CURSOS DE VERANO 2014

APROXIMACIÓN PRÁCTICA A LA CIENCIA DE DATOS Y BIG DATA: HERRAMIENTAS KNIME, R, HADOOP Y MAHOUT

Introducción al análisis reproducible con R

Francisco Charte Ojeda

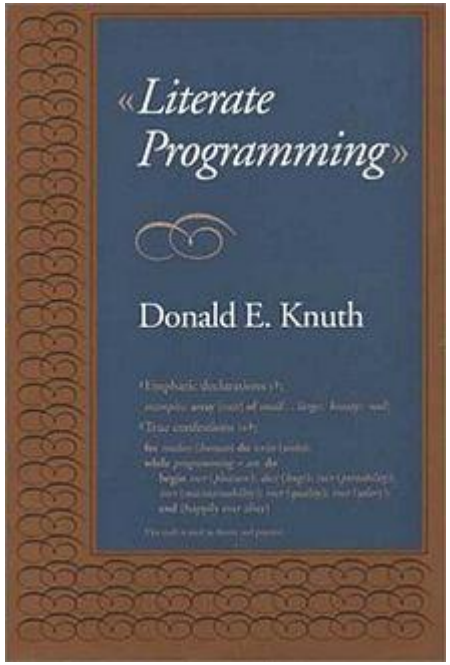
Contenidos

- *Literate programming*
- Investigación reproducible
- Exportación de resultados
- Introducción a Sweave
- Introducción a knitr

Literate programming

- Término acuñado por Donald Knuth en 1984

Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.



Investigación reproducible ► Componentes en un proyecto de investigación

- Datos a analizar - Orígenes heterogéneos (CSV, XLS, ARFF)
  - Integración, transformación, limpieza, etc.
- Algoritmos - Implementación (Java, C/C++, MatLab)
  - Agrupamiento, clasificación, regresión, etc.
- Análisis de resultados - Estadísticos (SPSS, R, Stata)
  - Agregación, comparación, tests estadísticos, etc.
- Representación - Composición (Excel, Gnuplot, R, MatLab)
  - Tablas, gráficas, animaciones, etc.
- Redacción - Documentación (LaTeX, HTML, Word)
  - Descripción de metodología, inclusión de resultados y gráficas

Investigación reproducible ► Problemática

- Demasiadas herramientas en el proceso
  - Distintos lenguajes en implementación de algoritmos, análisis, representación y redacción
- Proceso complejo y específico
  - Cada investigador define un procedimiento a su medida
- Falta de integración y automatización
  - ¿Usada la última versión de los datos tras preprocesamiento?
  - ¿Corresponde la tabla/gráfica a los resultados más recientes?
  - ¿Incorporadas nuevas tablas/gráficas en el documento final?
- Dificultad para reproducir los resultados
  - Altas probabilidades de introducir errores de transcripción

Investigación reproducible ► Solución

- El documento final como contenedor
  - Aloja toda la lógica necesaria para generar el documento
- Automatización
  - Sistema software que garantice la reproducibilidad del proceso
  - Tablas y gráficas producidas programáticamente, no manualmente
- Múltiples salidas
  - Ejecución de los algoritmos de tratamiento de datos
  - Aplicación del análisis a los resultados
  - Producción del documento final en distintos formatos
- Ausencia de errores de transcripción
  - Reproducibilidad de los resultados por parte del lector

Exportación de resultados ► Escenario

- Algoritmos de procesamiento
  - Implementación en C/C++/Java con salida de resultados a CSV
- Análisis de resultados
  - Importación de CSV a R
  - Producción de tablas y gráficas
- Exportación de resultados
  - Generación de tablas en LaTeX/HTML
  - Guardar salidas de tests estadísticos y otros análisis
  - Almacenamiento de gráficas en PDF/PNG
- Elaboración documento final
  - Integración en LaTeX/Web de los resultados exportados

- Paquete `xtable`

```
# Generar una tabla para cada categoría de producto
tabla <- lapply(split(ebay[ ,-1], ebay$Category), xtable)

# Tabla LaTeX estándar
print(tabla$Books[1:5, ], digits=2, include.rownames = FALSE)

# Tabla de tipo longtable
print(tabla$Books[1:5, ], tabular.environment="longtable")

# Tabla HTML
print(tabla$Books[1:5, ], type = 'HTML')
```

