

Generación de Documentos Científicos Integrando Gnuplot, R y L^AT_EX

RICARDO A. SALGADO¹

¹Escuela de Física - UNAH, mail: ricardo.salgado@unah.hn

Recibido: 16 de Octubre de 2015 / Aceptado: 10 de Noviembre de 2015

Resumen

R, Gnuplot and LaTeX are free tools and computing environments, which have been developed for data manipulation, statistical analysis, graphics generation and the production of high quality text documents. Each offers advantages and a lot of options that are useful in developing a scientific document, either a laboratory report, an article or a simple data analysis, but unfortunately many times we feel the need to work separately in all three environments and then unify everything in one file, something that ultimately becomes a cumbersome procedure and not as optimal.

The main purpose of this paper is to show how you can integrate these three powerful environments through Sweave and Gnuplottex, so that there is a single source file containing both text as command lines each, allowing well, optimally generate a dynamic and quality document.

Keywords: Sweave, Gnuplottex, Chunks, integrated files

R, Gnuplot y L^AT_EX son herramientas y entornos de computación de uso libre, que han sido desarrollados para la manipulación de datos, análisis estadístico, generación de gráficos y elaboración de documentos de texto de alta calidad. Cada uno ofrece ventajas y una gran cantidad de opciones que resultan útiles al elaborar un documento científico, ya sea un reporte de laboratorio, un artículo o un sencillo análisis de datos. Al elaborar documentos científicos en muchas ocasiones nos vemos en la necesidad de trabajar por separado en varios entornos y luego unificar todo en un solo archivo, algo que al final se vuelve un procedimiento engorroso y no tan óptimo.

El propósito principal de este trabajo es mostrar como se pueden integrar los entornos R, Gnuplot y L^AT_EX a través de Sweave y Gnuplottex, de manera que, exista un solo archivo fuente que contenga, tanto el texto como las líneas de comando de cada uno de ellos, permitiendo así, generar de forma óptima un documento dinámico y de calidad.

Palabras clave: Sweave, Gnuplottex, Chunks, archivos integrados.

I. INTRODUCCIÓN

Sweave proporciona un marco flexible para mezclar texto y código R en los documentos L^AT_EX con el objetivo de generar un documento de forma automática[1], mientras tanto Gnuplottex [2] permite incluir gráficos de gnuplot en los documentos de L^AT_EX, todo esto a través de la sintaxis noweb [3], que es una herramienta de programación-sencilla que permite la combinación de código en una secuencia de segmentos separados de texto y código de programación llamados **chunks**.

Mediante la utilización de los chunks, todo el poder de L^AT_EX, para la generación de documentos de alta calidad, de R para el análisis de datos y Gnuplot para la elaboración de gráficos se puede utilizar simultáneamente en un solo archivo fuente. Este archivo fuente tendrá la extensión **.nw**, la extensión noweb por defecto, pero, para nuestro propósito se utilizará la extensión **.Rnw**, ya

que es un archivo tipo R-noweb, es decir, que sus chunks contendrán código R.

Finalmente el archivo fuente contendrá el código de L^AT_EX, los trozos de código R y por medio del paquete Gnuplottex contenido dentro de la distribución L^AT_EX, se incluye el código Gnuplot; tras el proceso se obtiene un documento final con: el texto formateado, los códigos de R y Gnuplot y los resultados, es decir texto, ecuaciones, tablas, cálculos y gráficos. Este tipo de archivo integrado permite la generación y re-generación de un informe si existe un cambio en los datos de entrada o en el texto; también permite reproducir un análisis de datos con el mismo archivo haciendo ligeras modificaciones al código, sin necesidad de empezar de cero un nuevo documento.

En la figura 1 se resume el proceso de generación del archivo integrado.

