

# Generación automática de informes con Sweave y $\text{\LaTeX}$

Francesc Carmona  
Departamento de Estadística  
Universidad de Barcelona

Marzo 2012

## 1. Introducción

Como dice el manual oficial: “Sweave proporciona un marco flexible para mezclar texto y código R con el objetivo de generar documentos de forma automática”. Un único archivo fuente contiene el texto y el código R y tras el proceso se obtiene un documento final con

- el texto con formato
- el código R (que también se puede ocultar) y/o
- los resultados del código (texto, cálculos, gráficos)

Este proceso permite regenerar un informe cuando los datos cambian o reproducir un análisis con el mismo archivo con ligeras modificaciones y generar el correspondiente informe. La idea es incluir el código S para el análisis en un documento  $\text{\LaTeX}$  con una sintaxis noweb (Ramsey, 1998). Noweb es una herramienta de programación literal que permite combinar código fuente de programas y la documentación correspondiente dentro de un mismo archivo. Sweave reemplaza el código con sus resultados, de forma que el documento final contiene el texto y los resultados del análisis estadístico. Así se combinan la potencia del  $\text{\LaTeX}$  para dar formato a documentos de gran calidad y el análisis de los datos con R. Muchos de los usuarios de R son también usuarios de  $\text{\LaTeX}$  aunque, en caso contrario, Sweave es un buen argumento para aprender  $\text{\LaTeX}$ . Seguramente, los usuarios de  $\text{\LaTeX}$  disponen de un editor de texto como Emacs que tiene un modo ESS noweb ideal para nuestro propósito. En cualquier caso, Sweave es un sistema independiente y se puede utilizar con cualquier editor de texto, aunque siempre es preferible uno que nos ayude con  $\text{\LaTeX}$  y/o R como WinEdt, Tinn-R, . . . Aunque se pueden crear diferentes drivers para trasladar los archivos Sweave a diferentes lenguajes ( $\text{\LaTeX}$ , HTML, . . .) con diferentes motores S (R, S-Plus, . . .), actualmente únicamente disponemos del driver `RweaveLatex` que combina R y  $\text{\LaTeX}$ .

### Archivos noweb

Un archivo noweb es un archivo de texto simple que consiste en varios segmentos separados de texto y código de programación. Estos segmentos se llaman trozos (chunks). Los hay de dos tipos:

**Trozos de texto** que empiezan con el signo `@` como primer carácter, seguido por un espacio en blanco o un carácter de salto de línea. Si se escribe algo después del espacio en blanco, se considera un comentario y se ignora. Lo normal en estos trozos es introducir texto con instrucciones para darle formato como en  $\text{\LaTeX}$ .

**Trozos de código** que empiezan con `<<name>>=` al principio de línea. Aquí también el resto de la línea puede ser un comentario y es ignorado.

Un archivo fuente para Sweave es un archivo noweb estándar con algún añadido extra para controlar mejor el resultado final. La extensión habitual para los archivos noweb es `.nw` y así se puede escribir para los archivos Sweave. Además, Sweave reconoce las extensiones `.rnw`, `.Rnw`, `.snw` y `.Snw`. De todas ellas, nosotros nos inclinamos por `.Rnw` ya que refleja la relación con R.

En la figura 1 se muestra el procedimiento completo para trabajar con Sweave. Desde la línea de comandos de R y si tenemos la carpeta de trabajo adecuada, simplemente haremos:

```
> Sweave("doc.Rnw")
```

