

concordance=TRUE

Programación de Aplicaciones para Internet y la Nube

2016-I

TIPOS DE FORMATOS DE DOCUMENTOS REALIZADOS EN KNITR

Nombre del Estudiante: **Gabriel Andrés Alzate Acuña**

El paquete **knitr** usado dentro del ambiente de trabajo R brinda opciones para integrar L^AT_EX al desarrollo normal de un proyecto basado en R generando documentos científicos de investigación reproducible bajo el paradigma de programación literaria, convirtiéndose en una herramienta de alto valor para el científico en la generación de sus reportes de investigación, el presente documento relaciona los diferentes tipos de formato que pueden ser generados mediante el paquete knitr, y la sintaxis general de trabajo de las mismas, dando una revisión de la forma de realizarlo y los alcances del mismo.

1 Sintaxis de entrada

Los bloques de código en R que se insertan dentro del documento son llamados Chunks y su sintaxis se ubica en la subsección all patterns, por ejemplo, el patron de comienzo de un chunk es:

```
all_patterns$rnw$chunk.begin
## [1] "^\\s*<<(\\.&)*>>=\\.\\$"
```

Los caracteres para iniciar un chunk son `<< Alguntexto >>=` recordando que debes ser utilizados al principio de la línea de texto.

1.1 Opciones de chunk

El encabezado del chunk permite insertar opciones, como se menciona anteriormente de la forma

`opcion=valor`

Para demostrar esto se definirá en un chunk la variable *bar* que inicialmente tiene un valor de 3.

```
bar = 2
```

Ya definida la variable se crea un chunk que evalúe esta variable y según su valor decida si ejecuta la acción correspondiente con la siguiente sintaxis:

```
$ <<eval=if (bar < 5) TRUE else FALSE>>=\\.\\$
```

La expresión se evalúa si *bar* < 5, de la siguiente forma:

```
<< eval = if(bar < 5)TRUEelseFALSE >>=
bar
@
```

Y da como resultado:

```
bar
## [1] 2
```

Ahora se procede a cambiar el valor de *bar* a 7:

```
bar = 7
```

Y procedamos a ejecutar de nuevo la sentencia:

```
$<<eval=if (bar < 5) TRUE else FALSE>>=$.
```

Dando el siguiente resultado:

```
bar
```

Mostrando que no se ejecuta la presentación de la variable pues es mayor a 5, gracias a usar la función `eval` dentro de el encabezado del chunk.

1.2 Etiquetas de Chunk

Estrictamente las etiquetas de chunk son parte de las opciones de chunk, pero se hace hincapié en ellas pues son las únicas partes del documento que no deben tener la semántica de R, las opciones para la etiqueta válidas pueden ser:

- `<< foo >>=`
- `<< foo - bar >>=`
- `<< foo_bar >>=`
- `<< "foo" >>=`
- `<<' foo - bar' >>=`
- `<< label = "foo" >>=`
- `<< echo = FALSE, label = "foo - bar" >>=`

Las etiquetas son únicas en el documento, pues generan archivos externos como imagen y archivos temporales, así que no pueden ser llamadas dos de la misma forma pues Knitr generara un mensaje de error para no sobrescribir alguna ya usada.

1.3 Sintaxis de Chunk

En el paradigma de programación literaria la sintaxis que se usa para denotar el inicio de código computable es `<<>>=`, y usamos el carácter `@` para denotar el comienzo de la documentación, esto debido a que tradicionalmente en los reportes el código chunk aparece en menor medida que el texto normal y la realización se hace intuitiva, siendo de la siguiente manera:

```
<<>>=
Texto del documento
1+1
<<>>=
rnorm(10)
@
Mas texto del documento
```

2 Formatos de documento

Hasta el momento se ha presentado la sintaxis de Rnw como ejemplo, en la tabla 1 se presentara la forma de usar estas características en otros formatos de documento.

Formato	Inicio	Final	Línea
Rnw	<< * >>=	@	Sexpr{}
Rmd	“{r*}	”	‘r x’
Rhtml	<!--begin.rcode *	end.rcode -->	<!--rinline x -->
Rrst	.. {r *}	:r:x’
Rtex	% begin.rcode *	% end.rcode	rinline {x }
Rasciidoc	// begin.rcode *	// end.rcode	‘r x ‘
Rtextile	# # # . begin.rcode *	# # # . end.rcode	@r x@
brew			< %x% >

Table 1: Resumen de los formatos de documentos que genera Knitr.

2.1 Descripción de los formatos

Markdown permite generar documentos fáciles de escribir y fáciles de leer como texto plano, generalmente se validan como XHTML o HTML, este fue diseñado para ser simple y entre sus fortalezas está la fácil escritura de tablas, expresiones matemáticas o bibliografía.

Markdown fue diseñado en primera instancia para la web, para trabajos más complicados de escritura científica se prefiere \LaTeX , el cual ha sido el formato de trabajo del presente documento.

HTML (Hiper-Text Markup Language) es usado en publicaciones web, y es bastante usual encontrarlo en publicaciones científicas, además su presentación es bastante agradable a la vista; el precio a pagar es que toma mucho más tiempo el desarrollo de un documento.

El formato reStructuredText (reSt) es similar a Markdown pero con más prestaciones (y más complicado), permite la adición de fórmulas y párrafos científicos de forma más robusta. AsciiDoc es un formato de texto plano que permite la inmediata conversión en múltiples salidas, como documentación, artículos, libros y páginas HTML.

Textile es otro lenguaje de marcado ligero de un uso no tan extendido como los anteriores pero que soporta las mismas características, es implementado en knitr debido a que es solicitado en algunas revistas que publican reportes científicos.

2.2 Personalización

Es posible definir nuestra propia sintaxis para un documento ya que knitr usa el objeto *knitpatterns* para manejar las expresiones regulares, para especificar nuestra propia sintaxis podemos usar la función:

```
knit_patterns$get(
  c("chunk.begin", "chunk.end", "inline.code")
)

## $chunk.begin
## [1] "^\\s*<<(.*)>>=.*$"
##
## $chunk.end
## [1] "^\\s*@\\s*(%+.*|)$"
##
## $inline.code
## [1] "\\Sexpr\\{([~}]+)\\}"
```

Dándonos la oportunidad de redefinir estos con el vector *c* propuesto en el ejemplo.

3 Conclusiones

Las posibilidades de realización de documentos dinámicos brindan una excelente alternativa al momento de realizar artículos de investigación debido a la forma en que se organizan las ideas con el soporte matemático correspondiente y la visualización interactiva que presentan este tipo de herramientas.

Knitr es una herramienta flexible que permite la integración de una amplia cantidad de opciones al momento de trabajar diferentes formatos de salida, sea para desarrollo web, presentación y visualización dinamica o presentar documentos de una forma organizada y clara.

El uso de knitr implica tener cierto grado de conocimiento de \LaTeX esto debido a que el paquete no trae consigo una opción de autocompletado de sintaxis del mismo, solo el propio de la interfaz R trabajada, así pues si nos e conoce la sintaxis es bastante engorroso el trabajo de la herramienta.