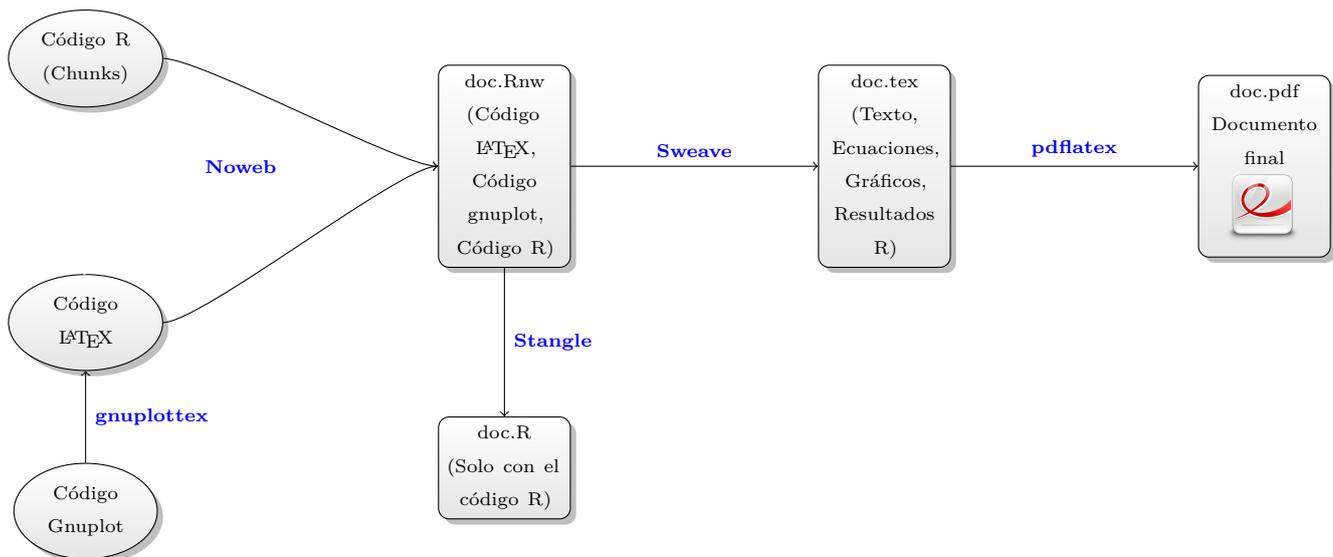


Figura 1: Diagrama del proceso de integración entre Gnuplot, \LaTeX y R

II. GNUPLOTTEX

El paquete **Gnuplottex** permite incluir gráficos gnuplot en sus documentos \LaTeX . [2].

Su funcionamiento es muy básico, el código de gnuplot se extrae a partir del documento y se escribe en un archivo .gnuplot, Si se tiene activada la opción shell escape en la distribución \LaTeX , los archivos de gráficos se procesan automáticamente junto con el código \LaTeX y serán incluidos en el documento final. Si no se utiliza shell escape, el usuario tendrá que convertir manualmente los archivos mediante la ejecución de gnuplot en los archivos .gnuplot obtenidos.

La opción shell escape está disponible en las distribuciones \LaTeX , como Texlive y permite la ejecución de código shell durante la compilación de un archivo .tex. Está desactivada por defecto, el usuario tendrá que editar los archivos de configuración de la distribución o permitir la opción shell escape en su editor.

A continuación se presenta como activar Shell scape en Texlive desde la consola de GNU Linux

- Abrir el siguiente archivo de la distribución Texlive:


```
sudo gedit /usr/share/texmf/web2c/texmf.cnf
```
- Buscar `shell_escape = p`, y cambiar la p por t.
- Guardar y cerrar
- Recompilar sus formatos desde la consola con:


```
texexec --make --all
```
- También usar:


```
texexec --make --all cont-xx
```

Para cargar el paquete gnuplottex basta escribir la instrucción:

```
\usepackage{gnuplottex}
```

en el preámbulo del documento.