- Paquete `Hmisc`

```
# Tabla LaTeX almacenada automáticamente en un archivo
tabla <- latex(ebay[ebay$Category == 'Books',][1:5, ])

readLines(tabla$file)  # Archivo con el resultado
```

Exportación de resultados ► Guardar salidas de tests y análisis estadístico

- Redirección de la salida estándar a un archivo
  - Almacena la salida de la consola de R
  - Necesidad de convertir posteriormente a LaTeX, HTML, etc.
  - Ejemplo:

```
sink('archivo.txt', append=T)       # Salida a archivo.txt
lm(Sepal.Length ~ Sepal.Width, iris) # Modelo lineal
sink()                              # Cierre de la redirección
```

- Útil en otros casos:
  - Guardar el resultado producido por `xtable`
  - Generación automática de *scripts*

Exportación de resultados ► Guardar salidas de tests y análisis estadístico

- Paquetes específicos para la tarea: `texreg`
  - Capaz de producir LaTeX, HTML, Word, etc.
  - Tablas elaboradas comparando varios modelos
  - Ejemplo:

```
modelo1 <- lm(Sepal.Length ~ Sepal.Width, iris)
modelo2 <- lm(Petal.Width ~ Sepal.Width, iris)
texreg(list(modelo1, modelo2),
       custom.model.names=c('Sepal length',
                            'Petal width'))
```

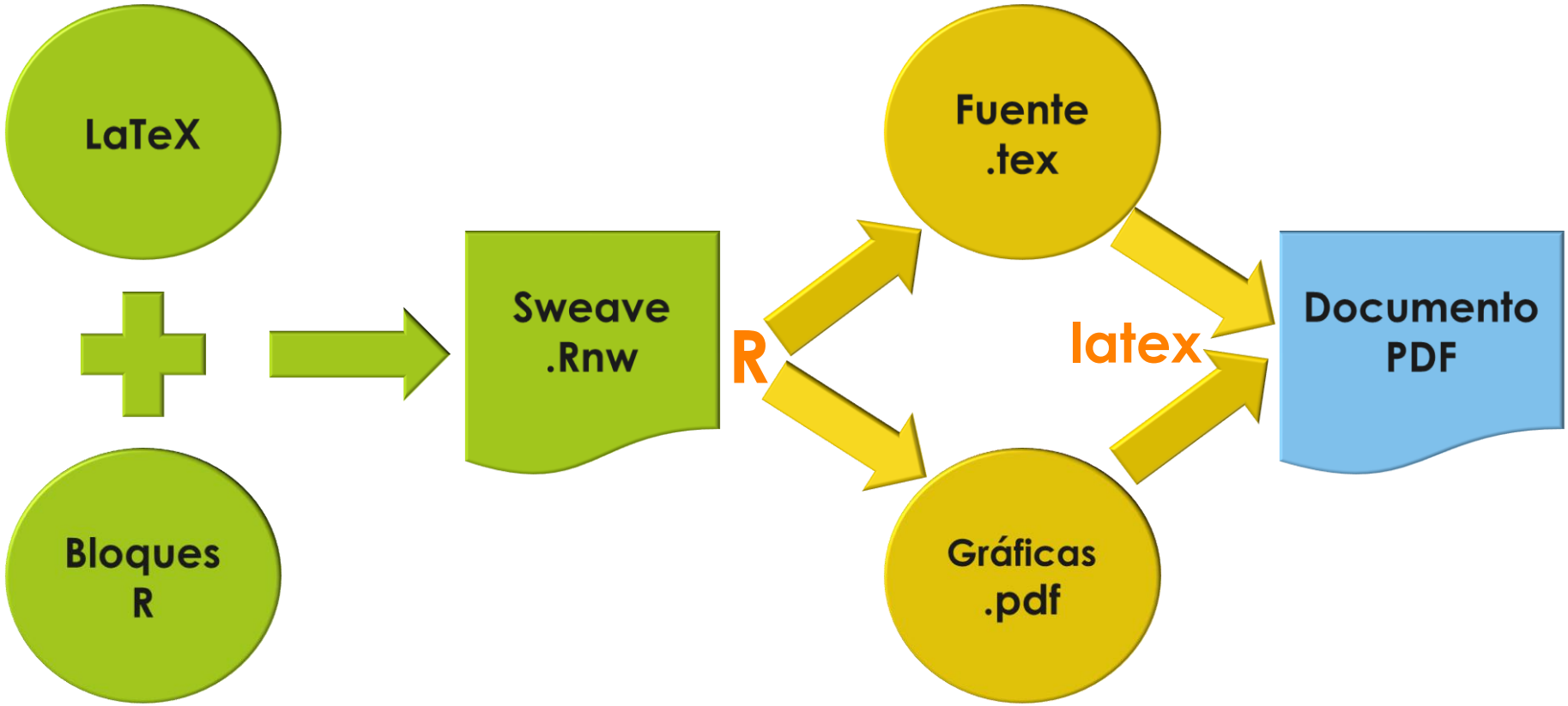
- Otras alternativas:
  - Paquetes `reporttools`, `memisc` y `tables`

