



UNIVERSITAS

Miguel Hernández

**ANÁLISIS ECONÓMICO DE
SERIES TEMPORALES**

Tema 01.02. “**Utilizando Rstudio para algo más**”

1465-Análisis estadístico de series económicas

Grado en Estadística empresarial

Profesor: Xavier Barber i Vallés

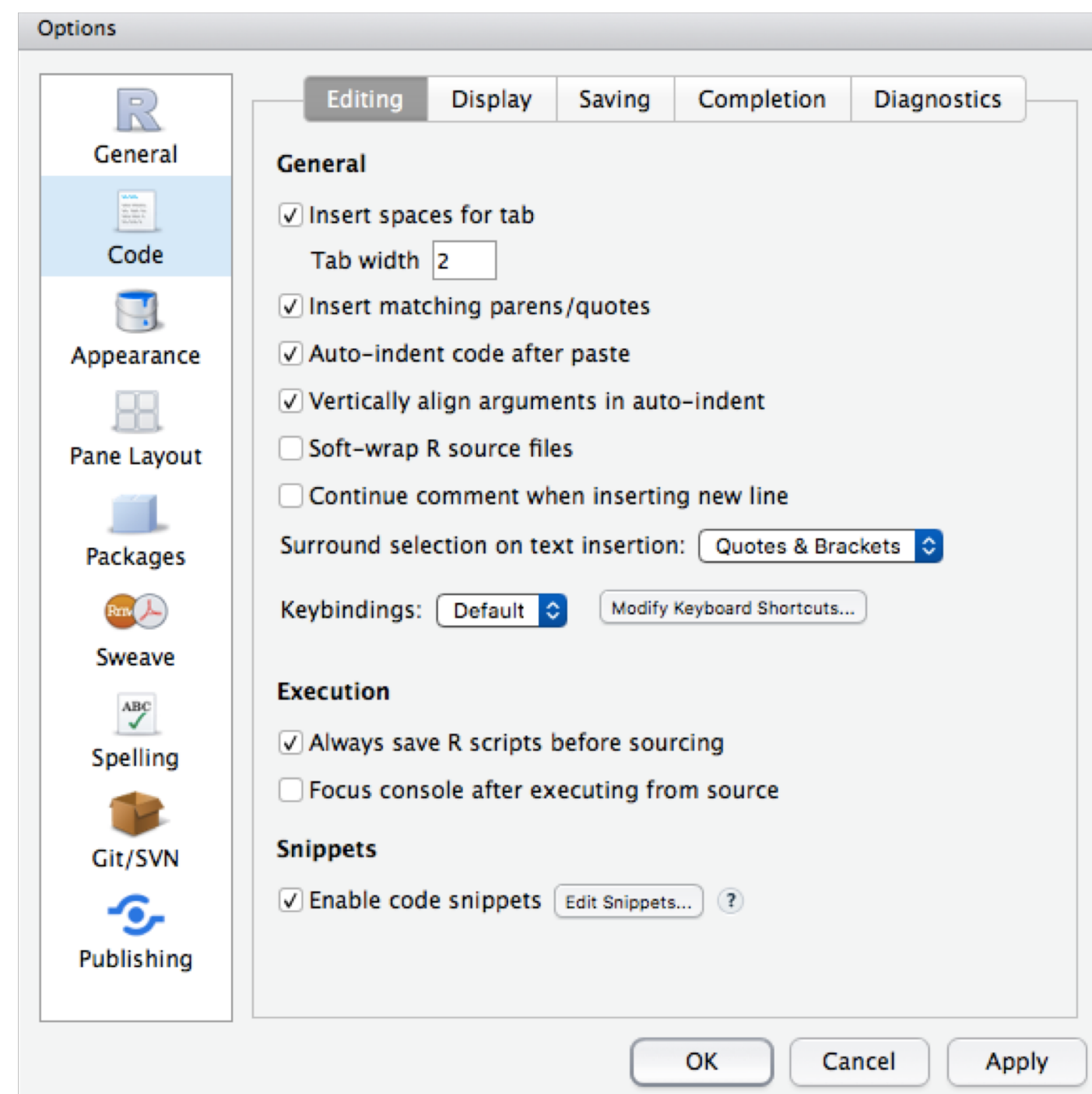
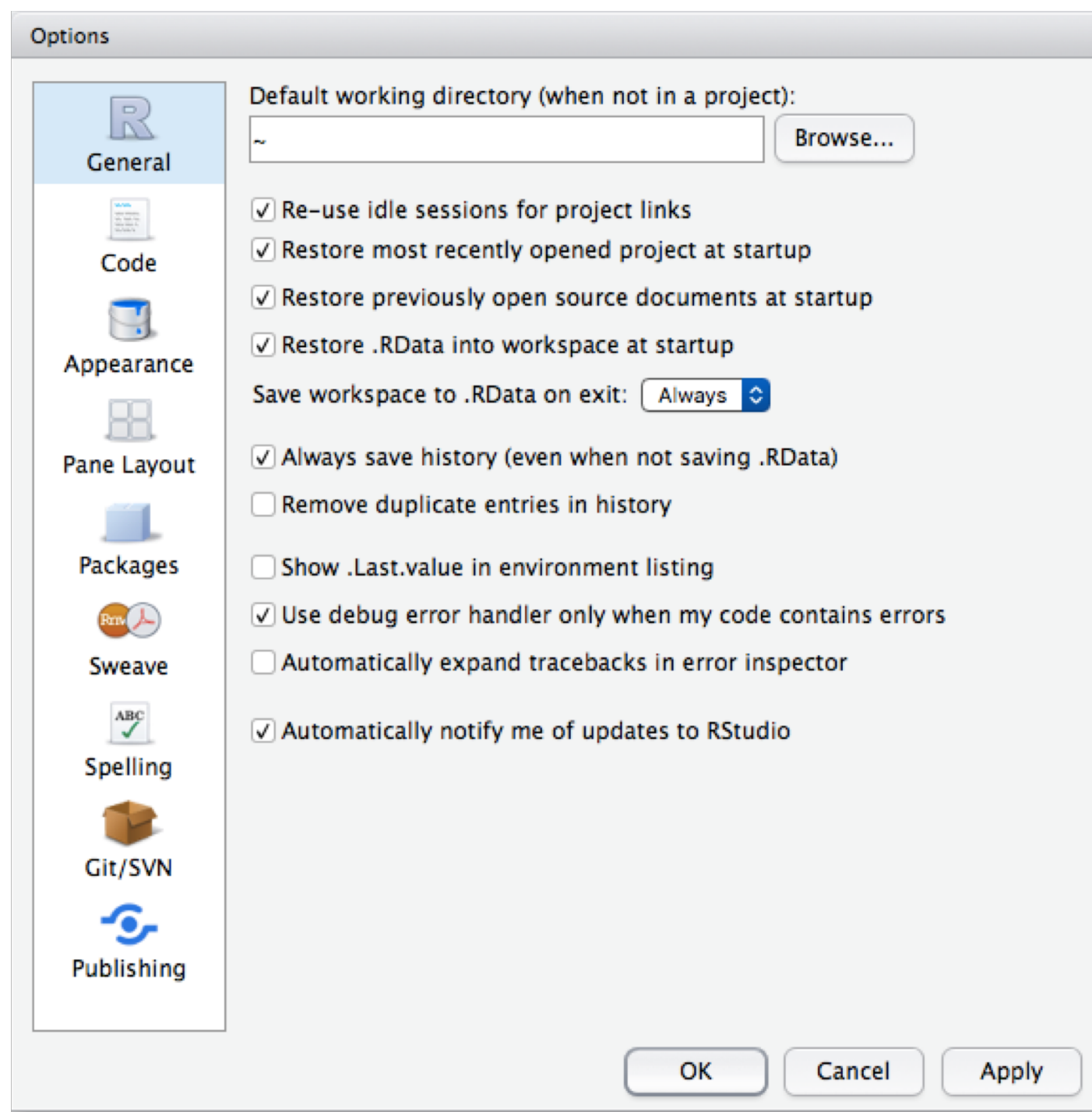
Departamento: Estadística, Matemáticas e Informática