Algunas de las opciones que acompañan al paquete son:

- [`\nosshell`]: No usa shell escape, por tanto los gráficos se deberán generar manualmente. La opción por defecto es shell.
- [`\subfolder`]: Genera los gráficos en un directorio secundario “gnuplottex”, el cuál se crea automáticamente.
- [`\cleanup`]: Elimina los archivos **.gnuplot** después de la conversión.

Para hacer uso del código gnuplot en un documento \LaTeX se usa el entorno:

```
\begin{gnuplot}[terminal = <terminal>]
...
\end{gnuplot}
```

La terminal latex, es la terminal por defecto en este entorno, aunque existen otras terminales que pueden utilizarse con el paquete, entre las más destacadas se tiene:

- eps
- epslatex
- cairolatex
- tikz

A. Ejemplos

A continuación se presentan algunos ejemplos básicos de cada terminal, con el fin de mostrar las diferencias entre cada una de ellas y su aplicación e integración con \LaTeX .

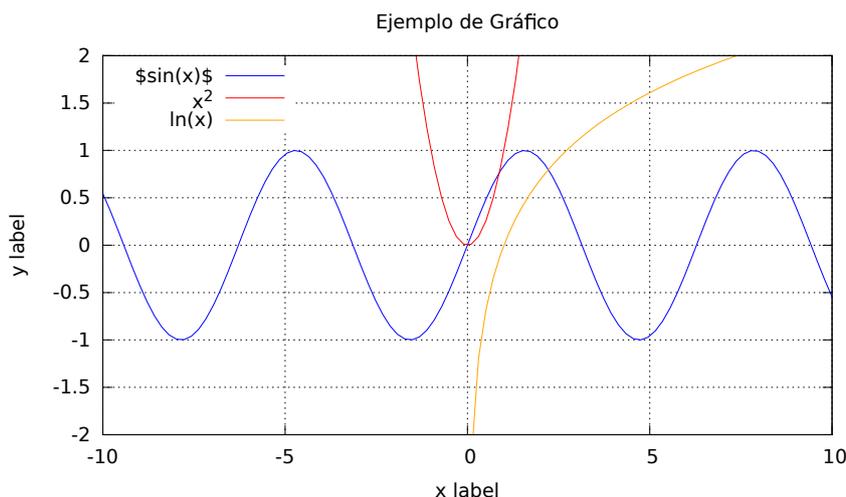


Figura 2: Ejemplo de gráfico usando la terminal eps

A.1. Terminal eps

El código utilizado para generar el gráfico de la figura 2 se muestra a continuación:

```

1 \begin{gnuplot}
2 [terminal=eps, terminaloptions=enhanced]
3 set title "Ejemplo de Gráfico"
4 set xlabel "x label"
5 set ylabel "y label"
6 set grid
7 set yrange [-2:2]
8 set key left top
9 plot sin(x) lc rgb "blue" t "$\sin(x)$",
10 x**2 lc rgb "red" t "x^2",
11 log(x) lc rgb "orange" t "ln(x)"
12 \end{gnuplot}

```

Código 1: Ejemplo de código para gráfico eps

Se puede observar que la figura 2 tiene una buena calidad, como es de esperarse con la terminal eps, pero es importante recalcar que esta no reconoce código L^AT_EX, lo cual se puede apreciar en la leyenda del $\sin x$ donde se intentó utilizar a manera de prueba el modo matemático de L^AT_EX.

A.2. Terminal epslatex

Una buena opción, si se desea utilizar la terminal eps junto a los comandos de L^AT_EX, es la terminal epslatex, que unifica las virtudes de ambas terminales.