Exportación de resultados ► Almacenamiento de gráficas

- Las gráficas han de generarse sin intervención manual
  - Producción programática: `plot`, `qplot`, etc.
- Evitar copiar y pegar imágenes
  - Guardar las gráficas desde R: `pdf`, `tiff`, `png`
  - Importar al documento: `includegraphics`
- Automatización del proceso con un *script* o *makefile*

```
#!/bin/bash
R preparaDatos.r
java -jar ejecutaAlgoritmos
R procesaResultados.r
latex documentoFinal.tex
```

Introducción a Sweave ► LaTeX + R



- El documento .Rnw contendrá LaTeX estándar
  - Cualquier tipo de documento
  - LaTeX sin limitaciones

- Bloques de código R delimitado con `<<opciones>>=` y `@`

```
\section{Listas}

En R las listas\index{listas} pueden contener datos
heterogéneos, incluyendo \textbf{data.frames} y otras
listas

…

<<Listas>>=
lst <- list(3.1415927, 'Hola', TRUE, fdias[4])
lst
unlist(lst)
@
```

Introducción a Sweave ► Pros y contras

- Ventajas
  - Autocontenido y automatizado
    - Documento conteniendo lógica y descripción textual
    - R procesa todo el contenido y genera resultados/figuras
    - Documento siempre actualizado ante cambios en datos
  - Integración en RStudio
- Desventajas
  - Algunas limitaciones en bloques R (sólo un gráfico)
  - Lento para documentos grandes y actualización frecuente
  - No todos los usuarios dispuestos a usar LaTeX para escribir
  - Solamente PDF como salida (posibilidad de usar **pandoc**)

Introducción a knitr ► Markdown + R

- Basado en Sweave
  - Documento contenedor de descripción y código
  - Automatización del proceso de generación del documento
  - Integración con RStudio
- Diferencias
  - Markdown en lugar de LaTeX - Más sencillo de escribir
  - Mayores posibilidades - Documento, presentación
  - Más formatos de salida - LaTeX, HTML, Word, Shiny
  - Mejor rendimiento - Caché de resultados
  - Más flexibilidad - Bloques multi-gráfico

Introducción a knitr ► Sintaxis de Markdown

Fuente Markdown

```
# Encabezado nivel 1

### Encabezado nivel 3

Texto en **negrita** o en *cursiva*

> cita

* Elemento en lista sin ordenar
      + Elemento en nivel 2

1. Elemento en lista ordenada
      + Elemento en nivel 2

Título col1 | Título col2
----------- | -----------
Celda 1     | Celda 2

![](ruta/imagen.png)
```