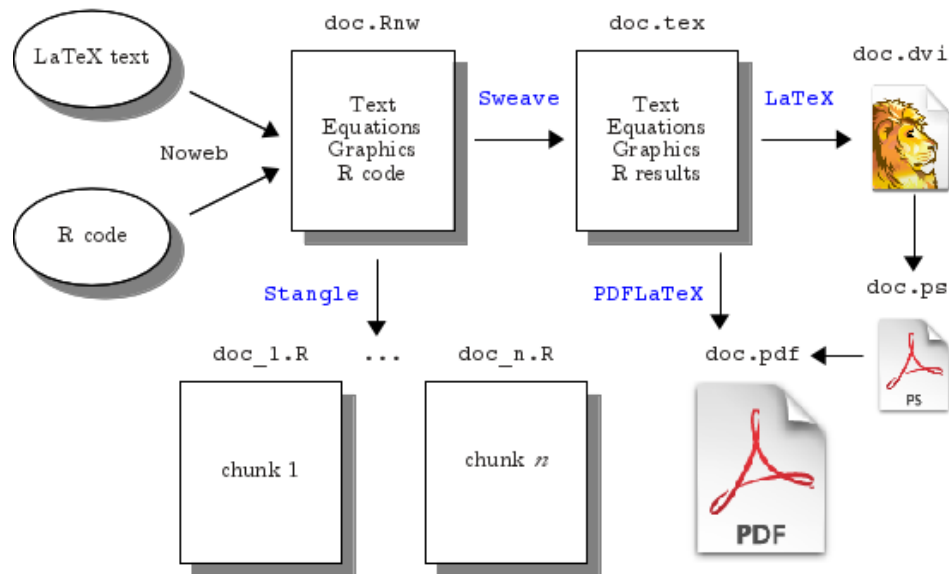


Figura 1: Diagrama de archivos para trabajar con Sweave.

y obtenemos el archivo `doc.tex` para procesar con  $\text{\LaTeX}$  o  $\text{\PDFLaTeX}$ .  
 Por otra parte, si hacemos:

```
> Stangle("doc.Rnw")
```

el archivo que se obtiene es `doc.R` que contiene el código R de todo el documento original. Con la opción `split=TRUE` activa, los diversos trozos se escriben en archivos separados.

## 2. Un ejemplo sencillo

A continuación presentamos un ejemplo sencillo de archivo Sweave que contiene dos trozos de código entre el texto y las instrucciones de formato de un típico archivo  $\text{\LaTeX}$  que llamaremos `ejemplo.Rnw`. Se trata de escribir el texto como vemos en la figura 2 en un programa como `RStudio` o cualquier procesador de archivos  $\text{\LaTeX}$  como `TEXmaker`, `WinEdt`, `Kile`, etc.

A continuación se procesa este archivo con extensión `.Rnw` con `Sweave`. Suponiendo que tengamos los programas necesarios bien configurados, este proceso se puede hacer de varias formas.

El procedimiento más sencillo es desde `RStudio`, ya que obtiene el archivo PDF directamente con la instrucción `compilePdf()` o con el botón correspondiente.

```
> compilePdf("~/R/Sweave/ejemplo.Rnw")
```

Así se obtiene el archivo `ejemplo.pdf` que podemos ver en la figura 4.

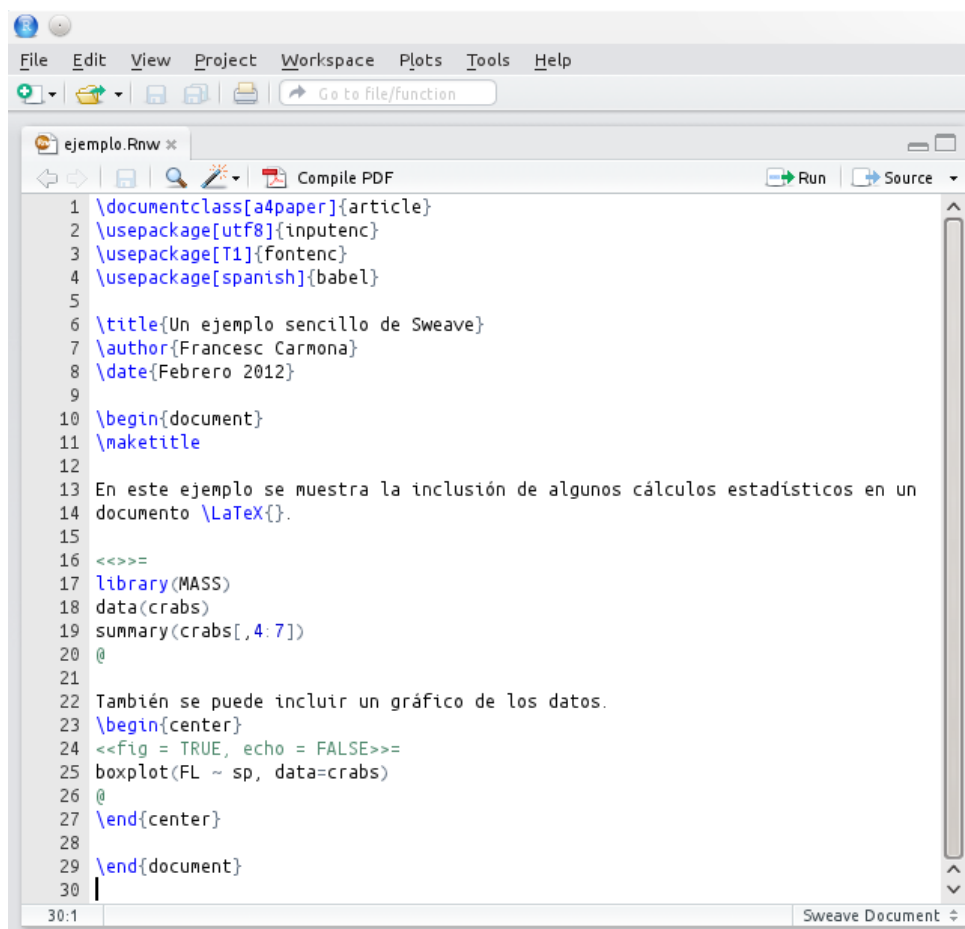
En realidad el procedimiento consiste en dos pasos. El primero para generar el archivo `ejemplo.tex` y el siguiente para procesar este archivo con  $\text{\LaTeX}$  y obtener el archivo `ejemplo.pdf`. Los archivos `ejemplo.tex` y `ejemplo.pdf` se dejan en la misma carpeta de trabajo, junto con sus archivos asociados como `ejemplo.log`. En la figura 3 podemos ver el archivo `ejemplo.tex`.