R / RStudio

- **R:** Lenguaje de programación orientado a análisis de datos
- **RStudio:** GUI para R
- Probablemente una de las más potentes herramientas para el análisis de datos

Rstudio: el gigante dormido

Opciones de RStudio



Tipos de Documentos

- En RStudio podemos hacer más cosas a parte de crear un archivo *.R con una sintaxis y ejecutarla.
- Podemos:
 - Crear un proyecto y olvidarnos de dónde guardamos las cosas
 - Generar salidas de los resultados de forma atractiva
 - HTML5, LATEX, WORD, etc.

Opciones de RStudio

The image displays two screenshots of the RStudio Options dialog box, illustrating different configuration options.

Left Screenshot: Appearance Tab

- Editor font:** CourierStd
- Font size:** 12
- Editor theme:** A list of themes is shown, with 'Cobalt' selected. Other themes include Ambiance, Chaos, Chrome, Clouds Midnight, Clouds, Crimson Editor, Dawn, Dreamweaver, Eclipse, Idle Fingers, Katzenmilch, Kr Theme, Merbivore Soft, Merbivore, Mono Industrial, Monokai, Pastel On Dark, Solarized Dark, Solarized Light, and TextMate.
- Code Editor Preview:**

```
# plotting of R objects
plot <- function(x, y, ...)
{
  if (is.function(x) &&
      is.null(attr(x, "class")))
  {
    if (missing(y))
      y <- NULL

    # check for ylab argument
    hasylab <- function(...)
    {
      !all(is.na(
        pmatch(names(list(.
          "ylab"))))

    if (hasylab(...))
      plot.function(x, y, ...)

    else
      plot.function(
        x, y,
        ylab = paste(
          deparse(substitute(
            "(x)"),
            ...))
  }
  else
```

Right Screenshot: Pane Layout Tab

Choose the layout of the panes in RStudio by selecting from the controls in each quadrant.

- Source:** Source
- Console:** Console
- Environment, History, Build, VC:** Environment, History, Build, VC (checked)
- Files, Plots, Packages, Help, Viewer:** Files, Plots, Packages, Help, Viewer (checked)

Buttons: OK, Cancel, Apply

Opcior

Options

General

Code

Appearance

Pane Layout

Packages

Sweave

Spelling

Git/SVN

Publishing

Program defaults (when not in a project)

Weave Rnw files using Sweave knitr ?

Typeset LaTeX into PDF using: pdfLaTeX ?

NOTE: The Rnw weave and LaTeX compilation options are also set on a per-project (and optionally per-file) basis. Click the help icons above for more details.

LaTeX editing and compilation

Clean auxiliary output after compile

Enable shell escape commands

Insert numbered sections and subsections

PDF preview

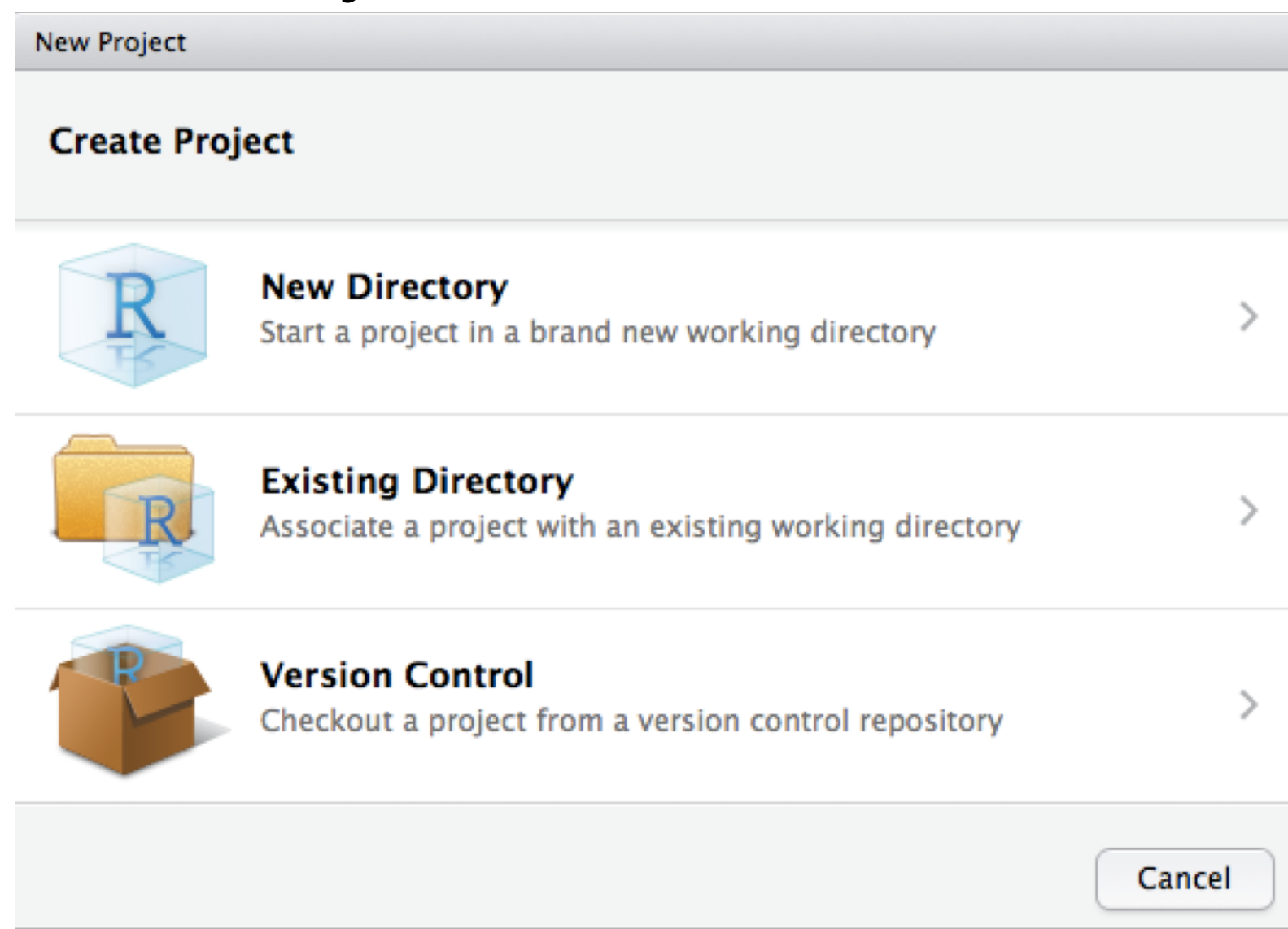
Preview PDF after compile using: RStudio Viewer ?