Resultado obtenido

Encabezado nivel 1

Encabezado nivel 3

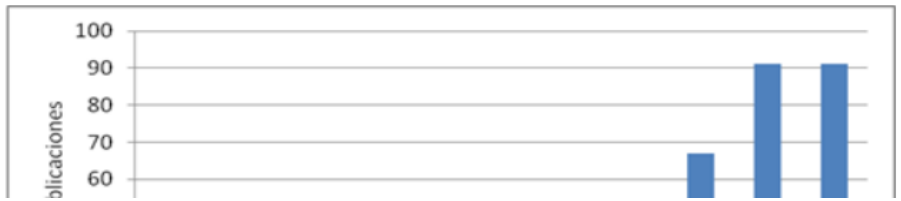
Texto en **negrita** o en *cursiva*

> cita

- Elemento en lista sin ordenar
  - Elemento en nivel 2

1. Elemento en lista ordenada
   - Elemento en nivel 2

Título col1	Título col2
Celda 1	Celda 2



Introducción a knitr ► Bloques de código R y código *inline*

Fuente Markdown

````
# Análisis del dataset *iris*

Correlación entre longitud y
anchura del pétalo con `r
nrow(iris)` muestras

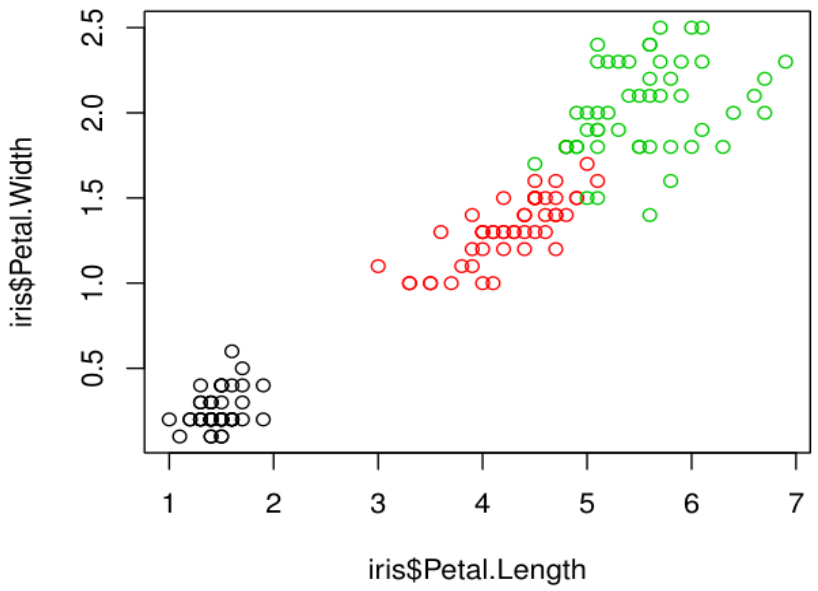
```{r, fig.width=5}
plot(iris$Petal.Length,
     iris$Petal.Width,
     col=iris$Species)
```
````

Resultado obtenido

Análisis del dataset *iris*

Correlación entre longitud y anchura del pétalo con 150 muestras

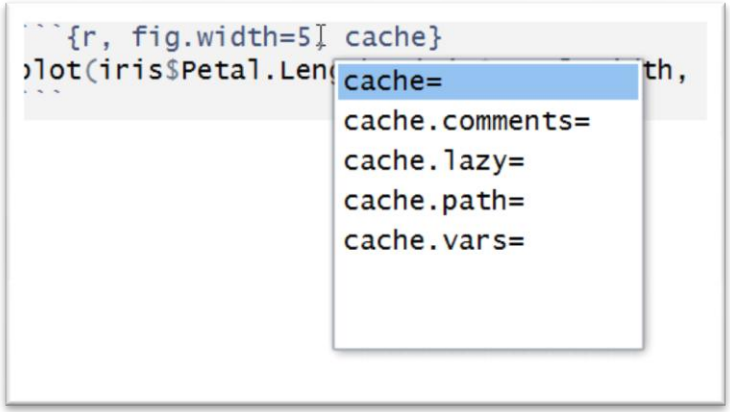
```
plot(iris$Petal.Length, iris$Petal.Width, col=iris$Species)
```