Otro procedimiento, como se ha explicado, consiste en generar el archivo `ejemplo.tex` desde la consola de una sesión de R con la instrucción

```
> Sweave("ejemplo.Rnw")
```

```
Writing to file ejemplo.tex
Processing code chunks with options ...
 1 : echo keep.source term verbatim
 2 : keep.source term verbatim pdf
```

```
You can now run (pdf)latex on 'ejemplo.tex'
```



The image shows a screenshot of the RStudio interface. The main window displays the content of a file named 'ejemplo.Rnw'. The code is a Sweave document, which is a LaTeX document with embedded R code. The code includes package loading, document metadata, a title, author, and date. It also contains R code for loading a library, creating a data object, and performing a summary and a boxplot. The document is currently at line 30, column 1.

```
1 \documentclass[a4paper]{article}
2 \usepackage[utf8]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage[spanish]{babel}
5
6 \title{Un ejemplo sencillo de Sweave}
7 \author{Francesc Carmona}
8 \date{Febrero 2012}
9
10 \begin{document}
11 \maketitle
12
13 En este ejemplo se muestra la inclusión de algunos cálculos estadísticos en un
14 documento \LaTeX{}.
15
16 <<>>=
17 library(MASS)
18 data(crabs)
19 summary(crabs[,4:7])
20 @
21
22 También se puede incluir un gráfico de los datos.
23 \begin{center}
24 <<fig = TRUE, echo = FALSE>>=
25 boxplot(FL ~ sp, data=crabs)
26 @
27 \end{center}
28
29 \end{document}
30 |
```

Figura 2: Archivo ejemplo.Rnw en RStudio.

```
1 \documentclass[a4paper]{article}
2 \usepackage[utf8]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage[spanish]{babel}
5
6 \title{Un ejemplo sencillo de Sweave}
7 \author{Francesc Carmona}
8 \date{Febrero 2012}
9
10 \usepackage{Sweave}
11 \begin{document}
12 \maketitle
13
14 En este ejemplo se muestra la inclusión de algunos cálculos estadísticos en un
15 documento \LaTeX}.
16
17 \begin{Schunk}
18 \begin{Sinput}
19 > library(MASS)
20 > data(crabs)
21 > summary(crabs[,4:7])
22 \end{Sinput}
23 \begin{Soutput}
24
25      FL           RW           CL           CW
26 Min.   : 7.20   Min.   : 6.50   Min.   :14.70   Min.   :17.10
27 1st Qu.:12.90   1st Qu.:11.00   1st Qu.:27.27   1st Qu.:31.50
28 Median :15.55   Median :12.80   Median :32.10   Median :36.80
29 Mean   :15.58   Mean   :12.74   Mean   :32.11   Mean   :36.41
30 3rd Qu.:18.05   3rd Qu.:14.30   3rd Qu.:37.23   3rd Qu.:42.00
31 Max.   :23.10   Max.   :20.20   Max.   :47.60   Max.   :54.60
32 \end{Soutput}
33 \end{Schunk}
34
35 También se puede incluir un gráfico de los datos.
36 \begin{center}
37 \includegraphics{ejemplo-002}
38 \end{center}
39 \end{document}
40
```

Figura 3: Archivo `ejemplo.tex` en RStudio.

