Dynamic reports in R (with LaTeX or HTML)

Manuel Febrero Bande
Universidad de Santiago de Compostela

DEPARTAMENTO DE ESTATÍSTICA
E INVESTIGACIÓN OPERATIVA

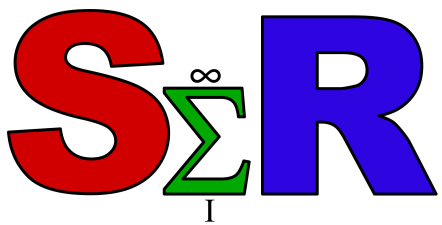


Table of Contents

Índice

Manual reports vs Dynamic reports

Manual report

- Import a dataset into a statistical software.
- Run the procedure to get the results
- Copy and paste selected pieces into your word processor adding your own narratives.

Drawbacks

- It is error-prone due to too much manual work.
- A lot of human effort to do tedious jobs.
- Not always easy to reproduce (changes in style, data)

Dynamic report

- Create a code that put together the code (that automatically reads the data, performs the statistical analyses, create the plots and the tables) and the report.

Drawbacks

- A lot of effort for a single task.

Advantages

- No investments in software licenses.
- Reproducible (research or report).

Manual reports vs Dynamic reports

Manual report

- Import a dataset into a statistical software.
- Run the procedure to get the results
- Copy and paste selected pieces into your word processor adding your own narratives.

Drawbacks

- It is error-prone due to too much manual work.
- A lot of human effort to do tedious jobs.
- Not always easy to reproduce (changes in style, data)

Dynamic report

- Create a code that put together the code (that automatically reads the data, performs the statistical analyses, create the plots and the tables) and the report.

Drawbacks

- A lot of effort for a single task.

Advantages

- No investments in software licenses.
- Reproducible (research or report).

Manual reports vs Dynamic reports

Manual report

- Import a dataset into a statistical software.
- Run the procedure to get the results
- Copy and paste selected pieces into your word processor adding your own narratives.

Drawbacks

- It is error-prone due to too much manual work.
- A lot of human effort to do tedious jobs.
- Not always easy to reproduce (changes in style, data)

Dynamic report

- Create a code that put together the code (that automatically reads the data, performs the statistical analyses, create the plots and the tables) and the report.

Drawbacks

- A lot of effort for a single task.

Advantages

- No investments in software licenses.
- Reproducible (research or report).

Manual reports vs Dynamic reports

Manual report

- Import a dataset into a statistical software.
- Run the procedure to get the results
- Copy and paste selected pieces into your word processor adding your own narratives.

Drawbacks

- It is error-prone due to too much manual work.
- A lot of human effort to do tedious jobs.
- Not always easy to reproduce (changes in style, data)

Dynamic report

- Create a code that put together the code (that automatically reads the data, performs the statistical analyses, create the plots and the tables) and the report.

Drawbacks

- A lot of effort for a single task.

Advantages

- No investments in software licenses.
- Reproducible (research or report).

Manual reports vs Dynamic reports

Manual report

- Import a dataset into a statistical software.
- Run the procedure to get the results
- Copy and paste selected pieces into your word processor adding your own narratives.

Drawbacks

- It is error-prone due to too much manual work.
- A lot of human effort to do tedious jobs.
- Not always easy to reproduce (changes in style, data)

Dynamic report

- Create a code that put together the code (that automatically reads the data, performs the statistical analyses, create the plots and the tables) and the report.

Drawbacks

- A lot of effort for a single task.

Advantages

- No investments in software licenses.
- Reproducible (research or report).

What is needed?

- R language: `http://www.r-project.org`, `http://cran.fiocruz.br/`
- `knitr` package: Package for the literate programming `http://yihui.name/knitr/`. Better than Sweave.
- `rmarkdown` package: Package for the literate programming `http://rmarkdown.rstudio.com/`.
- Optional tools
  - TEX compiler: MikTeX (windows):`http://www.miktex.org`
    MacTeX (MacOS): `http://tug.org/mactex/`
    TEXLive (Linux):`http://tug.org/texlive/`
  - TEX editor: TeXstudio: `http://texstudio.sourceforge.net/`
  - Rstudio: `http://www.rstudio.com`

What is needed?