Introducción a knitr ► Más control sobre la salida

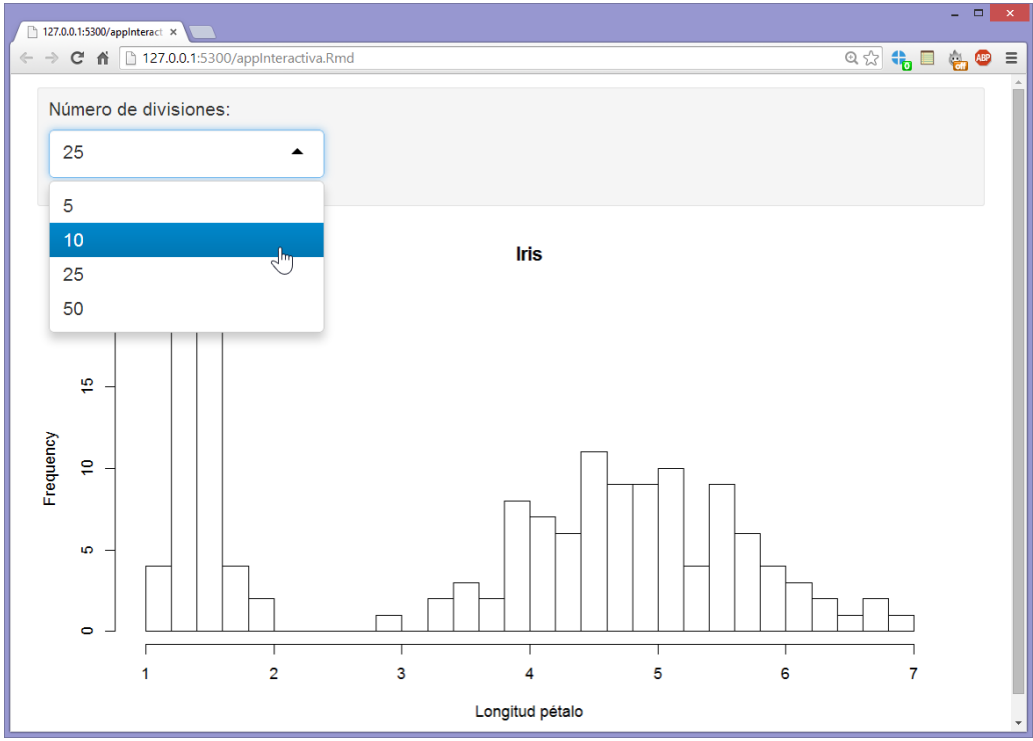
```
# Análisis del dataset *iris*

Correlación entre longitud y anchura del pétalo
```{r,warning=F,message=F,results='asis',}
library('texreg')

texreg(list(lm(Petal.Length ~ Petal.Width, iris)), table = F)
```
```

fig.retina=
fig.scap=
fig.show=
fig.subcap=
fig.width=
highlight=
include=

Análisis del dataset *iris*

Correlación entre longitud y anchura del pétalo

```
library('texreg')

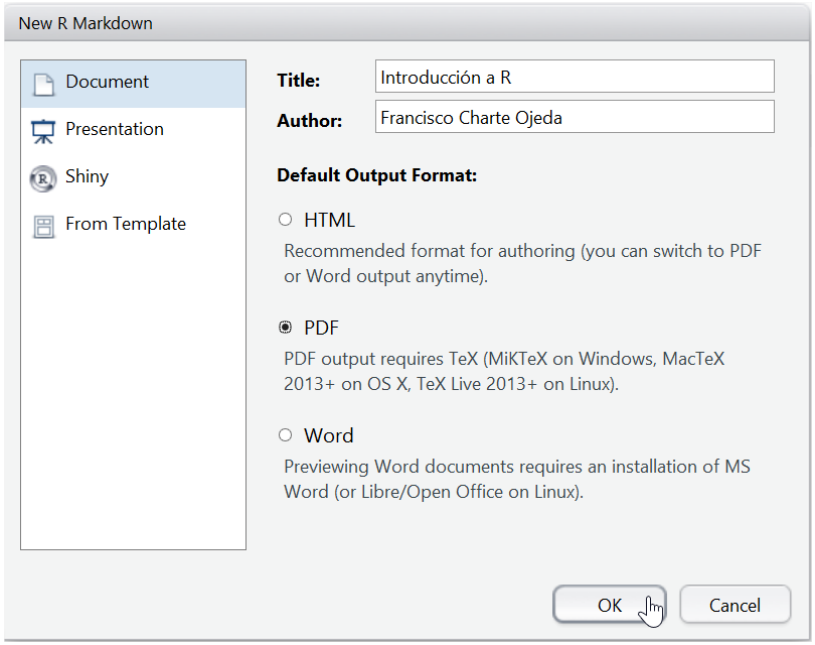
texreg(list(lm(Petal.Length ~ Petal.Width, iris)), table = F)
```