## Un ejemplo sencillo de Sweave

Francesca Carmona

Febrero 2012

En este ejemplo se muestra la inclusión de algunos cálculos estadísticos en un documento  $\LaTeX$ .

```
> library(MASS)
> data(crabs)
> summary(crabs[,4:7])
```

FL	RW	CL	CW
Min. : 7.20	Min. : 6.50	Min. :14.70	Min. :17.10
1st Qu.:12.90	1st Qu.:11.00	1st Qu.:27.27	1st Qu.:31.50
Median :15.55	Median :12.80	Median :32.10	Median :36.80
Mean :15.58	Mean :12.74	Mean :32.11	Mean :36.41
3rd Qu.:18.05	3rd Qu.:14.30	3rd Qu.:37.23	3rd Qu.:42.00
Max. :23.10	Max. :20.20	Max. :47.60	Max. :54.60

También se puede incluir un gráfico de los datos.

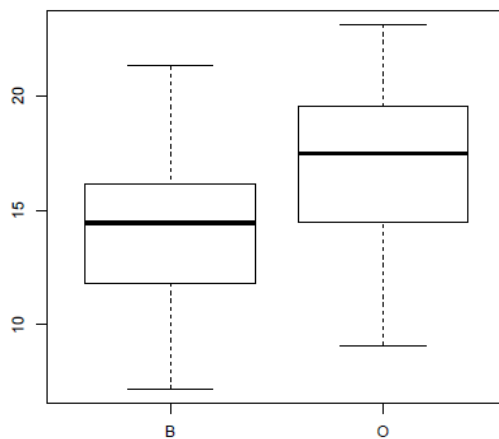


Figura 4: Archivo ejemplo.pdf.

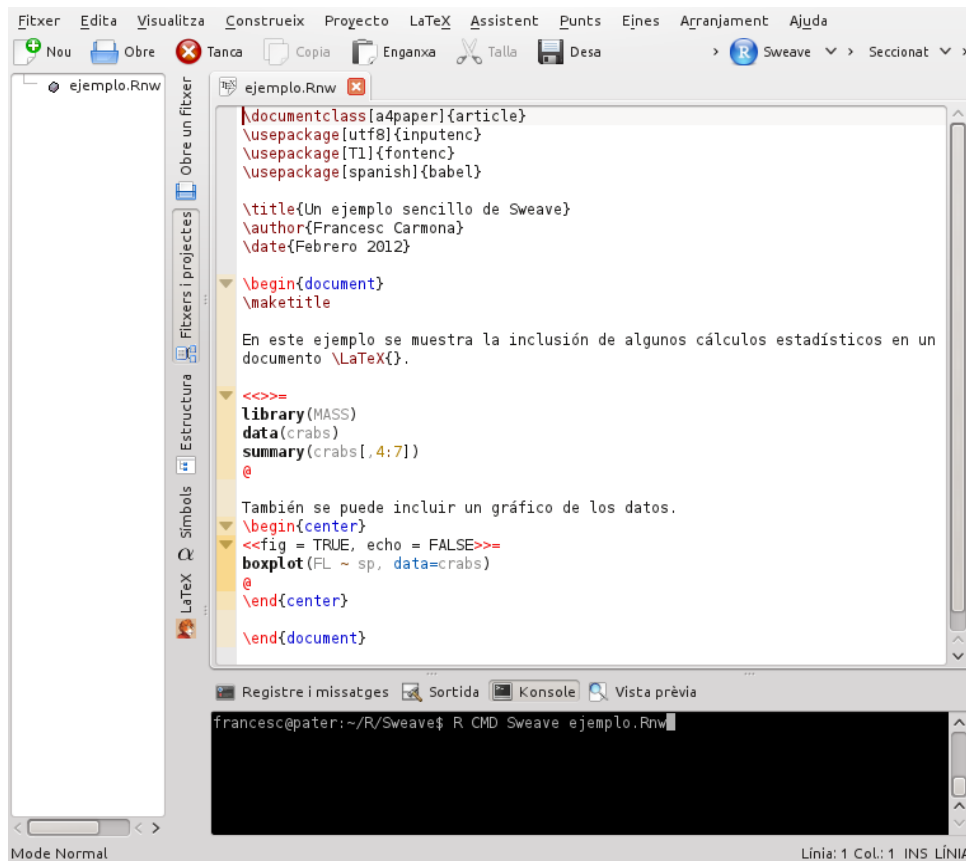


Figura 5: Archivo `ejemplo.Rnw` en Kile.

Para después procesar el archivo `ejemplo.tex` también en la misma sesión R:

```
> tools::texi2pdf("ejemplo.tex")
```

Sin embargo, es posible que sea más productivo escribir el documento `ejemplo.Rnw` en un editor  $\text{\LaTeX}$ , obtener el documento `ejemplo.tex` con Sweave y abrir este último con el mismo editor para procesarlo y obtener el PDF.