- R language: `http://www.r-project.org`, `http://cran.fiocruz.br/`
- `knitr` package: Package for the literate programming `http://yihui.name/knitr/`. Better than `Sweave`.
- `rmarkdown` package: Package for the literate programming `http://rmarkdown.rstudio.com/`.
- Optional tools
  - TEX compiler: MikTeX (windows):`http://www.miktex.org` MacTeX (MacOS): `http://tug.org/mactex/` TEXLive (Linux):`http://tug.org/texlive/`
  - TEX editor: TeXstudio: `http://texstudio.sourceforge.net/`
  - Rstudio: `http://www.rstudio.com`

What is needed?

- R language: `http://www.r-project.org`, `http://cran.fiocruz.br/`
- `knitr` package: Package for the literate programming `http://yihui.name/knitr/`. Better than `Sweave`.
- `rmarkdown` package: Package for the literate programming `http://rmarkdown.rstudio.com/`.
- Optional tools
  - TEX compiler: MikTeX (windows):`http://www.miktex.org`
    MacTeX (MacOS): `http://tug.org/mactex/`
    TEXLive (Linux):`http://tug.org/texlive/`
  - TEX editor: TeXstudio: `http://texstudio.sourceforge.net/`
  - Rstudio: `http://www.rstudio.com`

Editors

Depending on the use you can work with simple plain text editors or with more elaborated ones.

- `RStudio`: `http://www.rstudio.com`: Best for beginners. Better for editing chunk codes or for managing Markdown files. Not so good for creating LaTeX documents. See the `Tools|Global Options|Sweave` for configuration.
- LaTeX editors: `TexStudio`, `Texmaker`. Better for documents. It must be personalize to compile `.Rnw` files. (`"path/to/Rbin/Rscript.exe" -e "library(knitr); knit2pdf('%.Rnw')"`)
- `Notepad++`. One of the most powerful plain text editors. Supports a lot of programming languages.

Índice

Índice

Beginning our first dynamic report with LaTeX I

1. Use your favourite template of LaTeX changing the extension of the file to `.Rnw`.
2. Insert the code chunks within the delimiters « »= and @. Knitr evals the code and show results, messages, warnings and figures.

Listing 1: first.Rnw

```
\documentclass{article}
\begin{document}
\title{A First Example}
Simple example of a simulation of $n=50$ samples from a
standard normal  and the computation of the CI for the mean.
<<simu,fig.height=4,fig.width=4,fig.align='center'>>=
n=50
x=rnorm(n)
plot(density(x),main="N(0,1)",ylim=c(0,0.4))
curve(dnorm(x),from=-4,to=4,add=TRUE,col=2,lwd=2)
cimean=mean(x)+c(-1,1)*qnorm(0.975)*sd(x)/sqrt(n)
@
The confidence interval for the mean is $(\Sexpr{round(cimean,3)})$.
\end{document}
```

Beginning our first dynamic report with LaTeX II

```
knit2pdf("first.Rnw")


>
>
> processing file:  first.Rnw

>
  |
  |                                                                 |   0%
  |
  |.....................                                            |  33%
>   ordinary text without R code
>
> List of 3
>  $ fig.height: num 4
>  $ fig.width : num 4
>  $ fig.align : chr "center"
>
  |
  |.....................................................................| 100%
>    inline R code fragments

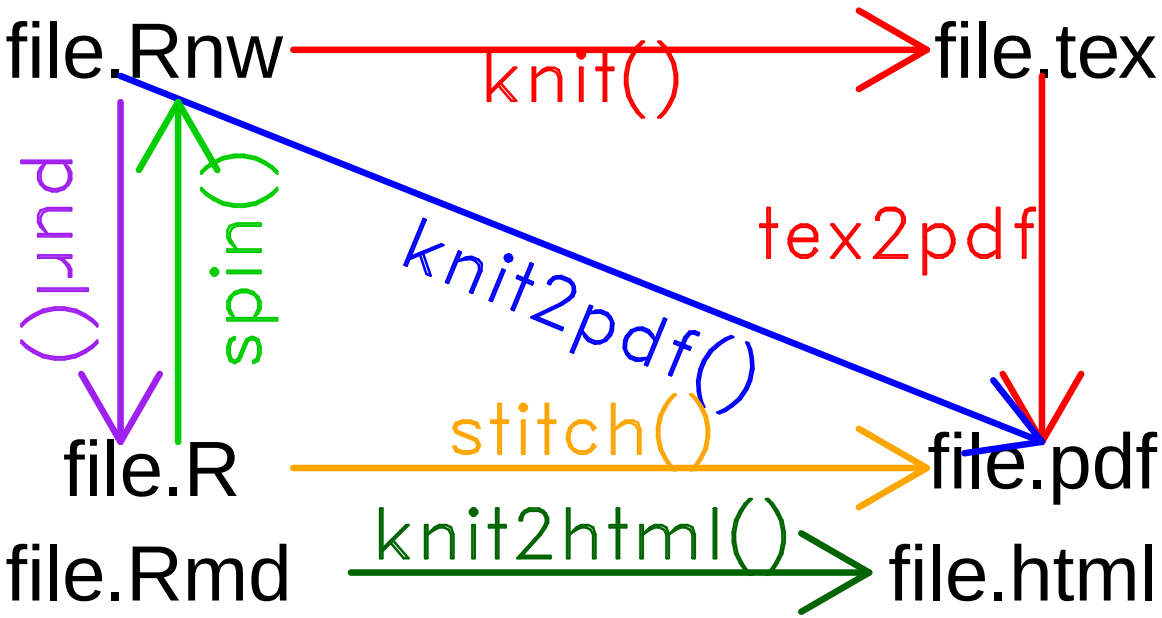
> output file:  first.tex

> [1] "first.pdf"
```