El siguiente script es el encargado de generar el gráfico mostrado en la figura 3:

```

1 \begin{gnuplot}[terminal=epslatex]
2 set title "Ejemplo de Gráfico"
3 set xlabel "$x$ label"
4 set ylabel "$y$ label"
5 set grid
6 set yrange [-2:2]
7 set key left top
8 plot sin(x) lc rgb "blue" t '$\sinx$',

```

```

9         x**2 lc rgb "red" title '$x^2$',
10        log(x) lc rgb "orange" title
11        '$\ln(x)$'
12 \end{gnuplot}

```

Código 2: Ejemplo de código para gráfico epslatex

Se puede apreciar que esta terminal produce una salida con muy buena calidad, en la cual, se ha integrado el uso de código L^AT_EX dentro del script que genera el gráfico, a diferencia del terminal eps mejorado, que no es compatible con ello.

A.3. Terminal cairolatex

Se trata de una terminal que ofrece resultados de alta calidad. A diferencia de la epslatex, la terminal generará salida directamente pdf. Se permite el uso de la opción siunitx, así como la inclusión de comandos L^AT_EX. Con el siguiente script se generará el gráfico mostrado en la figura 4 para esta terminal:

```

1 \begin{figure*}
2 \centering
3 \begin{gnuplot}[terminal=cairolatex]
4 set term cairolatex header "\\small"
5 set title "Ejemplo de Gráfico"
6 set xlabel "$x$ label"
7 set ylabel "$y$ label"
8 set grid
9 set yrange [-2:2]
10 set key left top
11 plot sin(x) lc rgb "blue" t '$\sin(x)$',
12        x**2 lc rgb "red" t '$x^2$',
13        log(x) lc rgb "orange" t '$\ln(x)$'
14 \end{gnuplot}

```

Código 3: Ejemplo de código para gráfico cairolatex

A.4. Terminal tikz

Es probable que sea la mejor terminal de salida actualmente, aunque la diferencia con cairolatex y epslatex no