Desde una consola del sistema, sin necesidad de abrir una sesión R, y si estamos en la carpeta de trabajo podemos ejecutar Sweave con la instrucción:

```
$ R CMD Sweave ejemplo.Rnw
```

Entonces el archivo `ejemplo.tex` se abre o se actualiza en el editor  $\text{\LaTeX}$  y se procesa con el  $\text{\PDF\LaTeX}$ . Podemos ver este procedimiento con el editor Kile en la figura 5.

Además podemos incorporar esta instrucción a las opciones del editor. Por ejemplo, en la configuración del editor  $\text{\TeX}$ maker que podemos ver en la figura 6 tenemos dicha instrucción lista para ejecutarse desde el menú.

Finalmente hay una tercera posibilidad mediante la ejecución en la consola del sistema o la inclusión en el editor  $\text{\LaTeX}$  de un archivo `Sweave.sh` o `Sweave.bat`, según el sistema operativo<sup>1</sup>. Por ejemplo, una vez obtenido e instalado correctamente el archivo `Sweave.sh` en un sistema Linux, podemos ejecutar en la consola la instrucción

```
$ Sweave -ld ejemplo.Rnw
```

y obtener directamente el archivo PDF. La ventaja de este script, que se puede obtener del CRAN, son sus múltiples opciones. Por ejemplo, con la siguiente instrucción

```
$ Sweave -otld=acrobat ejemplo.Rnw
```

<sup>1</sup>Consultar el artículo sobre la instalación en Linux, Windows y Mac OS en el **Blog de los errores**: <http://erre-que-erre-paco.blogspot.com/2012/02/sweave-desde-un-editor-de-texto.html>.

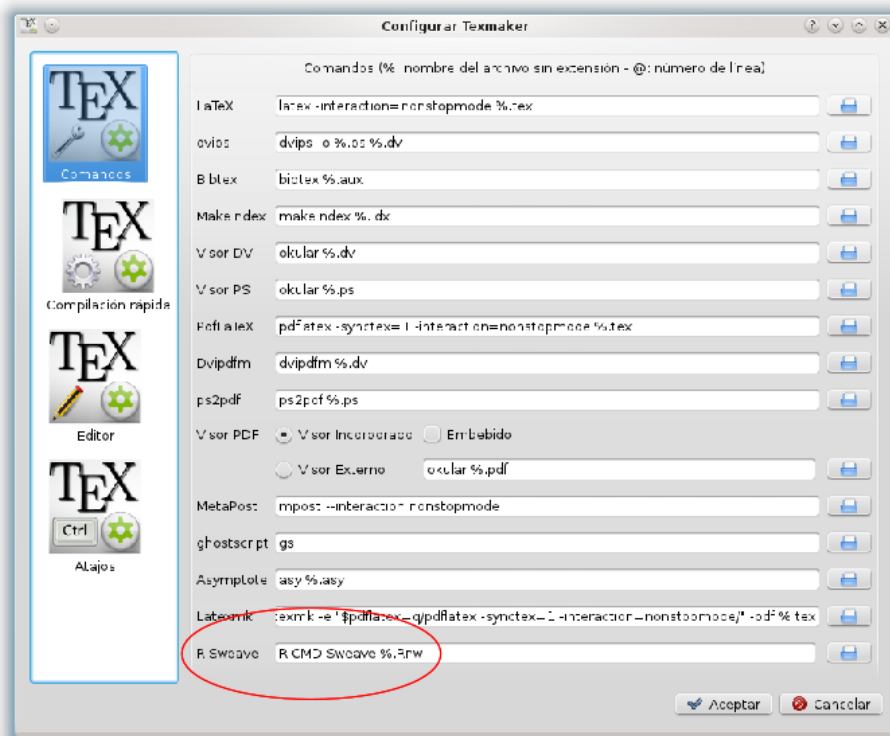


Figura 6: Configuración de TeXmaker.

se crea el archivo `ejemplo.pdf` y se abre con un visor “no estándar”.

Un detalle importante es la utilización de una codificación especial de los caracteres en los trozos de código  $\LaTeX$  y en los trozos de código R. En el ejemplo se ha utilizado la instrucción

```
\usepackage[utf8]{inputenc}
```

propia de los documentos escritos en Linux. El paquete `inputenc` de  $\LaTeX$  proporciona muchas más codificaciones. Este tipo de codificación de caracteres es más difícil de usar en los trozos de código R y es mejor leer atentamente el apartado 3.7 del manual de Sweave.

