*(). When the document is processed, the session info is added at the bottom of the document using this style.	

Note that we can compute these since the .dotm file is like a regular Word (.docx) file in structure, and the styles defined as XML.

```
library(RWordXML)
doc = wordDoc("RStyles.dotm")
sty = doc[["styles.xml"]]
```
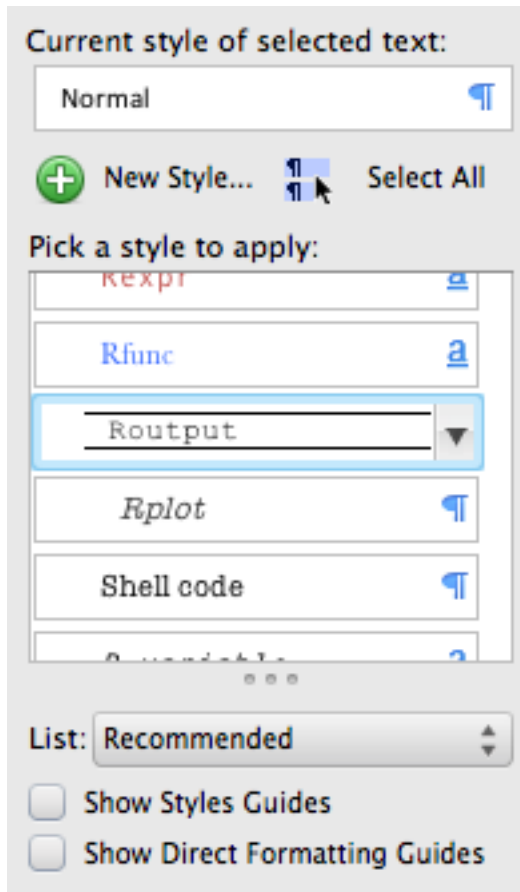
The names of the styles are

```
xpathSApply(sty, "//w:style/w:name", xmlGetAttr, "val", namespaces = "w")
```

Using the Styles

Adding New Styles

It is easy to add new styles or alter existing styles. Use the View menu and under the Toolbox section, select Styles. This will bring up an additional window which manages the styles.

You can click the "New Style" button to start creating a new style. That will bring up a dialog that looks like

. You provide a new name, chose whether the style is for a paragraph or character/inline within text. You can chose a style as the template and then customize that, or just use the default. For a paragraph style, it is often convenient to have the "Style for following paragraph" be Normal so that when the author is writing and hits return, she doesn't have to explicitly change the style for the new paragraph.

If you want this style to be defined outside of this single document, you must remember to check the "Add to template" near the bottom of the dialog.

You can also set many additional characteristics of this style via the menu item that reads "Format". That has options for Font, Tabs, Border, etc. It also allows you to specify a shortcut key sequence for applying this style. This can be very convenient for commonly used styles.