R Markdown Output Type Test

R. M. D. Trickster

2022-03-12 09:29

Contents

1 Output format testing

This document provides an example of using a single Rmd file to generate output with different content based on the output document type.

The envelope-pushing examples below are complicated, and in many cases require generating strings that are rendered using the structure:

The examples are unlikely–it is hard to imagine a case for which a single Rmd file should generate completely different output. But the examples should serve to demonstrate how different output can be included based on the type of output file.

If this is an HTML document, the following apply:

1. Text below indicates "This is HTML output and shows a Leaflet map."
2. A Leaflet map is shown, along with a caption and cross-reference
3. Code chunks are displayed inline, "hidden" by default.
4. There is no source code at the end of the document.

If this is PDF output, the following apply:

1. Text below indicates "This is PDF output and therefore a Leaflet map cannot be added."
2. The Poisson distribution equation is shown, with an equation number and cross reference.
3. Code chunks are not shown inline.
4. Source code is printed at the end of the document.

This is PDF output and therefore a Leaflet map cannot be added.

See the Poisson distribution (1) for details

$$P(x) = \frac{e^{-\lambda}\lambda^x}{x!} \tag{1}$$

2 Source code

```
---
title: "R Markdown Output Type Test"
author: "R. M. D. Trickster"
date: '`r format(Sys.time(), "%Y-%m-%d %H:%M")`'
header-includes: #allows you to add in your own Latex packages
```

~~~
- \usepackage{float} #use the 'float' package
- \floatplacement{figure}{H} #make every figure with caption = h
output:
bookdown::pdf_document2:
number_sections: true
toc: true
fig_cap: yes
keep_tex: yes
bookdown::html_document2:
number_sections: true
self_contained: true
code_folding: hide
toc: true
toc_float:
collapsed: true
smooth_scroll: false
---

```{r setup, include=FALSE}
library(sf)
library(leaflet)
library(kableExtra)
library(magrittr)
library(knitr)
library(tidyverse)
library(captioner)

# captions
eqn_captions <- captioner::captioner(prefix="")
fig_captions <- captioner::captioner(prefix="Figure")


# test for output type; if HTML, include code; if PDF do not include code
if(is_html_output()){
knitr::opts_chunk$set(echo = TRUE)
} else {
knitr::opts_chunk$set(echo = FALSE)
}

# path to this file name
if (!interactive()) {
fnamepath <- current_input(dir = TRUE)
fnamestr <- paste0(Sys.getenv("COMPUTERNAME"), ": ", fnamepath)
} else {
fnamepath <- ""
}
```

# Output format testing
This document provides an example of using a single Rmd file to generate output
    with different content based on the output document type.

The envelope-pushing examples below are complicated, and in many cases require
    generating strings that are rendered using the structure:
~~~

~~~
```{md}
if(is_html_output){
cat("
some R Markdown syntax
")
}
```

The examples are unlikely--it is hard to imagine a case for which a single Rmd
    file should generate completely different output. But the examples should
    serve to demonstrate how different output can be included based on the type
    of output file.

If this is an HTML document, the following apply:

1. Text below indicates "This is HTML output and shows a Leaflet map."
1. A Leaflet map is shown, along with a caption and cross-reference
1. Code chunks are displayed inline, "hidden" by default.
1. There is no source code at the end of the document.

If this is PDF output, the following apply:

1. Text below indicates "This is PDF output and therefore a Leaflet map cannot
    be added."
1. The Poisson distribution equation is shown, with an equation number and cross
    reference.
1. Code chunks are not shown inline.
1. Source code is printed at the end of the document.

```{r, results='asis'}
if(is_html_output()){
foo <- ""
cat("Because this is HTML output, a Leaflet map could be included!\n\n")
cat("## A Leaflet Map\n")
}
```

```{r}
if(is_html_output()){
# the Space Needle
snxy <- data.frame(name = "Space Needle", x = -122.3493, y = 47.6205)
space_needle <- st_as_sf(snxy, coords = c("x", "y"), crs = 4326)

# a leaflet map
m <- leaflet() %>%
addTiles() %>%
addCircleMarkers(data = space_needle)
m
}
```

```{r, results='asis'}
if(is_html_output()){
~~~

~~~
# make and print the caption
# this creates a string object from the citation, and drops all white space
cap <- fig_captions(name = "Leaflet", caption = "A Leaflet map centered at the
    Space Needle\n\n")
cat(cap)

cite <- fig_captions(name = "Leaflet", display = "cite")

# This adds the text
cat(sprintf("See the Leaflet map (%s) for details.", cite))
}
```

```{r, results='asis'}
# if this is LaTeX output then print the Poisson distribution and use an
    equation number and cross-reference
if(is_latex_output()){
cat("This is PDF output and therefore a Leaflet map cannot be added.\n\n")

# an equation caption
eqn_captions(name = "poisson", caption = "")

# this creates a string object from the caption, and drops all white space
cite <- eqn_captions(name = "poisson", display = "cite") %>%
str_remove_all(pattern = "\\s")

# This adds the text
cat(sprintf("See the Poisson distribution (%s) for details", cite))

# `cat' to print out an equation
cat("
\\begin{equation}
P(x) = \\frac{ e^{ - \\lambda } \\lambda ^x }{ x! }\
\\end{equation}
")
}
```

```{r results='asis'}
# only print the source code if this is LaTeX output
if(is_latex_output()){
cat("# Source code\n\n")
}
```

```{r comment='', }
# only print the source code if this is LaTeX output
if(is_latex_output()){
srccode <- readLines(fnamepath) %>%
str_wrap(width = 80, exdent = 4)
cat(srccode, sep = '\n')
}
```
~~~