### 3. Las modificaciones que produce Sweave

Ahora fijemos nuestra atención en el archivo `ejemplo.tex` de la figura 3. Observemos que en el archivo se ha introducido el estilo `Sweave.sty` que proporciona las definiciones de los entornos del código R tanto de entrada `Sinput` como de salida `Soutput`. Además los trozos de texto  $\LaTeX$  del documento original se han copiado sin ninguna modificación.

El verdadero trabajo de Sweave se muestra en los trozos de código.

Los nombres de los trozos de código se utilizan para pasar opciones de Sweave diferentes a las previstas por defecto y que modifican el resultado final en cada segmento. Ésta es una característica especial de Sweave frente a la sintaxis noweb.

La opción `fig=TRUE` ordena a Sweave que grave un archivo en el formato PDF (por defecto) para el gráfico creado con las instrucciones del código. Además, se escribe un `\includegraphics` en el archivo  $\LaTeX$  para que muestre el gráfico insertado en el documento definitivo.

La opción `echo=FALSE` indica que el código R no se incluirá en el documento final, es decir, no hay entorno `Sinput` para este trozo.

En el siguiente apartado se explican todas las posibles opciones.

## 4. Las opciones de Sweave

Las opciones controlan el modo en que el código y sus resultados (texto, figuras) se transfieren del archivo `.Rnw` al archivo `.tex`. Todas las opciones tienen la forma `clave=valor`, donde el valor puede ser un número, una palabra o *string* o un valor lógico. Se pueden especificar varias opciones a la vez, separadas por comas. Las opciones lógicas pueden tomar el valor verdadero (`true` o `TRUE` o abreviadamente `t` o `T`) o falso (`false` o `FALSE`, `f` o `F`).

Las opciones pueden especificarse de dos formas:

1. Entre los símbolos `<<>>` al principio del trozo de código, de manera que se modifique la apariencia únicamente para ese trozo, o
2. En cualquier lugar de un trozo de texto con la instrucción

```
\SweaveOpts{opt1=valor, opt2=valor, ..., optN=valor}
```

que modifica los valores por defecto en el resto del documento, desde ese lugar y para todos los siguientes trozos de código.

Además, ciertas opciones globales se pueden especificar (como una lista de `clave=valor` separadas por comas) en la variable de entorno `SWEAVE_OPTIONS` o con el añadido `--options=` a la instrucción `R CMD Sweave`.

Las opciones disponibles dependen del controlador en uso, pero todos los controladores deben tener las dos siguientes opciones (las opciones aparecen con sus valores por defecto, si los hay):

`split=FALSE` un valor lógico de forma que todos los resultados se escriben en un único archivo. Cuando es `TRUE` los resultados se distribuyen en varios archivos.

`label=nombre` es una etiqueta para el trozo que se utiliza como nombre para el archivo, si la opción `split=TRUE` está activa.

En cualquier trozo de código, la primera opción, y sólo la primera, puede ponerse sin clave y entonces se interpreta como el nombre de la etiqueta. Por ejemplo:

```
<<boxplot, split=FALSE>>
```

es lo mismo que

```
<<label=boxplot, split=FALSE>>    o también    <<split=FALSE, label=boxplot>>
```

pero

```
<<split=FALSE, boxplot>>
```

da un error de sintaxis. También se puede dejar sin nombre.

Para que el archivo Sweave sea totalmente compatible con noweb es preciso que los trozos no tengan nombre y todas las opciones se especifiquen con la instrucción `\SweaveOpts`.

El controlador o *driver* de Sweave que traduce los trozos de código R a  $\text{\LaTeX}$  es `RweaveLatex()`.

Las posibles opciones de este dispositivo son (se presentan con su valor por defecto):

`engine="R"` una palabra que describe el motor para procesar el código. Actualmente los únicos valores posibles son "R" y "S".

`echo=TRUE` valor lógico para incluir o no el código R en el archivo de salida.

`keep.source=TRUE` valor lógico que reproduce el código original (por defecto) o el código *deparsed*.

`eval=TRUE` si lo ponemos en `FALSE`, el código no se evalúa y por lo tanto no produce ningún resultado.

`results=verbatim` palabra que indica que los resultados se escribirán en el modo literal (tipo ordenador o consola) con el entorno `Soutput`. Si escribimos `results=tex`, se supone que la salida es ya correcta en  $\text{\LaTeX}$  y se incluye tal cual. Si hacemos `results=hide`, los resultados no se presentan, aunque sí se calculan.