Always enable Rnw concordance (required for syntex)

OK Cancel Apply

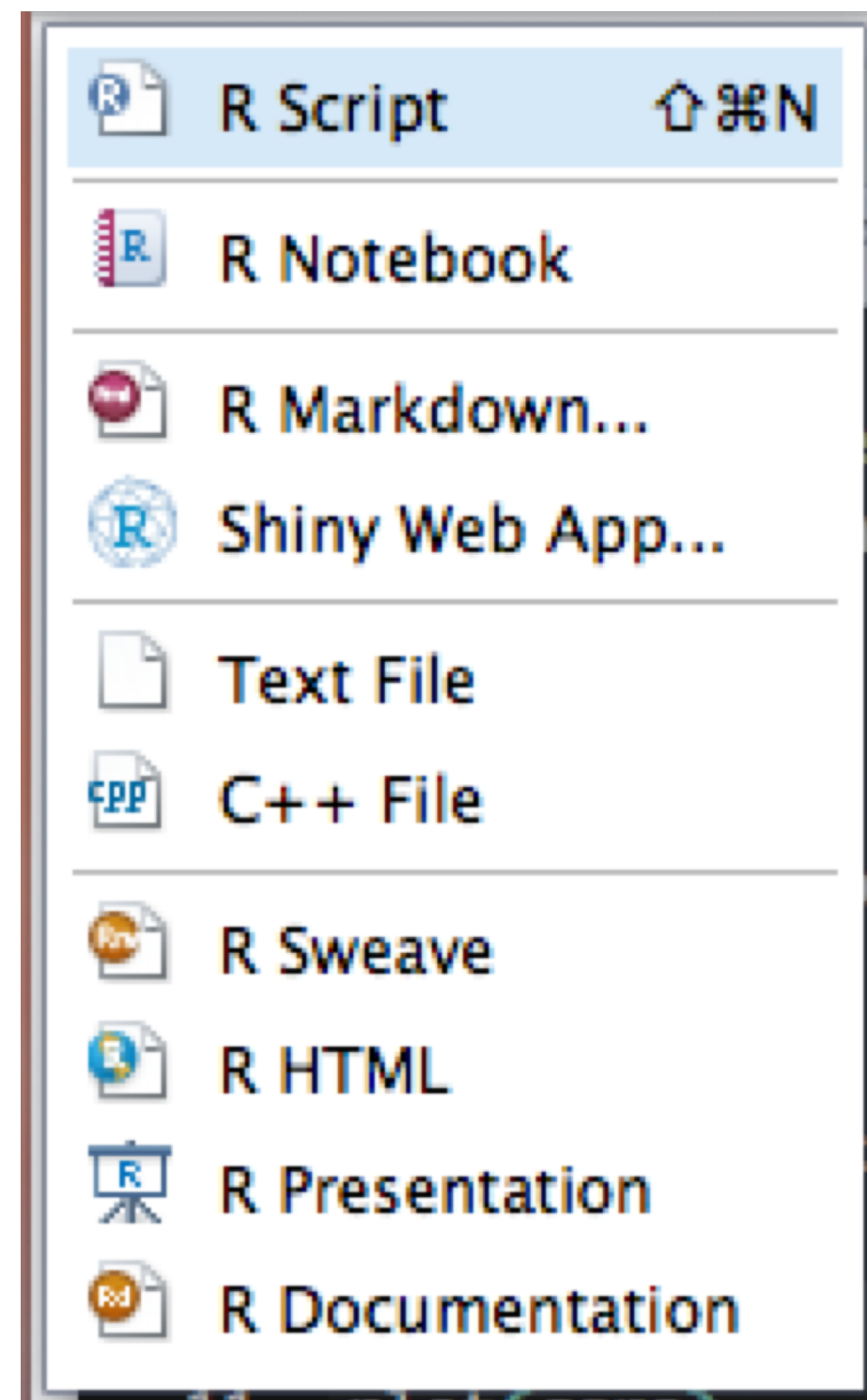
Creando un Nuevo Proyecto

- Archivo → Nuevo Proyecto

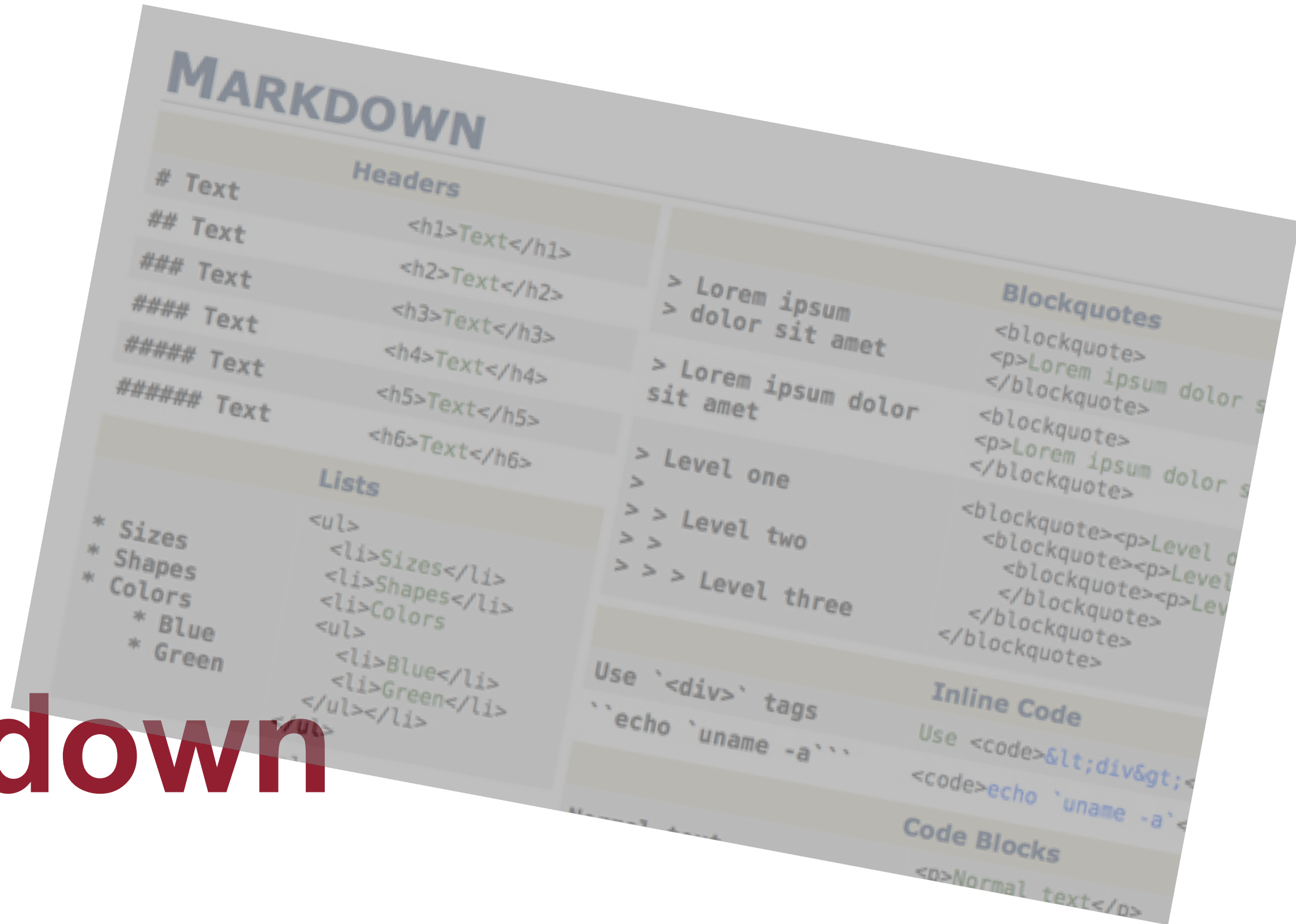


Tipo de Documento a Crear

- Sintaxis o Código de R
- R Notebook
- Documento tipo Markdown
- Aplicación Web de Shiny
- Archivo de Texto
- Sintaxis o código de C++
- Documento tipo Swave
- HTML
- Una presentación de R
- Un archivo de Documentacion de R



markdown



Markdown

- Markdown es un lenguaje de marcado que **facilita la aplicación de formato** a un texto empleando una serie de caracteres de una forma especial.
- En principio, fue pensado para elaborar textos cuyo destino iba a ser la web con más rapidez y sencillez que si estuviésemos empleando directamente HTML.

Markdown

- En este enlace podréis encontrar un buen resumen de cómo funciona Markdown en Rstudio: [Enlace](#).

R Markdown Hoja de Referencia
lee mas en rmarkdown.rstudio.com
rmarkdown 0.2.50 Actualizado: 8/14


1. Flujo de trabajo R Markdown para escribir informes reproducibles y dinámicos con R. Usalo para incluir código R y resultados en presentaciones, documentos pdf, html, Word y mas. Para crear un informe:

i. **Abre** - Abre un archivo con extensión .Rmd. ii. **Escribe** - Escribe contenido con la sintaxis R Markdown. iii. **Incluye** - Incluye código R para crear salidas en el informe iv. **Genera** - Reemplaza el código R con sus salidas y transforma el informe a una presentación, PDF, HTML o Word.



2. Abre archivo Empieza guardando un archivo de text con la extensión .Rmd o abre RStudio.

3. Markdown Ahora, escribe tu informe en texto plano. usa la sintaxis de markdown para describir el formato de tu texto en el informe final.