See the result: first.pdf

Functions to be used within R

- `knit("first.Rnw")`: evaluates the code and creates a TEX output: `first.tex`.
- `knit2pdf("first.Rnw")` evaluates the code and creates a PDF output: `first.pdf`
- `knit2html("first.Rmd")` evaluates the code and creates a HTML output: `first.html`
- `stitch("first.R")` creates an automatic PDF report from your code.
- `purl("first.Rnw")` extracts the code within the file and creates a R script: `first.R`
- `spin("first.R")` creates an `.Rnw` file using the lines with `#'` as regular text and those with `#+` as chunk options.



Functions in knitr

Options for chunks in knitr I

Every chunk can have a label and several options «[label=]name,opt1=val1,opt2=val2,...»=. In the options we can write regular R expressions.

- «eval=TRUE|FALSE|c(1,3:4)»: Which lines must be evaluated?
- «echo=TRUE|FALSE|c(1,3:4)»: Which lines must be printed?
- «results='markup'|'asis'|'hold'|'hide'»: How the results are showed.
- «warning=TRUE|FALSE»: Are the warnings showed?
- «message=TRUE|FALSE»: Are the messages showed?
- «error=TRUE|FALSE»: Should R stop with errors?
- «split=FALSE|TRUE»: Should R split the results?
- «tidy=TRUE|FALSE»: Decorate your code using formatR. Use tidy.opts=list(blank=FALSE, width.cutoff=60) for the details.
- «prompt=FALSE|TRUE»: The prompt symbol is included with the results.
- «comment='##'»: Character for the comments.
- «size='normalsize'»: Font size of the results.

Options for chunks in knitr II

- `«background='F7F7F7'»`: Background color.
- `«cache=TRUE»`: The result is stored in cache to avoid reevaluations.
- `«fig.path='figure/'»`: Folder where the figures are stored.
- `«fig.keep='high'|'all'|'none'|'first'|'last'»`: Which figure is stored?
- `«fig.show='asis'|'hold'|'animate'|'hide'»`: How to show the plots?
- `«dev='CairoPDF'»`: Graphics devices for figures. Several devices are allowed: `bmp`, `postscript`, `pdf`, `png`, `svg`, `jpeg`, `pictex`, `tiff`, `win.metafile`, `cairo_pdf`, `cairo_ps`, `CairoJPEG`, `CairoPNG`, `CairoPS`, `CairoPDF`, `CairoSVG`, `CairoTIFF`, `cairo_pdf`, `cairo_png`, `cairo_ps`, `cairo_svg`, `tikz`. The options for devices can be included in `dev.args`.
- `«fig.width=5, fig.height=6,fig.asp=1»`: Width, height (in inches) and aspect ratio for figures.
- `«out.width='.8\\linewidth', out.height='4in', out.extra='angle=90'»`: Final output options for the figures. (always with units)
- `«fig.align='center', fig.env='figure', fig.cap='My caption'»`: Options for figures.