`print=FALSE` si `TRUE`, cada una de las expresiones se rodea con un `print` de forma que los valores de todas las expresiones se pueden ver.

`term=TRUE` el resultado emula una sesión de R. Si `FALSE`, los únicos resultados visibles vienen dados por las instrucciones `print` o `cat`.

`split=FALSE` si `TRUE`, cada uno de los trozos se escribe en archivos separados.

`strip.white="true"` (no es un valor lógico) Si `"true"` se eliminan las líneas en blanco al principio y al final de la salida. Si `"all"`, se eliminan todas. Si `"false"` se conservan todas.

`prefix=TRUE` Si `TRUE`, los nombres de los archivos de las figuras y los de salida tienen un prefijo común.

`prefix.string="nombre"` Por defecto es el nombre del archivo fuente sin la extensión `.Rnw`. Se puede utilizar para especificar un directorio como parte del prefijo, por ejemplo si hacemos

```
\SweaveOpts{prefix.string=figures/fig}
```

los gráficos generados se ubicarán automáticamente en la subcarpeta `figures`. Por supuesto que esta carpeta debe existir.

`include=TRUE` Genera automáticamente la instrucción `\includegraphics`.

`fig=FALSE` Indica si el código generará una figura. Con `fig=TRUE` sólo se puede producir una figura por trozo. La etiqueta del trozo se utiliza en el nombre del archivo gráfico.

`eps=FALSE` Indica si las figuras se escriben en un archivo EPS. Se ignora si `fig=FALSE`.

`pdf=TRUE` Indica si las figuras se escriben en un archivo PDF. Se ignora si `fig=FALSE`.

`png=FALSE` Indica si las figuras se escriben en un archivo PNG. Se ignora si `fig=FALSE`.

`jpeg=FALSE` Indica si las figuras se escriben en un archivo JPEG. Se ignora si `fig=FALSE`.

`grdevice=NULL` Para utilizar un dispositivo gráfico especial o definido previamente.

`width=6` Valor numérico en pulgadas (6in = 15.24cm) de la anchura de las figuras.

`height=6` Valor numérico en pulgadas (6in = 15.24cm) de la altura de las figuras.

`resolution=300` La resolución en pixels por pulgada que se utilizará en los gráficos PNG y JPEG.

`concordance=FALSE` Se trata de escribir un archivo para enlazar los números de las líneas de entrada con los números de las líneas de salida. Es experimental.

`figs.only=FALSE` Por defecto, cada trozo con una figura se ejecuta una vez y se vuelve a ejecutar para cada tipo de gráfico seleccionado. Eso abrirá un dispositivo gráfico predeterminado para el trozo de la primera figura y utilizará ese dispositivo para la primera evaluación de todos los bloques siguientes. Si esta opción es `TRUE`, el trozo sólo se ejecuta para cada tipo de gráfico seleccionado, de modo que un nuevo dispositivo gráfico se abre y se cierra.

Más detalles de estas opciones se pueden ver en la ayuda de la función

```
> help(RweaveLatex)
```

## Figuras

Respecto al tamaño de las figuras, hay que notar que el estilo `Sweave.sty` fija el ancho de la figura  $\LaTeX$  (independiente del tamaño del gráfico generado) en

```
\setkeys{Gin}{width=0.8\textwidth}
```

Si se desea usar otro ancho para las figuras generadas automáticamente por Sweave, se puede añadir una instrucción como la anterior *después* de `\begin{document}`. Si lo que se quiere es que no haya un valor por defecto para el ancho de las figuras, basta con incluir una instrucción `\usepackage[nogin]{Sweave}` en el preámbulo del archivo. Recordemos también que para cada trozo de código que genera una figura se abre un nuevo dispositivo y es aquí donde podemos fijar los parámetros gráficos con `par()`. Estos parámetros se olvidan al acabar el trozo y, por tanto, cerrar el dispositivo gráfico.

El dispositivo gráfico por defecto para los trozos de código con `fig=TRUE` es `pdf()`. Las opciones `pdf`, `eps`, `png` y `jpeg` permiten la generación en esos formatos para un trozo particular o (vía `\SweaveOpts`) para todo el documento. Para algún trozo particular, puede ser conveniente seleccionar el formato PNG con algo así:

```
<<NombreFig, fig=TRUE, pdf=FALSE, png=TRUE>>
```

de forma que `pdflatex` incluirá automáticamente el archivo `.png` para ese trozo.

Desde la versión 2.13.0 de R, ya es posible definir nuestro propio dispositivo gráfico y utilizarlo con la opción `grdevice=mi.Swd`, donde la función dispositivo (aquí `mi.Swd`) se debe definir en un trozo de código oculto.