	Model 1
(Intercept)	1.08***
	(0.07)
Petal.Width	2.23***
	(0.05)
R^2	0.93
Adj. R^2	0.93
Num. obs.	150

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Introducción a knitr ► Mejor rendimiento

- Almacenamiento en cache de resultados
- Control de cache global y para cada bloque de código
- Posibilidad de procesar documentos más extensos



Introducción a knitr ► Creación de aplicaciones web interactivas

```
```{r, echo=FALSE}
inputPanel(
  selectInput("divisiones",
  label = "Número de divisiones:",
  choices = c(5, 10, 25, 50),
  selected = 10)
)

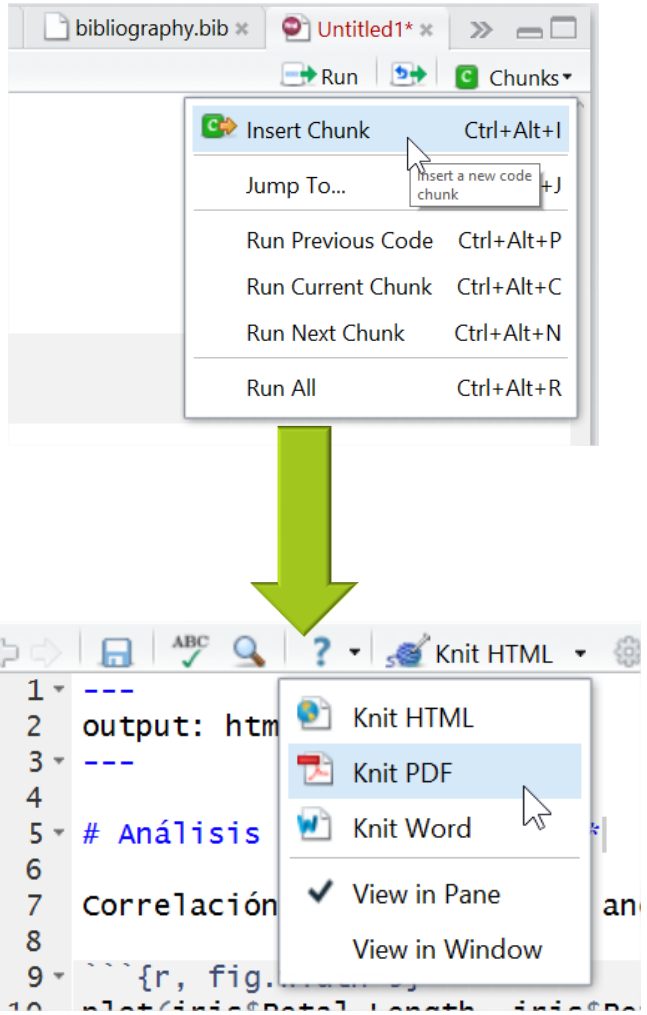
renderPlot({
  hist(iris$Petal.Length,
  breaks=
    as.numeric(input$divisiones),
  main='Iris',
  xlab = "Longitud pétalo")
})

```
```



Introducción a knitr ► Integración en RStudio





CURSOS DE VERANO 2014

APROXIMACIÓN PRÁCTICA A LA CIENCIA DE DATOS Y BIG DATA: HERRAMIENTAS KNIME, R, HADOOP Y MAHOUT

Introducción al análisis reproducible con R

Francisco Charte Ojeda