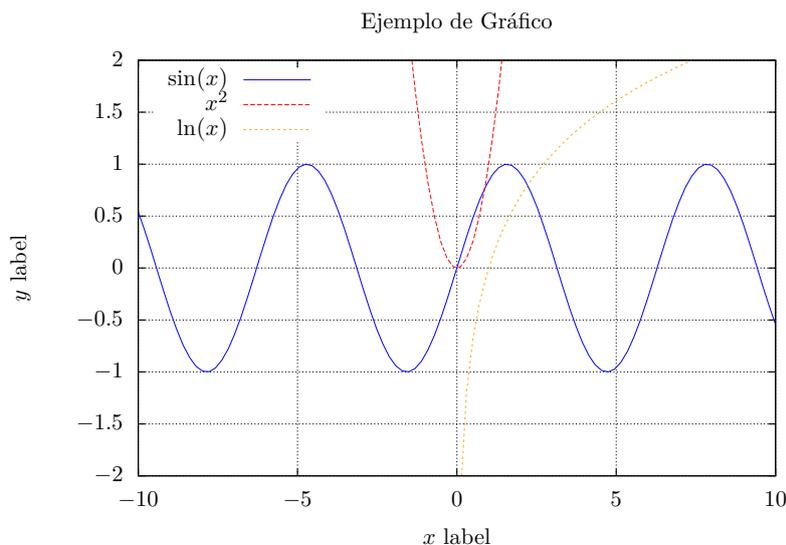


Figura 3: Ejemplo de gráfico usando la terminal *epslatex*

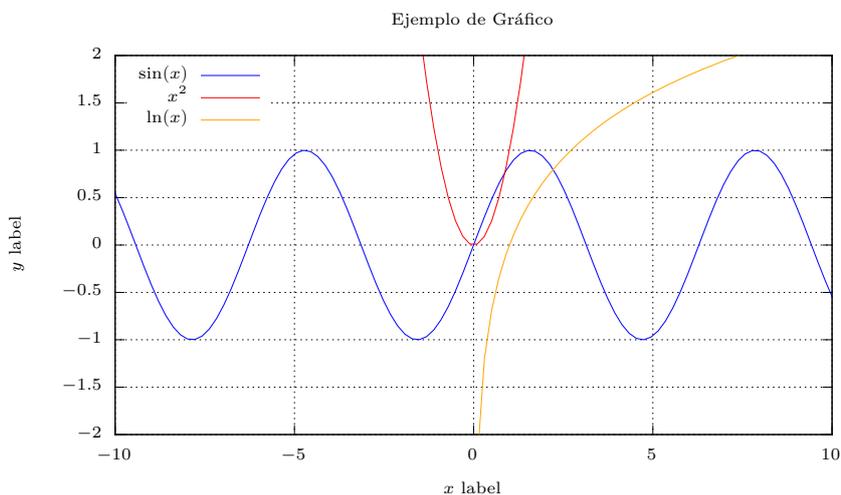


Figura 4: Ejemplo de gráfico usando la terminal *cairolatex*

es tan grande. Así mismo la opción `siunitx` está disponible, los comandos de \LaTeX también se pueden utilizar y una de las virtudes es que se puede exportar fácilmente constantes desde dentro `gnuplot`.

El script para generar el gráfico de la figura 5 es el siguiente:

```

1 \begin{figure*}
2 \centering
3 \begin{gnuplot}[terminal=tikz,
4 terminaloptions={color dashed
5 providevars a,c}]
6 f(x)=a*x+c
7 fit f(x) 'datos1' u 1:2 via a, c
8 set key box at -4.5, 1.85
9 set key width 1.25
10 set key height 0.5
11 set key spacing 1.26
12 set grid xtics lt 0 ls 0
13 set grid ytics lt 0 ls 0

```

```

14 set key opaque
15 set title "Ejemplo de Gráfico"
16 set xlabel "$x$ label"
17 set ylabel "$y$ label"
18 set yrange [-2:2]
19 plot sin(x) lc rgb "blue" t '$\sin(x)$',
20 x**2 lc rgb "red" t '$x^2$',
21 log(x) lc rgb "orange" t '$\ln(x)$'
22 \end{gnuplot}
23 \caption{Ejemplo de gráfico usando la
24 terminal tikz. La pendiente de la función
25 $f(x)=ax+c$ es $a=\num{\gpgetvar{a}}$,
26 mientras que el intercepto es
27 $c=\num{\gpgetvar{c}}$}
28 \label{pic:tikz}
29 \end{figure*}

```

Código 4: Ejemplo de código para gráfico *eps*

Se puede apreciar en el script que se realizó una regresión lineal y como se mencionó anteriormente, la terminal `tikz` tiene la virtud de exportar variables desde `gnuplot`

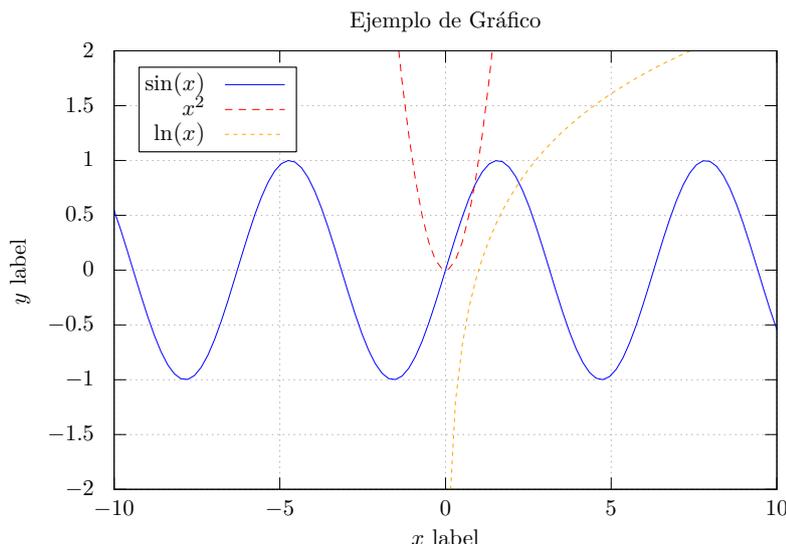


Figura 5: Ejemplo de gráfico usando la terminal `tikz`. La pendiente de la función $(ax + c)$ es $a = 15.96$, mientras que el intercepto es $c = 0.340\,000\,000\,000\,01$.

