Working with Code environments in texor

by Abhishek Ulayil

Abstract This is a small sample article to demonstrate usage of texor to convert code environments.

1 Introduction

Pandoc naturally converts verbatim environment easily, however the redefination of other commands such as example, example*, Sinput etc to verbatim does not work well in pandoc.

Hence, the **texor** package uses a stream editor to search find and replace matching code environments to verbatim before pandoc touches it.

This way the the code is not lost in conversion, also a pandoc extension is used to add attributes to the markdown code using `fenced_code_attributes`

Code Environment types are well summarized in the table 1

Code Environment Type					
Example	example	example*			
S.series	Sin	Sout	Sinput	Soutput	Scode
Special Verbatim	boxedverbatim				

Table 1: Code Environment support in texor

2 Environments

Verbatim Series

While verbatim is naturally supported in pandoc,other extensions of verbatim environment like boxedverbatim from moreverb package (Fairbairns, Duggan, Schöpf, Eijkhout , 2011) falls back to normal verbatim.

1. verbatim :

```
print("Hello world")
```

2. boxedverbatim :
```
print("Hello world")
```

S series

S series code environement is defined in Rjournal.sty file. Most of these are extensions of verbatim environment and retain the original formatting of the content in HTML.

1. Sinput :

```
print("Hello world")
```

2. Soutput :

```
[1] "hello world"
```

3. Sin :

```
print("Hello world")
```

4. Sout :

```
[1] "hello world"
```

Example series

Example series of code environment is defined in Rjournal.sty file. Examples are extensions of verbatim environment and retain the original formatting of the content in HTML.

1. example :

```
print("Hello world")
```

2. example* :

```
print("Hello world")
```

3 Code in Figure Environments

A small example of this is visble in 1. This is a common practice in Rnews articles as it used to add a boxed border around the code which looks attractive. However, in web articles there isn't much advantage to it.

```
code_in_figure <- function() {
  if (pandoc_version >= 3) {
    print("Code in Figure Supported")
  }
  else {
    print("code in Figure not supported")
  }
}
```

Figure 1: Example Code inside Figure environment

Pandoc v3 or greater (Krewinkel, Lucero , 2023) has a Figure object which allows non-image figures to be treated like one. This is why **texor** package requires atleast version 3 of pandoc.

4 Code in Table Environments

We can use code environments in a table using minipage environments. This is not a common practice among LaTeX article authors, but a few articles had such complex structures. So, as a example to demonstrate pandoc and **texor** package's capabilities, I have included a few of them.

Table 2 is an example of code environments within a table.

Language	Function Defination Syntax

R	```fun <- function(){
	  print("A function in R")
	  return(0)
	}```

Python	```def fun(): print("A function in Python")```
Lua	```function fun() print("A function in Lua") end```

Table 2: Code in a table

A similar arrangement can be had for figures/plots besides code environment. Table 3 demonstrates a table with code and figure.
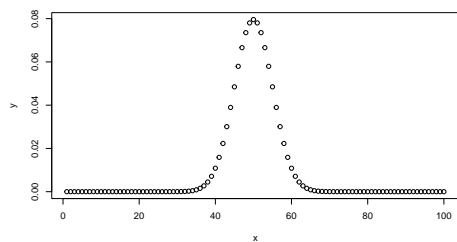
Code	Plot
| ```
x <- 1:100
y <- dbinom(x,100,prob = 0.5)
plot(x,y)
``` |  |

Table 3: Code and Plot side by side

5 Inline Code usage

Using inline code in LaTeX is possible using \verb command. It would be repoduced similarly, as a Inline code element.

```
\verb|x <- 1:100|
```

will be represented as x <- 1:100 in Inline format.

6 Code chunks using Schunk

Code chunks within an Schunk environment to demonstrate Input/Output

Input :

```
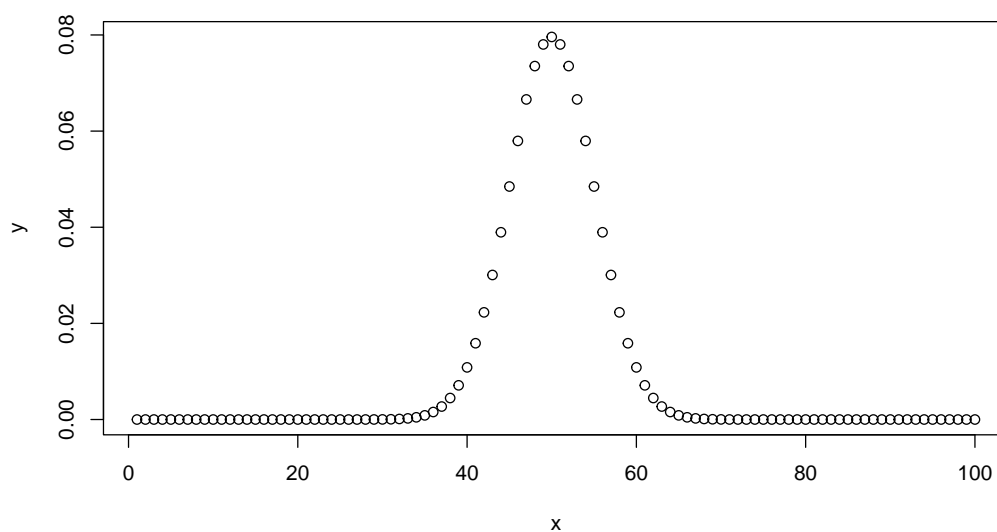print("Hello world")
```

Output :

```
[1] "hello world"
```

Similar arrangement can be had for Plots as well using figure environnment.

Input :

```
x <- 1:100
y <- dbinom(x,100,prob = 0.5)
plot(x,y)
```

Output :

7 Including 'R code chunks'

Including R code chunks in an article that is to be converted to R markdown can be a bit tricky as the **knitr** package will interpret and execute the R code chunks as actual code bits. A workaround for this issue has been described in this posit support article. Below is an example with the workaround adapted from Stephens (2022).

```
`r ''`````{r}
plot(cars)
```
```

8 Summary

In summary the **texor** package supports:

- Almost all code environments in RJournal.
- Code Highlight for R language.
- Inline Code.
- Code in different environments like tables/figures.

Bibliography

A. Krewinkel and A. Lucero pandoc 3.0 Release notes *pandoc* 2023 URL https://pandoc.org/releases.html [p2]

R. Fairbairns, A. Duggan, R. Schöpf and V. Eijkhout The moreverb package documentation *CTAN* 2011 URL https://mirror.niser.ac.in/ctan/macros/latex/contrib/moreverb/moreverb.pdf [p1]

Nathan Stephens Including verbatim R code chunks inside R Markdown *Posit support forum*, 2022 URL https://support.posit.co/hc/en-us/articles/360018181633-Including-verbatim-R-code-chunks-inside-R-Markdown

[p4]

Abhishek Ulayil
Student, Institute of Actuaries of India
Mumbai, India
ORCiD: 0009-0000-6935-8690