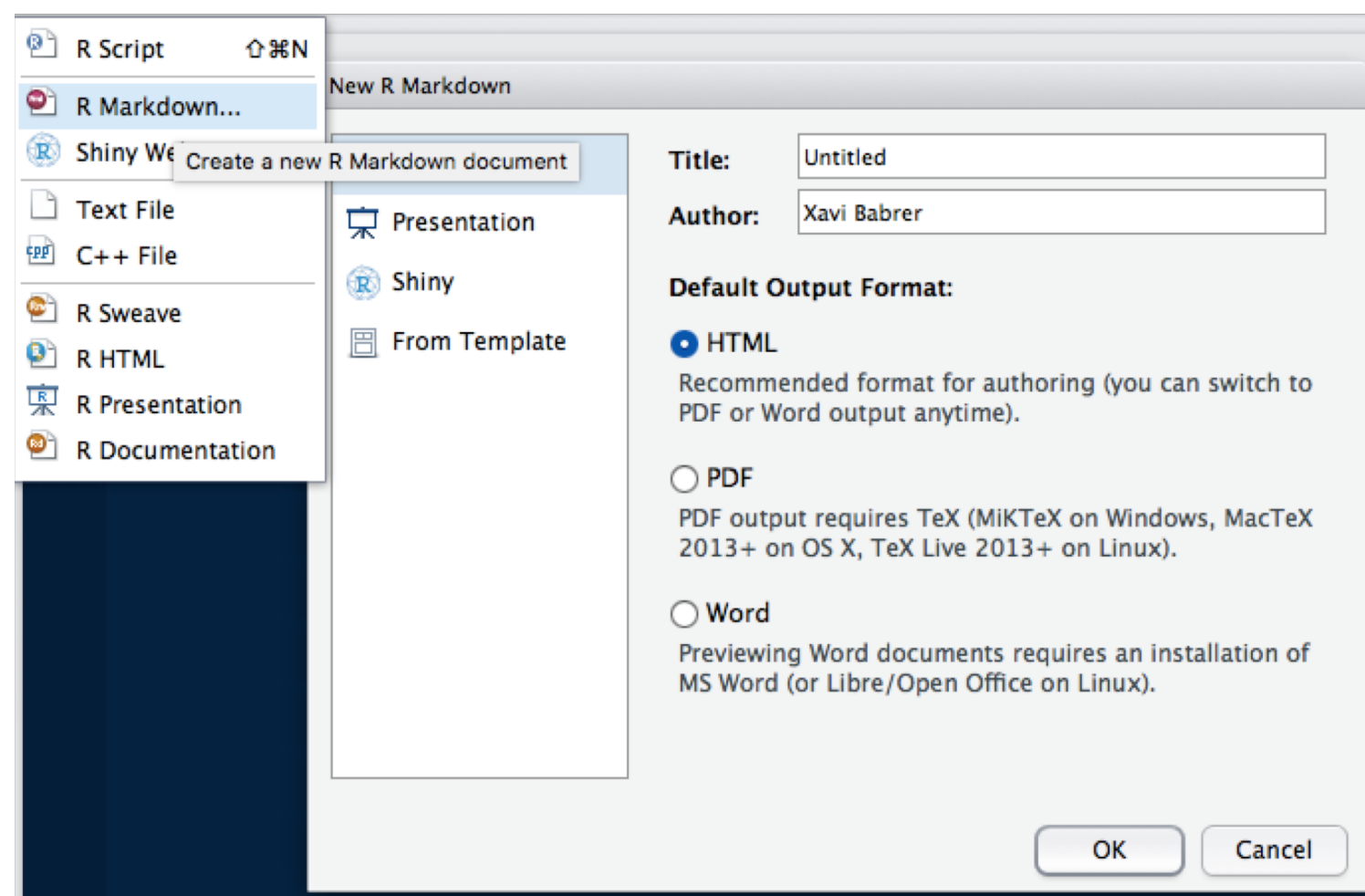
Markdown Ejemplo (I)

```
install.packages("rmarkdown")
```

```
install.packages("knitr")
```

```
library(rmarkdown)
```

```
library(knitr)
```



Markdown Example (I)

```
1 ---
2 title: "Untitled"
3 author: "Xavi Babrer"
4 date: "10 de febrero de 2016"
5 output: word_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
11
```

```
17
18 ```{r cars}
19 summary(cars)
20 ```
21
22 ## Including Plots
23
24 You can also embed plots, for example:
25
26 ```{r pressure, echo=FALSE}
27 plot(pressure)
28 ```
29
30 Note that the `echo = FALSE` parameter was used to
code that generated the plot.
```

Markdown Example (I)

002_markdown_example_1.docx (1 página)

Buscar

Untitled

Xavi Babrer
10 de febrero de 2016

R Markdown

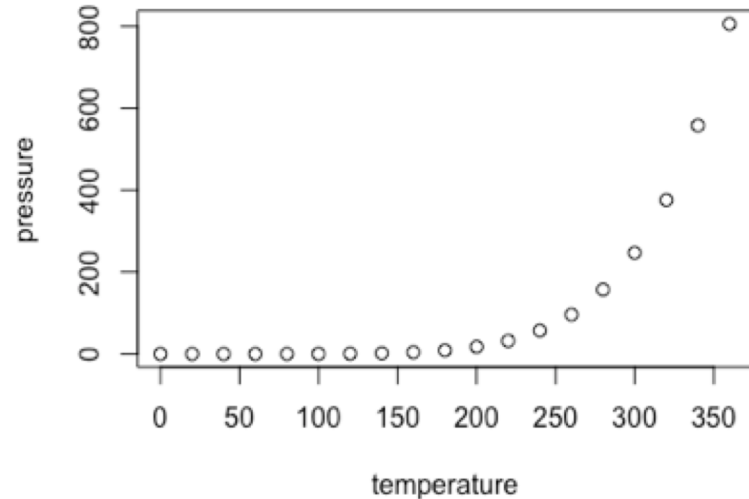
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
##      speed      dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



A scatter plot showing the relationship between temperature (x-axis) and pressure (y-axis). The x-axis ranges from 0 to 350 with major ticks every 50 units. The y-axis ranges from 0 to 800 with major ticks every 200 units. The data points show a clear upward trend, starting near zero pressure for low temperatures and increasing exponentially as temperature rises, reaching approximately 800 pressure at 350 temperature.

temperature	pressure
0	0
50	0
100	0
150	0
200	0
220	50
240	100
260	150
280	200
300	300
320	400
340	600
350	800

Markdown Example (II)

- `eval` TRUE Whether to evaluate the code and include its results
- `echo` TRUE Whether to display code along with its results
- `warning` TRUE Whether to display warnings
- `error` FALSE Whether to display errors
- `message` TRUE Whether to display messages
- `tidy` FALSE Whether to reformat code in a tidy way when displaying it
- `results` "markup" "markup", "asis", "hold", or "hide"
- `cache` FALSE Whether to cache results for future renders
- `comment` "###" Comment character to preface results with
- `fig.width` 7 Width in inches for plots created in chunk
- `fig.height` 7 Height in inches for plots created in chunk

Markdown Example (II)

```
## creando dos variables X e Y
```

```
```{r }
```

```
x <- 1:10
```

```
y <- round(rnorm(10, x, 1), 2)
```

```
df <- data.frame(x, y)
```

```
df```
```

```
y ahora la salida de esto:
```

```
```{r , results='asis', echo=FALSE}
```

```
cat("Here are some dot points\n\n")
```

```
cat(paste("* The value of y[", 1:3, "] is ", y[1:3],  
        sep="", collapse="\n"))```
```


Markdown Example (II)

```
## creando dos variables X e Y

```{r }
x <- 1:10
y <- round(rnorm(10, x, 1), 2)
df <- data.frame(x, y)
df```

y ahora la salida de esto:
```{r , results='asis', echo=FALSE}
cat("Here are some dot points\n\n")
cat(paste("* The value of y[", 1:3, "] is ", y[1:3],
        sep="", collapse="\n"))```
```

