

# Improving PDF Accessibility Using R

*S. & R. Schwinn*

*4/21/2018*

This guide describes an approach for improving the accessibility of documents generated in L<sup>A</sup>T<sub>E</sub>X. The first step involves *knitting* an RMarkdown file, however adding the following commands to the header of your standard L<sup>A</sup>T<sub>E</sub>X document will allow you to follow along from step two.

```
\RequirePackage{accsupp}
\RequirePackage{pdfcomment}
\newcommand{\AccTool}[2]{\BeginAccSupp[method=pdfstringdef,unicode,Alt={{#1}}}\pdftooltip{{#2}}{{#1}}\E
```

## Step 1: Generate T<sub>E</sub>X Output Using RMarkdown

Click this text to access the example Rmd file, or download all of the content by clicking here. The first difference between this file and a plain Rmd is the YAML. The **fontsize** command does what you might expect. The **keep\_tex** command ensures we have access to the intermediate files for step two. Finally, the **header\_includes** command adds required L<sup>A</sup>T<sub>E</sub>Xcode to the header.

```
fontsize: 11pt
output:
  pdf_document:
    keep_tex: yes
header-includes:
  - \RequirePackage{accsupp}
  - \RequirePackage{pdfcomment}
  - \newcommand{\AccTool}[2]{\BeginAccSupp[method=pdfstringdef,unicode,Alt={{#1}}}\pdftooltip{{#2}}{{#1}}
```

## Step 2: Make Modifications to the T<sub>E</sub>X

This file will modify the T<sub>E</sub>X output from step one. It uses R's **gsub** command to add tooltips and accessibility tags to the document. One can design clever regular expressions to capture caption text for the accessibility tags. I used a ~100-line variation of this file to add more than 1,000 tags to this 218-page document.

The last line of the script uses the **texi2pdf** command to generate a PDF file.

## Step 3: Run Adobe Acrobat Scripts

The final step requires Adobe Acrobat Pro. Open the PDF file generated in step two and perform the accessibility actions defined in this Acrobat script. You can install the scripts by simply opening the accessibility.sequ file if Acrobat is installed. For those managing multiple documents, Acrobat offers simple batch processing of the accessibility.sequ commands.

## Results

After performing all three steps, your document will be far closer to 508 compliance than the original *knitr* output. Among the four issues remaining after processing, **logical reading order** and **color contrast** always require manual judgement, **Title** is fixed with two clicks, and the final **Headings-Nesting** rule can be skipped. Notice the figure's alternative text tooltip.

