

# Gráficas en documentos dinámicos con Knitr

Johan Nicolás Cuellar Salinas  
Cod. 20152195007

3 de junio de 2016

## Introducción

Una de las herramientas más importantes en el análisis de datos son las gráficas. Estas ayudan a entender los datos recogidos, a hacer inferencias sobre las relaciones entre estos y permiten validar las conclusiones a las que se llegan luego de finalizar el estudio. Debido a ello, las gráficas son una parte vital de los documentos de análisis de datos, en los que una sola gráfica podría incluso resumir un estudio completo.

En general, los documentos de análisis de datos se presentan como artículos científicos generados en  $\text{\LaTeX}$  o documentos web, comúnmente HTML. Esto, sumado a la gran cantidad de imágenes que se pueden necesitar en un estudio y lo engorroso que resulta el proceso de creación de las imágenes, añadir las al documento, modificarlas y/o eliminarlas dentro de entornos como  $\text{\LaTeX}$  y HTML, muestran la gran utilidad de la generación dinámica de documentos, en donde se minimizan los tiempos necesarios para dicho proceso y a la vez se hace más transparente para el usuario.

En este artículo se presenta una guía para la generación dinámica de gráficas en documentos  $\text{\LaTeX}$ (pdf) y web (HTML) utilizando R, Rstudio y el paquete Knitr. Esta guía está basada en el capítulo 7 del libro *Dynamic Documents with R and knitr*.

## Chunks

Los Chunks (segmentos de código de R) necesarios para generar imágenes con Knitr son muy similares a los usados normalmente para cualquier cálculo, la principal diferencia es que estos Chunks guardan las imágenes generadas en la carpeta definida por el usuario, para que puedan ser usadas por el explorador (HTML) o el compilador ( $\text{\LaTeX}$ ). Es decir, en el resultado final se puede ver la gráfica y el código que la genera, lo cual ayuda al análisis y la reproducibilidad del estudio realizado en el documento. Un ejemplo del resultado final se puede apreciar en la Figura 1.

```

par(mar = rep(2, 4))
library(ggplot2)
pie <- ggplot(diamonds, aes(x = factor(1), fill = cut))+
  ylab("CUT") + xlab("CUT") +geom_bar(width = 2)+
  ggtitle("Calidad del Corte")

pie + coord_polar(theta = "y")

```

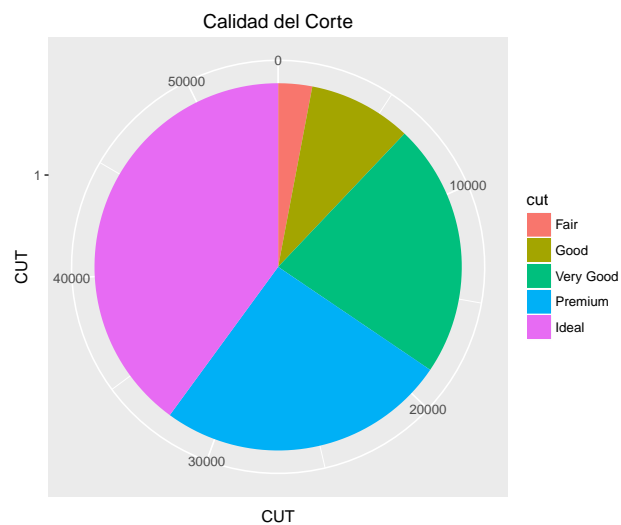


Figura 1: Ejemplo del resultado obtenido usando knitr en la generación de las gráficas. Arriba el código usado, abajo la imagen. Para el ejemplo se usó el dataset *Diamonds* del paquete *ggplot2* de R

## Opciones del Chunk

Se pueden configurar los chunks de las gráficas usando ciertos comandos en la etiqueta al inicio del chunk, con estos comandos se puede asignar características diferentes a las gráficas según las preferencias del usuario. Una lista de los comandos de configuración más comunes se muestra a continuación:

- `fig.align`: Permite al usuario decidir la alineación de la gráfica, 'default', 'left', 'center' y 'right' son los posibles valores.
- `fig.width`: Define el ancho de la imagen generada.
- `fig.height`: Define el alto de la imagen generada.
- `out.width`: Aunque `fig.width` define el ancho de la imagen al ser generada, este puede no corresponder con el ancho de la gráfica en el documento. Para solucionar esto se puede usar la opción `out.width`. Un valor comúnmente asignado a esta opción es ' $a \backslash \backslash linewidth$ ' donde  $a$  es una constante entre 0 y 1, este método permite al usuario definir el ancho de la gráfica como una proporción del ancho de la línea en el documento.
- `out.height`: Aunque `fig.height` define la altura de la imagen al ser generada, esta puede no corresponder con el ancho de la gráfica en el documento. Para solucionar esto se puede usar la opción `out.height`.
- `out.extra`: Esta opción permite la modificación de varias características de la figura, la más usada es el ángulo. Por ejemplo, si se quiere rotar una gráfica noventa grados se debe usar el valor '`angle=90`'.

En la Figura 2 se muestra a manera de ejemplo una gráfica configurada de la siguiente forma: `fig.align='right'`, `out.width='0.5 \backslash linewidth'`, `out.extra='angle=45'`.

## Devices

En R, los devices son los métodos para guardar imágenes, es decir el formato. En los archivos Rwd (Latex) el device por defecto es PDF y para Rmd/Rhtml/Rrst es PNG. Los Devices más utilizados son:

- Del paquete `grDevices`, generalmente preinstalado en R
  - pdf:
  - xfig
  - bitmap
  - png
  - jpeg
  - bmp
  - tiff

```
library(ggplot2)

mtcars$gear <- factor(mtcars$gear, levels=c(3,4,5),
  labels=c("3gears", "4gears", "5gears"))
mtcars$am <- factor(mtcars$am, levels=c(0,1),
  labels=c("Automatic", "Manual"))
mtcars$cyl <- factor(mtcars$cyl, levels=c(4,6,8),
  labels=c("4cyl", "6cyl", "8cyl"))

qplot(mpg, data=mtcars, geom="density", fill=gear, alpha=I(.5),
  main="Distribucin de rendimiento de gasolina", xlab="Millas por Galn",
  ylab="Densidad")
```

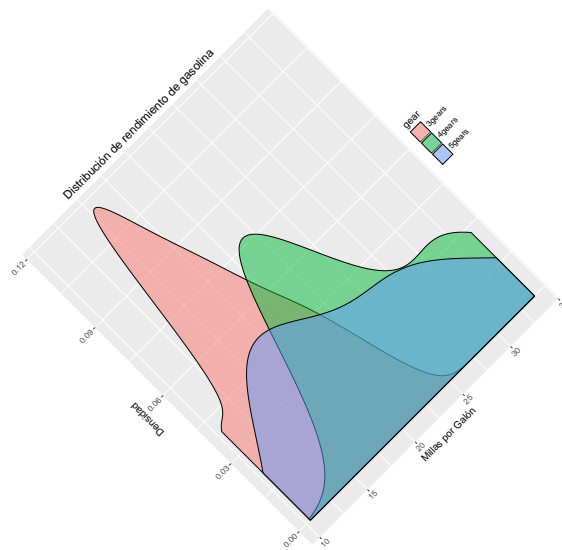


Figura 2: Ejemplo del resultado obtenido usando la configuración del chunk mencionada. Para el ejemplo se usó el dataset *mtcars* y el paquete *ggplot2* de R

```
plot(rep(0:1, 10), pch = 1:20, col = 2, xlab = "xlab font",
     ylab = "ylab font")
mtext("Familia Bookman para un device PDF", side = 3, cex = 1.2)
text(6, 0.5, "Aa Bb Cc\nRr Ss Tt\nXx Yy Zz", cex = 1.5)
text(16, 0.5, "y", cex = 6, col = 3)
```