## 5. Uso de múltiples archivos de entrada

Los archivos  $\LaTeX$  pueden incluir otros archivos del mismo tipo con la instrucción `\input{}`. Esto también se puede hacer en archivos Sweave, pero estos archivos serán incluidos simplemente como archivos  $\LaTeX$  y Sweave no los procesará. Si se desea que sean procesados por Sweave, la instrucción equivalente es `\SweaveInput{}`.

Los archivos así incluidos deben utilizar la misma sintaxis Sweave y la codificación del archivo principal.

## 6. Introducción de escalares en el texto

Sweave proporciona una función específica para escribir resultados en medio de un trozo de texto. Se trata de la función `\Sexpr{expr}` que tiene un uso limitado, ya que transforma el valor de `expr` en un vector de caracteres y muestra el primer elemento. Por ejemplo, `\Sexpr{round(pi,4)}` se sustituye por 3.1416 en medio de un párrafo.

La expresión se evalúa en el mismo entorno que los trozos de código, de manera que podemos acceder a los objetos definidos previamente. En la expresión no puede haber llaves, si las necesitamos se pueden escribir en un trozo de código anterior (oculto o no).

## 7. Inclusión de tablas en formato $\LaTeX$

Se puede utilizar el paquete `xtable` de R con la opción `results=tex` para insertar tablas con formato  $\LaTeX$ . Seguramente deberemos instalar previamente en R este paquete.

Por ejemplo, el código:

```
<<echo = F, results = tex>>=
library(xtable)
g <- lm(FL ~ sp, data=crabs)
xtable(anova(g),"Análisis de la varianza")
@
```

proporciona el oportuno código  $\LaTeX$  para que la tabla quede formateada con estilo:

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
sp	1	466.35	466.35	45.99	0.0000
Residuals	198	1964.90	9.92		

Table 1: Análisis de la varianza

## 8. El paquete knitr

El paquete Sweave no resuelve muchos de los problemas relacionados con el código R y los gráficos generados. Recientemente ha aparecido el paquete `knitr` de Yihui Xie[1] que pretende ser modular, incluir muchas de las soluciones paralelas a Sweave como `cacheSweave` y otras, más flexible e integrador. Entre las características y funcionalidades de este paquete tenemos:

- mejoras en el tratamiento de los resultados como prints, plots o incluso warnings y mensajes. Inclusión de los gráficos generados por paquetes como `ggplot2` o `lattice`, sin necesidad de la instrucción `print()`.
- incorpora y mejora las ideas del paquete `cacheSweave`.
- un mejor formato del código R con el paquete `formatR`: líneas largas, espacios, sangrías,...
- soporte para 20 dispositivos gráficos, por ejemplo `CairoPNG()` o `tikz()`.
- mayor flexibilidad en los gráficos: mejor control del ancho y alto de los gráficos en el dispositivo y en el documento. Control de la localización del gráfico en el documento. Combinación de gráficos estándar y gráficos *grid-based*.
- el código R puede incorporarse desde *scripts* externos.
- mayor personalización: es posible redefinir las instrucciones que abren con `<<>>` y cierran con `@` los trozos de código o la instrucción `\Sexpr{}`.

También se ha mejorado el aspecto final de los trozos de código R que están resaltados y con un entorno en gris suave gracias al paquete `framed`.

## Referencias

- [1] Yihui Xie (2012) `knitr`: A General-Purpose Tool for Dynamic Report Generation in R. <http://yihui.name/knitr/>.
- [2] Friedrich Leisch (2002). Sweave: Dynamic generation of statistical reports using literate data analysis. In Wolfgang Härdle and Bernd Rönz, editors, *Compstat 2002 — Proceedings in Computational Statistics*, pages 575–580. Physica Verlag, Heidelberg. ISBN 3-7908-1517-9. <http://www.ci.tuwien.ac.at/~leisch/Sweave>
- [3] R Development Core Team (2005). R: A language and environment for statistical computing, reference index version 2.14.2. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, <http://www.r-project.org>
- [4] Norman Ramsey (1998). *Noweb home page*. University of Virginia, USA. Version 2.9a, <http://www.cs.virginia.edu/~nr/noweb>.
- [5] Morales Rivera, M.A. Generación automática de reportes con R y  $\LaTeX$ .
- [6] Integrate `Sweave.sty` in the  $\TeX$ live 2010 search path on Linux. <http://tex.stackexchange.com/questions/6763/integrate-sweave-sty-in-the-texlive-2010-search-path-on-linux>