y una muestra de ello se presenta en el caption de la figura 5 donde se hace el llamado a las constantes calculadas en la regresión lineal, pero no se debe olvidar definir las variables a exportar en `terminaloptions`, así mismo es importante recalcar que esta terminal requiere cargar el paquete `{gnuplot-lua-tikz}` en el preámbulo del documento.

III. SWEAVE

Ahora se procederá a integrar código \LaTeX incluyendo `gnuplottex` con R a través de Sweave.[1]

A. El write18

En muchas ocasiones cuando se trabaja en investigación se hace uso de objetos externos cuando se preparan documentos en \LaTeX estos objetos pueden ser cálculos matemáticos, análisis estadístico, gráficos, entre otros; generalmente éstos se generan con otros programas independientes como ser Gnuplot, R, Phyton, etc. Sería óptimo poder realizar la generación de estos objetos dentro del propio documento, que simplificaría el mantenimiento de los mismos y del propio documento que se está elaborando.

La opción `write18` permite que \LaTeX haga un llamado desde dentro del documento a programas externos para generar objetos que incluirá automáticamente al documento.

Esta opción está deshabilitada por defecto, pero se puede habilitar, ya sea desde la distribución `texlive` o simplemente para un determinado editor de \LaTeX como `Texmaker`, etc.

B. Chunks

La sintaxis para un chunk es la siguiente:

```
<<options>>=
Código R
@
```

El siguiente listado incluye algunas de las opciones que pueden tener cada uno de los chunks utilizados en el documento que se está elaborando. Se enlistará cada uno con la opción por defecto.[1]

- `echo = TRUE` opción lógica que permite incluir el código R en el documento final. Si es `FALSE` no se incluye el código R en el documento final.
- `eval = TRUE`, evalúa el código, cuando es `FALSE`, el código no se evalúa y por lo tanto no produce ningún resultado.
- `results = verbatim` indica que los resultados se escriban en modo literal tipo ordenador o consola. Si es `results=tex` se supone que la salida es ya correcta en \LaTeX y se incluye tal cual. Si se hace `results=hide`, los resultados no se presentan, aunque sí se calculan.
- `split=FALSE` separa o no los chunks en archivos separados. Si es `TRUE` cada uno de los trozos o chunks existentes en el documento se escribe en archivos separados.

C. Integrando \LaTeX , Gnuplot, R

A continuación se presenta un ejemplo de como realizar el procedimiento de integrar \LaTeX , Gnuplot y R en un solo archivo, se aborda el ejemplo por dos caminos, el primero a través de la consola Linux y el segundo por medio del ambiente R Studio.

C.1. Usando la consola

Se debe crear un archivo con extensión `Rnw` como por ejemplo `ejemplo.Rnw`, 6 usando un editor de preferencia