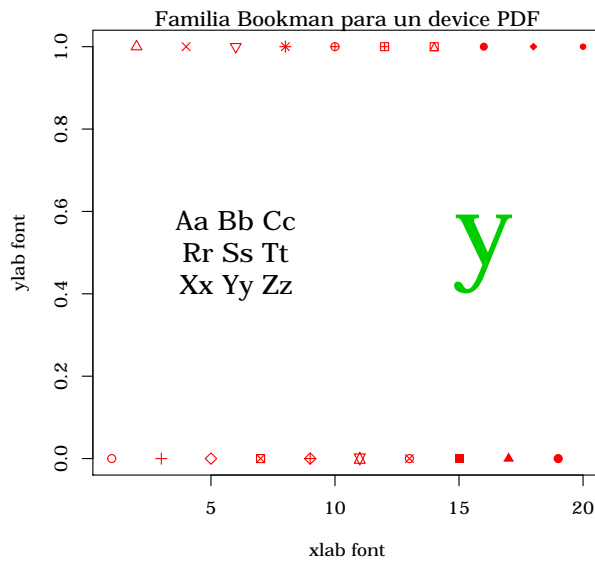


Figura 3: Ejemplo del resultado obtenido al cambiar la fuente de la gráfica. Para el ejemplo se usó la fuente *Bookman*

- Del paquete Cairo para imágenes de alta calidad, requiere instalación manual en R
  - cairo\_pdf
  - svg: graficas vectoriales
- quartz : solo para MAC

Además de la posibilidad de definir el alto, ancho e inclinación de las gráficas, se pueden configurar aún más opciones usando el comando `dev.arg`. Por ejemplo, se puede configurar la fuente de la gráfica introduciendo la familia de la fuente como argumento de `dev.arg`. En la Figura 3 se muestra un ejemplo de como cambia la fuente en las gráficas al usar el comando `dev.args = list(family = 'Bookman')`.

## Múltiples gráficas por Chunk

Knitr permite la creación de varias gráficas en un solo chunk, esto reduce el trabajo necesario si se tiene un procedimiento largo y se quiere observar diferentes resultados o diferentes etapas del análisis. El comando `fig.show` permite configurar la forma en la que se muestran las múltiples imágenes del chunk en el documento. Por defecto, el argumento de entrada de `fig.show` es `'asis'`, lo cual indica que las gráficas se insertaran en el mismo instante en el que son creadas. Por otra parte, si se usa `'hold'` para configurar `fig.show`, se mostrarán todas las gráficas al final del chunk, es decir después de que se terminen de ejecutar todas las órdenes en él.

## Animación

Una tercera opción para configurar la opción `fig.show` es el argumento de entrada `'animate'`. Este valor permite mostrar las imágenes generadas por el chunk en secuencia, generando una animación. Para observar las animaciones generadas de esta forma se debe instalar el paquete `R2SWF` en R y se debe añadir el paquete `animate` (`\usepackageanimate`) en el preámbulo de LaTeX. Sin embargo, este tipo de animaciones solo pueden observarse si el pdf generado es abierto usando adobe reader. Un ejemplo de la animación generada con este método se muestra en la figura 4

```

par(mar = rep(3, 4))
for (i in seq(pi/2, -3/2 * pi, length = 60)) {
plot(0, 0, pch = 20, ann = FALSE, axes = FALSE)
arrows(0, 0, cos(i), sin(i))
axis(1, 0, "VI"); axis(2, 0, "IX")
axis(3, 0, "XII"); axis(4, 0, "III"); box()
}

```

Figura 4: Ejemplo del resultado obtenido al cambiar la fuente de la gráfica. Para el ejemplo se usó la fuente *Bookman*