Creando dos variables X e Y

```
x <- 1:10
y <- round(rnorm(10, x, 1), 2)
df <- data.frame(x, y)
df
##      x      y
## 1     1  1.79
## 2     2  1.45
## 3     3  3.07
## 4     4  3.72
## 5     5  4.58
## 6     6  6.56
## 7     7  7.32
## 8     8  7.38
## 9     9  9.05
## 10    10 10.11
```

y ahora la salida de esto:

Here are some dot points

The value of y[1] is 1.79

The value of y[2] is 1.45

The value of y[3] is 3.07

Markdown Example (III)

```
## Una tabla
```{r , results='asis', echo=FALSE}

cat("x | y", "--- | ---", sep="\n")

cat(apply(df, 1, function(X) paste(X,
collapse=" | ")), sep = "\n")```
```

## Una tabla

<u>x</u>	<u>y</u>
1	1.8
2	1.01
3	2.18
4	6.27
5	5.4
6	4.37
7	6.33
8	9.1
9	8.46
10	9.85

# Markdown Example (iv)

Symbol	Meaning	Example
-----	-----	-----
%d	day as a number (0-31)	01-31
%a	abbreviated weekday	Mon
%A	unabbreviated weekday	Monday
%m	month (00-12)	00-12
%b	abbreviated month	Jan
%B	unabbreviated mont	January
%y	2-digit year	07
%Y	4-digit year	2007

# Markdown Example (iv)

Symbol	Meaning	Example
%d	day as a number (0-31)	01-31
%a	abbreviated weekday	Mon
%A	unabbreviated weekday	Monday
%m	month (00-12)	00-12
%b	abbreviated month	Jan
%B	unabbreviated month	January
%y	2-digit year	07
%Y	4-digit year	2007



# Shiny:

creando aplicaciones web desde rstudio

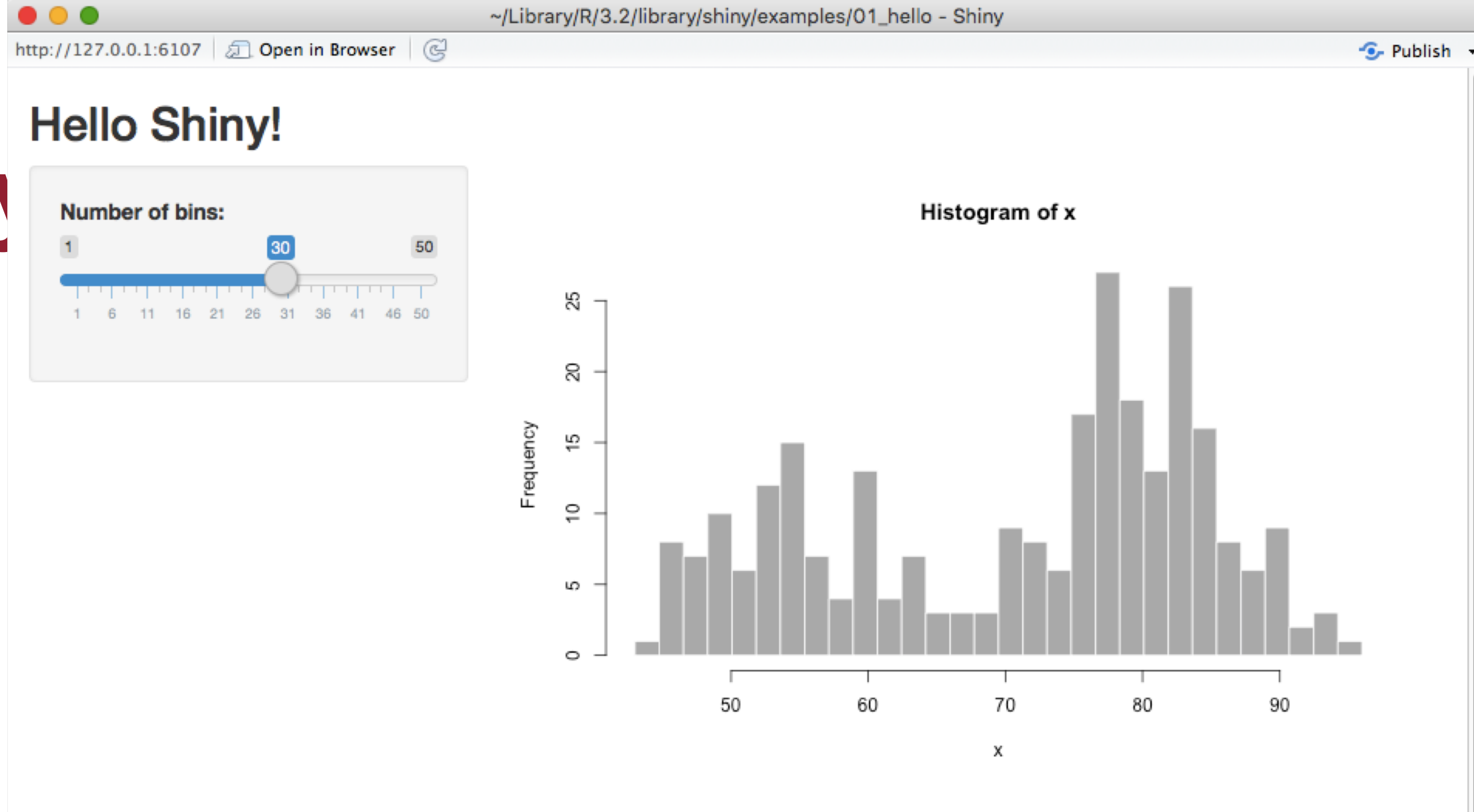
# ¿Qué es Shiny?

**Crea aplicaciones web interactivas (apps) desde R.**

- Shiny es una herramienta para crear fácilmente aplicaciones web interactivas (apps) que permiten a los usuarios interactuar con sus datos sin tener que manipular el código.

