

Informe de lectura: Dynamic Documents with R and knitr : Cross Reference

Wilson Ricardo López Sánchez

Universidad Distrital Francisco José de Caldas

June 3, 2016



Para ejecutar código R en un documento. Rnw es necesario utilizar la sintaxis:

```
<<>> =  
codigo R  
@
```

Para mayor facilidad podemos usar la combinación de teclas *control+alt+i* para generar la sintaxis anterior.

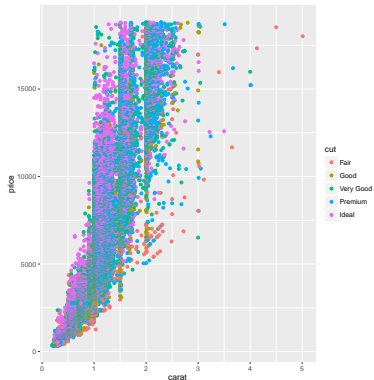
Después de instalar el paquete qqplot2 podemos cargar la librería y el siguiente ejemplo:

```
<<>> =  
library('ggplot2')  
qqplot(carat, price, data = diamonds, color = cut)  
@
```

```
library('ggplot2')
```

```
## Warning: package 'ggplot2' was built under R version  
3.2.5
```

```
qplot(carat, price, data = diamonds, color = cut)
```



Ejemplo en Gráficas tema

```
<<temagraficas1, eval=FALSE>> =  
theme(legend.text = element_text(size = 12, angle = 45)) +  
theme(legend.position = "bottom")  
@
```

Ahora podemos realizar nuestra gráfica de la siguiente manera:

```
<<>> =  
qplot(carat, price, data = diamonds, color = cut) +  
<<temagraficas1>>  
@
```

Referencias por segmentos embebidas

```
<<A>> =
```

```
x <- runif(1,0,10)
```

```
@
```

```
<<B>> =
```

```
x
```

```
<<A>>
```

```
x
```

```
@
```

Con lo que obtenemos:

```
x <- runif(1,0,10)
```

```
x <- runif(1,0,10)
```



Re-utilización de todo un segmento de código

```
<<chunkA, eval=FALSE>> =
```

```
x <- 5*6
```

```
@
```

```
<<chunkA, eval=TRUE>> =
```

```
@
```

Lo que da como resultado:

```
x <- 5*6
```

```
x <- 5*6
```

Re-utilización de todo un segmento de código

```
<<A1>> =  
x <- rnorm(1) @  
  cdigo  B1  
<<B1>> =  
y <- x + 2 @  
<<C, ref.label=c('A1', 'B1')>> = @
```

Lo que da como resultado:código A1

```
x <- rnorm(1)
```

Código B1

```
y <- x + 2
```

Código C

```
x <- rnorm(1)  
y <- x + 2
```



Segmentos por etiqueta

Se toma como ejemplo el siguiente código de R el cual guardaremos con el nombre de ejemplo1.R

```
\#\# ----- D -----  
m <- 2  
n <- 5  
m  
n
```

Para llamar el código de R en nuestro documento lo primero es ejecutar el script read_chunk de la siguiente manera:

```
<<>> =  
read\_chunk(" ejemplo1.R")  
@
```

Posteriormente ya podemos llamar el código por medio de su etiqueta

```
<<D>> =  
@
```



Segmentos de códigos basados en líneas

Como ejemplo guardamos los numero del 1 al 12 en un archivo con el nombre ejemplo2.R

1 2 3 4 5 6 7 8 9 10 11 12

Vamos a tomar la línea 1-4,5-8 y 11-12 del archivo de R anterior con ellos formaremos los segmentos de código con las etiquetas E, F y G, dicho proceso se realiza de la siguiente manera:

```
read\_chunk("ejemplo2.R", labels = c("E", "F", "G"), from = c(1,5,  
11), to = c(4,8,12))
```

Así podemos llamar cualquiera de los tres segmentos de código en cualquier momento:

```
<<E>> =
```

```
@
```

```
<<F>> =
```

```
@
```

```
<<G>> =
```

```
@
```



Como ejemplo se toma como documento principal este mismo documento llamado *referenciascruzadas.Rnw* y como documento hijo crearemos un documento llamado *cap1.Rnw* con el siguiente código:

```
<< H1 >>=
```

```
y < -sum(x2)
```

```
@
```

Incluimos el documento hijo de la siguiente forma:

```
<< H, child = ' cap1.Rnw' >>=
```

```
@
```

Modo independiente

Tomando el documento cap1.Rnw y agregamos el siguiente comando al documento hijo:

```
<< parent, include = FALSE >>=  
set_parent("referenciascruzadas.Rnw")  
@
```

Con lo cual el nuevo documento hijo nos quedaría de la siguiente forma:

```
<< parent, include = FALSE >>=  
set_parent("referenciascruzadas.Rnw")  
@  
<< H1 >>=  
y <  $-sum(x^2)$   
@
```

Al compilar el documento hijo observamos que compila sin problemas y con las mismas características del documento principal.



Codigo R en latex

Yihui Xie, 2013

Dynamic Documents with R and knitr

CRC Press 2016. 2 Ed.