ejemplo.Rnw

```

\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[spanish, es-tabla]{babel}
\usepackage{gnuplottex}
\title {Ejemplo de Sweave}
\author {Ricardo Salgado}
\begin {document}
\maketitle

En este ejemplo se muestra como integrar \LaTeX, R y Gnuplot por medio de
Sweave y gnuplottex.\\\
\textbf{Un gráfico con Gnuplot}

\begin{gnuplot}[terminal = cairolatex]
set term cairolatex header "\\scriptsize"
set title "Ejemplo de Gráfico"
set xlabel "$x$ label"
set ylabel "$y$ label"
set grid
set yrange[-2:2]
set key left top
plot sin(x) lc rgb "blue" t '$\sin(x)$', log(x) lc rgb "orange" t '$\ln (x)$'
\end{gnuplot}

\noindent \textbf{Una tabla a la cual se le realizó análisis estadístico
con código R}

<<echo = FALSE, results = tex>>=
angle <- c(NA, NA, 7, NA, NA, NA)
Long <- c(2.650, 2.651,2.651,2.650,2.650,2.651)
deltaL <- c(NA, NA, 0.001, NA, NA, NA)
tabla <- data.frame(angle, Long, deltaL)
m1 <- c(NA, NA, mean(tabla[1:6,2]), NA, NA, NA)
tabla[, "Mejor_valor_LO"] <- m1
ErrEst1 <- c(NA, NA, sd(tabla[1:6,2])/sqrt(6), NA, NA, NA)
tabla[, "sigma"] <- signif(ErrEst1, 1)
tabla[, "Delta L"]=signif(sqrt(tabla[, 3]**2 + tabla[, "sigma"]**2), 1)
library(xtable)
xtable(tabla, caption = "Análisis de Longitud", digits = c(1,3,3,3,3,4,3))
@

\end{document}

```

Figura 6: Ejemplo de archivo Sweave con extensión Rnw

```

ejemplo.tex
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[spanish, es-tabla]{babel}
\usepackage{gnuplottex}
\title {Ejemplo de Sweave}
\author {Ricardo Salgado}
\begin {document}
\maketitle

En este ejemplo se muestra como integrar \LaTeX, R y Gnuplot por medio de
Sweave y gnuplottex.\\\
\textbf{Un gráfico con Gnuplot}

\begin{gnuplot}[terminal = cairolatex]
set term cairolatex header "\\scriptsize"
set title "Ejemplo de Gráfico"
set xlabel "$x$ label"
set ylabel "$y$ label"
set grid
set yrange[-2:2]
set key left top
plot sin(x) lc rgb "blue" t '$\sin(x)$', log(x) lc rgb "orange" t '$\ln (x)$'
\end{gnuplot}

\noindent \textbf{Una tabla a la cual se le realizó análisis estadístico con
código R}

% latex table generated in R 3.2.2 by xtable 1.7-4 package
% Fri Oct 16 15:04:06 2015
\begin{table}[ht]
\centering
\begin{tabular}{rrrrrr}
\hline
& angle & Long & deltaL & Mejor\_Valor\_L0 & sigma & Delta L \\
\hline
1 & & 2.650 & & & & \\
2 & & 2.651 & & & & \\
3 & 7.000 & 2.651 & 0.001 & 2.651 & 0.0002 & 0.001 \\
4 & & 2.650 & & & & \\
5 & & 2.650 & & & & \\
6 & & 2.651 & & & & \\
\hline
\end{tabular}
\caption{Análisis de Longitud}
\end{table}
\end{document}

```

Figura 7: Resultado de aplicar Sweave, archivo *tex*

(nano, vi, etc), este archivo contendrá en el preámbulo el mismo código L^AT_EX que se incluye en un documento con extensión *tex*. Es decir la clase `\documentclass[]{}{}` y los paquetes `\usepackage[]{}{}`, etc. (no olvidar el paquete `gnuplottex` para trabajar con `gnuplot`). Cada vez que requiera agregar código R se deberá hacer uso de los chunks con la sintaxis apropiada mencionada anteriormente. Para trabajar con `gnuplot` se deberá usar el entorno mostrado en la sección II. Cuando el documento este listo se aplica `Sweave` al archivo con extensión *Rnw* dentro de R usando la siguiente sintaxis:

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

y el resultado en la consola de R debería verse como el mostrado a continuación:

```

Writing to file ejemplo.tex
Processing code chunks with
options ...
  1 : keep.source term tex
      (ejemplo.Rnw:28)