```
install.packages("shiny")
library(shiny)
runExample("01_hello")
```

# Shiny



## Hello Shiny!

by RStudio, Inc.

This small Shiny application demonstrates Shiny's automatic UI updates. Move the *Number of bins* slider and notice how the `renderPlot` expression is automatically re-evaluated when its dependant, `input$bins`, changes, causing a histogram with a new number of bins to be rendered.

server.R

ui.R

↑ show with app

```
library(shiny)

Define server logic required to draw a histogram
shinyServer(function(input, output) {

 # Expression that generates a histogram. The expression is
 # wrapped in a call to renderPlot to indicate that:
 #
 # 1) It is "reactive" and therefore should be automatically
 # re-executed when inputs change
 # 2) Its output type is a plot

 output$distPlot <- renderPlot({
 x <- faithful[, 2] # Old Faithful Geyser data
 bins <- seq(min(x), max(x), length.out = input$bins + 1)

 # draw the histogram with the specified number of bins
 hist(x, breaks = bins, col = 'darkgray', border = 'white')
 })
})
```

# Shiny

- Se necesita un “script” que será la *interface* y otra “script” que será el *server*:
  - Server.R
  - ui.R
- Se puede probar a nivel local, o bien en un repositorio o más a nivel profesional desde un Rstudio Server.



# Shiny: server.R

```
library(shiny)

Define server logic required to draw a histogram
shinyServer(function(input, output) {

 # Expression that generates a histogram. The expression is
 # wrapped in a call to renderPlot to indicate that:
 #
 # 1) It is "reactive" and therefore should be automatically
 # re-executed when inputs change
 # 2) Its output type is a plot

 output$distPlot <- renderPlot({
 x <- faithful[, 2] # Old Faithful Geyser data
 bins <- seq(min(x), max(x), length.out = input$bins + 1)

 # draw the histogram with the specified number of bins
 hist(x, breaks = bins, col = 'darkgray', border = 'white')
 })
})
```

# Shiny: ui.R

```
library(shiny)

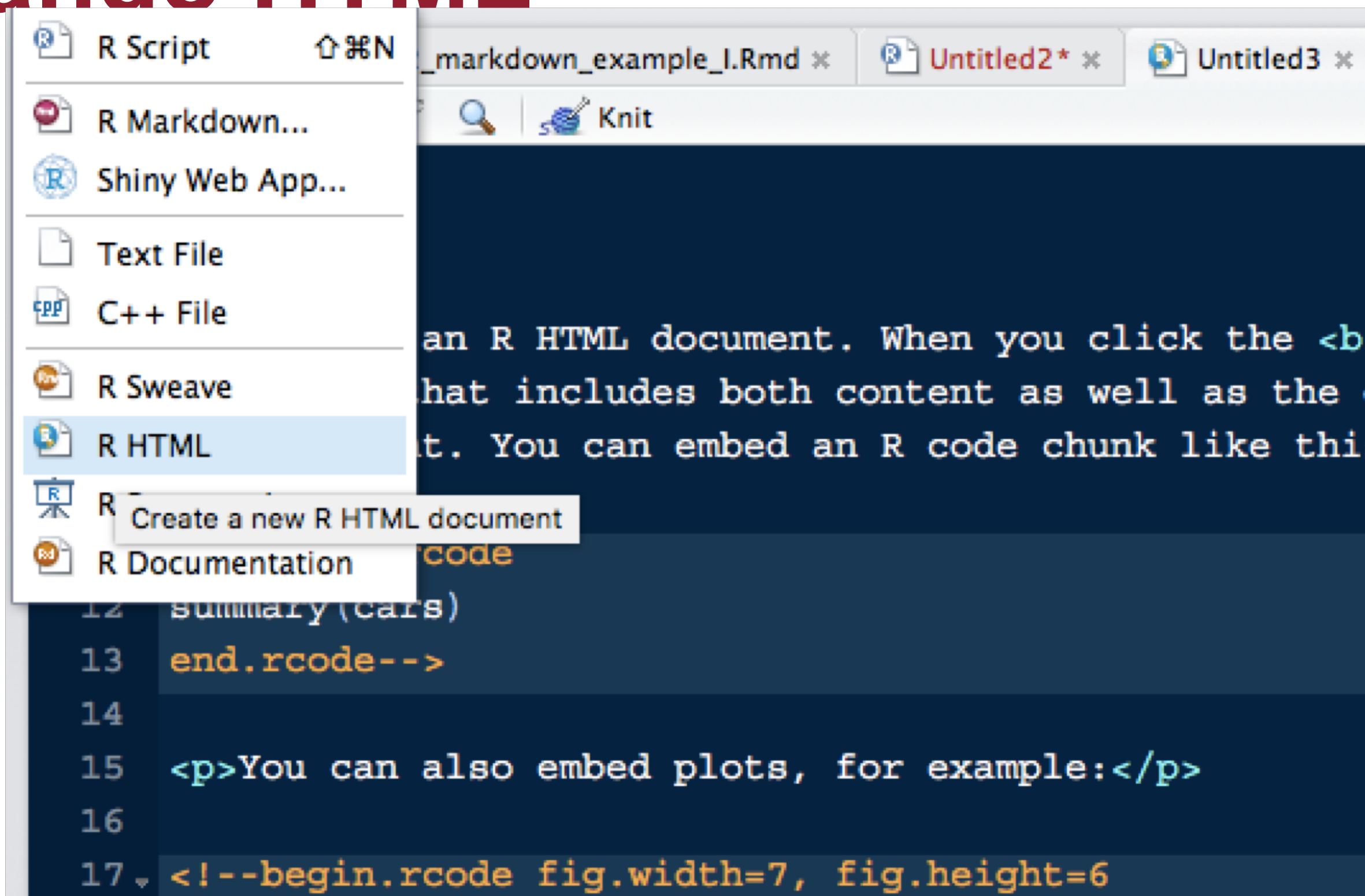
Define UI for application that draws a histogram
shinyUI(fluidPage(

 # Application title
 titlePanel("Hello Shiny!"),

 # Sidebar with a slider input for the number of bins
 sidebarLayout(
 sidebarPanel(
 sliderInput("bins",
 "Number of bins:",
 min = 1,
 max = 50,
 value = 30)
),

 # Show a plot of the generated distribution
 mainPanel(
```

# Creando HTML



# Creando HTML

```
<html>
```

```
<head>
```

```
<title>Title</title>
```

```
</head>
```

```
<body>
```

```
<p>This is an R HTML document. When you click the Knit HTML button a web page will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:</p>
```

```
<!--begin.rcode
```

```
summary(cars)
```

```
end.rcode-->
```

```
<p>You can also embed plots, for example:</p>
```



```
<!--begin.rcode fig.width=7, fig.height=6
```

```
plot(cars)
```

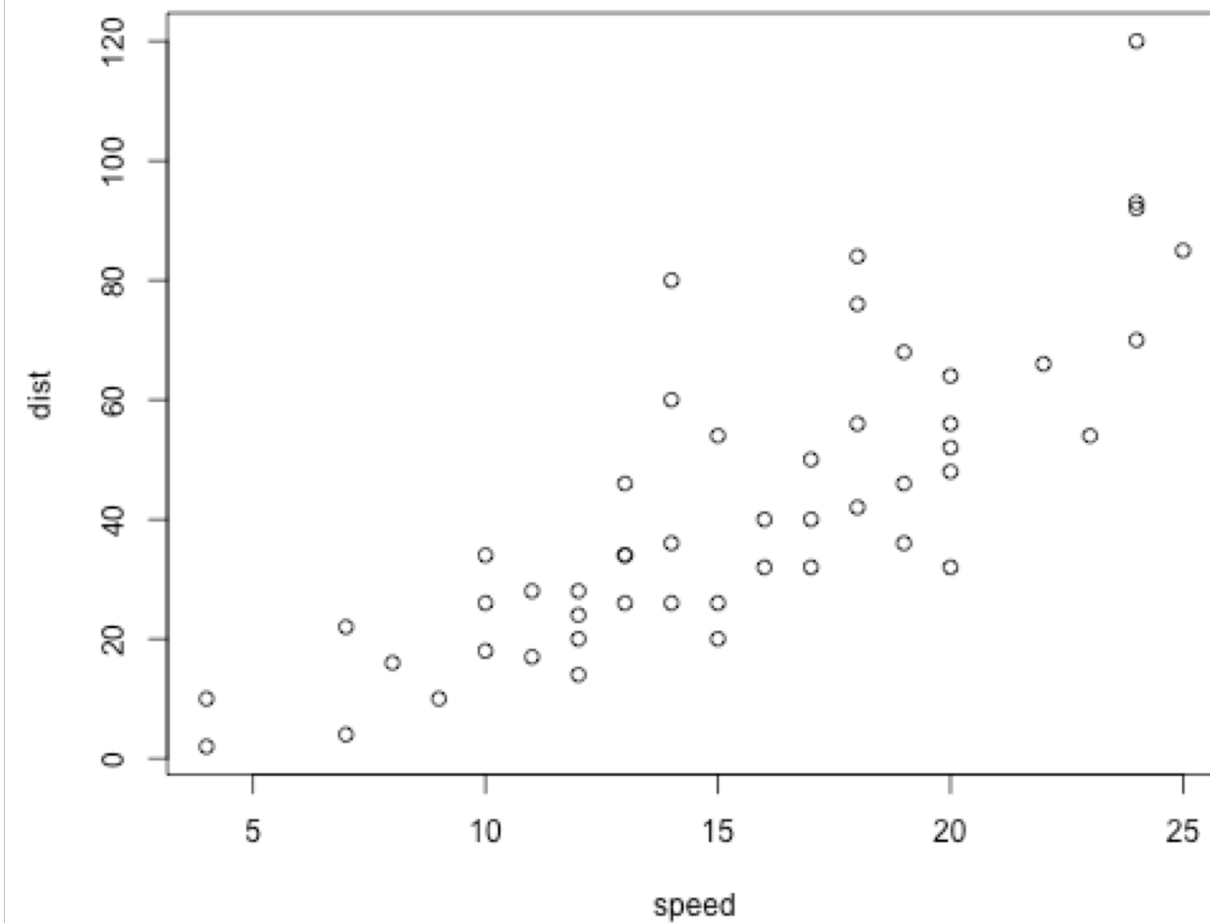
This is an R HTML document. When you click the **Knit HTML** button a web page will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
speed dist
Min. : 4.0 Min. : 2.00
1st Qu.:12.0 1st Qu.: 26.00
Median :15.0 Median : 36.00
Mean :15.4 Mean : 42.98
3rd Qu.:19.0 3rd Qu.: 56.00
Max. :25.0 Max. :120.00
```

You can also embed plots, for example:

```
plot(cars)
```



# R presentation

002\_R\_presentation

=====

author:

date:

autosize: true

First Slide

=====

For more details on authoring R presentations please visit <https://support.rstudio.com/hc/en-us/articles/200486468>.

- Bullet 1
- Bullet 2
- Bullet 3

Slide With Code

=====

```
```{r}
```

```
summary(cars)
```



R presentation

Presentation

First Slide

For more details on
[us/articles/200486](https://www.r-bloggers.com/us/articles/200486)

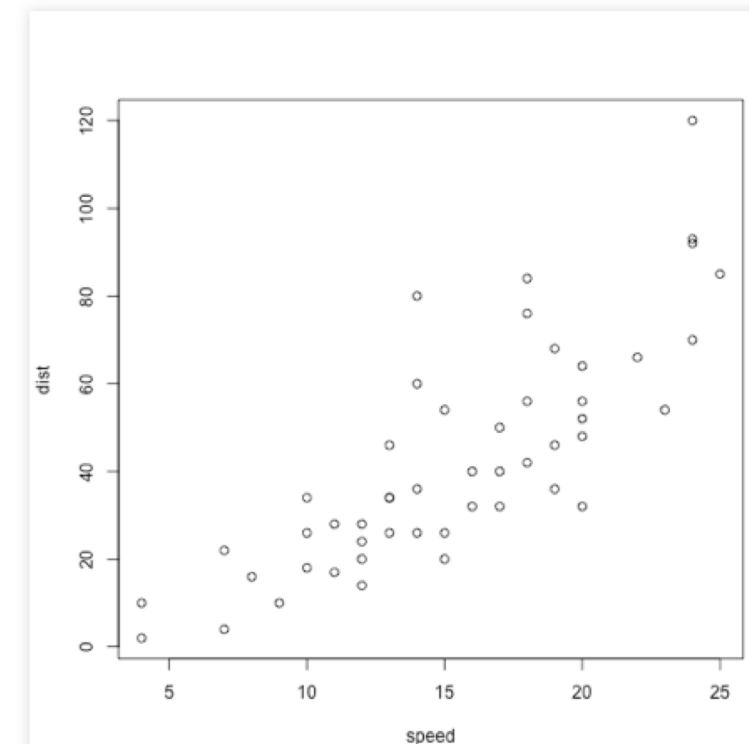
- Bullet 1
- Bullet 2
- Bullet 3

Slide With Code

```
summary(cars)
```

speed		dist	
Min.	: 4.0	Min.	: 2.00
1st Qu.:	12.0	1st Qu.:	26.00
Median	:15.0	Median	: 36.00
Mean	:15.4	Mean	: 42.98
3rd Qu.:	19.0	3rd Qu.:	55.00
Max.	:25.0	Max.	: 120.00

Slide With Plot





UNIVERSITAS

Miguel Hernández