```

```

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

```

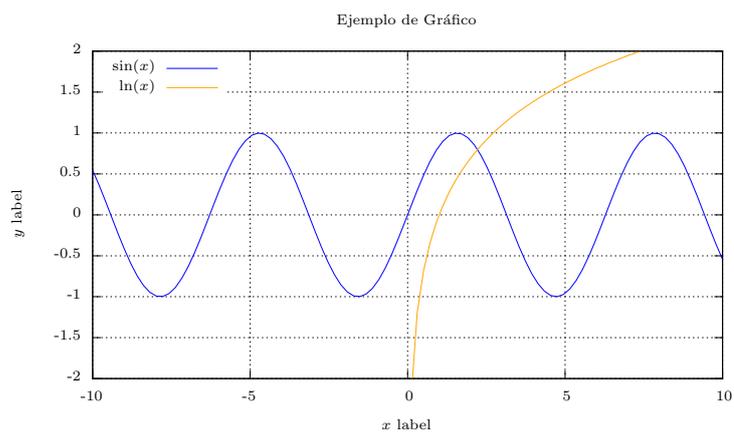
Ejemplo de Sweave

Ricardo Salgado

16 de octubre 2015

En este ejemplo se muestra como integrar \LaTeX , R y Gnuplot por medio de Sweave y gnuplottex.

Un gráfico con Gnuplot



Una tabla a la cual se le realizó análisis estadístico con código R

	angle	Long	deltaL	Mejor_valor_LO	sigma	Delta L
1		2.650				
2		2.651				
3	7.000	2.651	0.001	2.651	0.0002	0.001
4		2.650				
5		2.650				
6		2.651				

Tabla 1: Análisis de Longitud

finalmente se compila el archivo `ejemplo.tex` desde la consola de R haciendo uso de:

```
library(tools)
texi2pdf("ejemplo.tex")
```

para producir el archivo `ejemplo.pdf`. mostrado en la página anterior.

Si solo se desea extraer código R del archivo `.Rnw` se utiliza la opción `Stangle()`:

```
Stangle("ejemplo.Rnw")
```

y se generará un archivo con extensión `.R` en el directorio de trabajo.

C.2. Usando R Studio

- Crear un archivo con extensión `Rnw` dando clic en `file`, `New file`, `RSweave`, Este archivo como se comentó anteriormente contiene todo el código `LATEX`, etc.
- El procedimiento continua siguiendo los mismos pasos mencionados anteriormente en lo que se refiere al ingreso del código R y el código `Gnuplot`.
- Para obtener el archivo en formato `pdf` en R Studio, se debe dar clic en la opción `Compile PDF` y automáticamente se obtiene el documento `pdf` desde el `Rnw` sin necesidad de compilar el archivo `tex`.

C.3. Observaciones Finales

El paquete `\usepackage[nogin]{Sweave}` se debe agregar en el preámbulo del documento `Rnw` para evitar conflicto con figuras muy grandes.

Si R Studio crea el archivo `Rnw`, se agrega la opción `\SweaveOpts{concordance=TRUE}` inmediatamente después de `\begin{document}`, para crear una correspondencia entre los números de los archivos `Rnw` y `tex`, permitiendo tanto la navegación directa a errores de compilación en archivos `Rnw` como la compatibilidad con `SyncTeX` para la vista previa del documento `Pdf`, por lo que se sugiere colocar `concordance=FALSE` cuando hace uso de `Sweave` en la consola.

REFERENCIAS

- [1] FRIEDRICH LEISCH AND R-CORE TEAM (2015). *Sweave User Manual*. R Foundation for Statistical Computing. URL <https://www.statistik.lmu.de/~leisch/Sweave/>.
- [2] LARS KOTTHOFF, UDO HOFEL (2013). *The gnuplottex package*.
- [3] RAMSEY, N. (1998). *Noweb — a simple, extensible tool for literate programming*. URL <https://www.cs.tufts.edu/~nr/noweb/>.