See a complete list at `http://yihui.name/knitr/options`

Índice

Global options and hooks

The Global options can be included at the begining of the document.

```
<<cache=FALSE, include=FALSE>>=
opts_chunk$set(fig.path='figure/beamer-',fig.align='center',
fig.show='hold',size='footnotesize',fig.height=4,
comment=">",tidy=TRUE,tidy.opts=list(blank=FALSE))
@
```

A hook is a function to be applied to chunks or to the outputs (see later).

```
<<include=FALSE>>=
# Select lines to be showed, include outlines=begin:end
hook_output = knit_hooks$get("output")
knit_hooks$set(output = function(x, options) {
    if (!is.null(n <- options$out.lines)) {
        if (length(n) == 1)
            n <- c(n, n)
        x = unlist(stringr::str_split(x, "\n"))
        if (length(x) > n[1]) {
            # truncate the output
            x = x[intersect(seq_along(x), n)]
            if (n[1] > 1)
                x = c("....\n", x)
            if (range(n)[2] < length(x))
                x = c(x, "....\n")
        }
        x = paste(x, collapse = "\n")  # paste first n lines together
    }
    hook_output(x, options)
})
knit_theme$set("default")  #There are more than eighty themes
@
```

See examples and documentation at http://yihui.name/knitr/hooks/

Example

Source

«echo=1:3,out.lines=8:12»=

```
x = rnorm(100)
y = 5 * x + rnorm(100, sd = 0.1)
z = lm(y ~ x)
summary(z)
```

@

Result

```
x = rnorm(100)
y = 5 * x + rnorm(100, sd = 0.1)
z = lm(y ~ x)


....

> Coefficients:
>              Estimate Std. Error t value Pr(>|t|)
> (Intercept) -0.015550   0.010621  -1.464    0.146
> x            5.013906   0.009597 522.425   <2e-16 ***
> ---
....
```

We can use online expressions using \Sexpr{.}. The slope is obtained with \Sexpr{z$coef[2]}, e.g. $\beta = 5.0139059$

Example II

Source

```
«results='asis'»=
b = z$coef
cat(sprintf("$y = %.03f + %.03f x$", b[1], b[2]))
x = rnorm(1, sd = -1)
x = rnorm(-1, sd = 1)
message("Hello Niteroi")
@
```

Result

```
b = z$coef
cat(sprintf("$y = %.03f + %.03f x$", b[1], b[2]))
```

$y = -0.016 + 5.014x$

```
x = rnorm(1, sd = -1)
```

```
> Warning in rnorm(1, sd = -1):  NAs produced
```

```
x = rnorm(-1, sd = 1)
```

```
> Error in rnorm(-1, sd = 1):  invalid arguments
```

```
message("Hello Niteroi")
```

```
> Hello Niteroi
```

Índice

Simple tables

```
<<results='asis'>>=
M = matrix(rnorm(12), ncol = 4, nrow = 3)
colnames(M) = paste("rep", 1:4)
rownames(M) = paste("row", 1:3)
kable(M, col.names = paste("rep", 1:4), align = c("r", "l", "c", "r"), digits = c(1,
    2, 3, 4), caption = "Caption of my table")
```

Table: Caption of my table

	rep 1	rep 2	rep 3	rep 4
row 1	-0.9	-0.87	0.339	-1.3328
row 2	1.8	0.51	1.007	-1.8401
row 3	0.2	0.80	-1.847	-0.7022

Example Tables (with xtable)

«results='asis'»=

```
library(xtable)
xtab = xtable(M, caption = "The long caption of my table 2", label = "mytab",
    align = "|p{3cm}|rlc|p{1cm}|", digits = c(0, 1, 2, 3, 4), display = c("s",
        "f", "fg", "e", "G"))
print(xtab, hline.after = c(-1, 0, 0, 0, 1, 3, 3), size = "tiny", caption.placement = "top")
```

Table: The long caption of my table 2

	rep 1	rep 2	rep 3	rep 4
row 1	-0.9	-0.87	3.392e-01	-1.333
row 2	1.8	0.51	1.007e+00	-1.84
row 3	0.2	0.8	-1.847e+00	-0.7022

The table can be referred as `\ref{mytab}` (This is the Table 2)

See all options in function `print.xtable`

Example Tables (with knitLatex)

The package knitLatex allows the use of table(xTab), longtable(lTab) and supertabular(sTab) environments.

```
«results='asis'»=
```

```
library(knitLatex)
xTab(round(M, 4), caption.bottom = "The long caption of my table 3", label = "mytab2",
    rows = TRUE, coldef = "|p{3cm}|rlc|p{1cm}|")
```

	rep 1	rep 2	rep 3	rep 4
row 1	-0.8894	-0.8678	0.3392	-1.3328
row 2	1.8423	0.5117	1.0074	-1.8401
row 3	0.2496	0.8005	-1.8474	-0.7022

Table: The long caption of my table 3

The table can be referred as \ref{mytab2} (This is the Table 3)

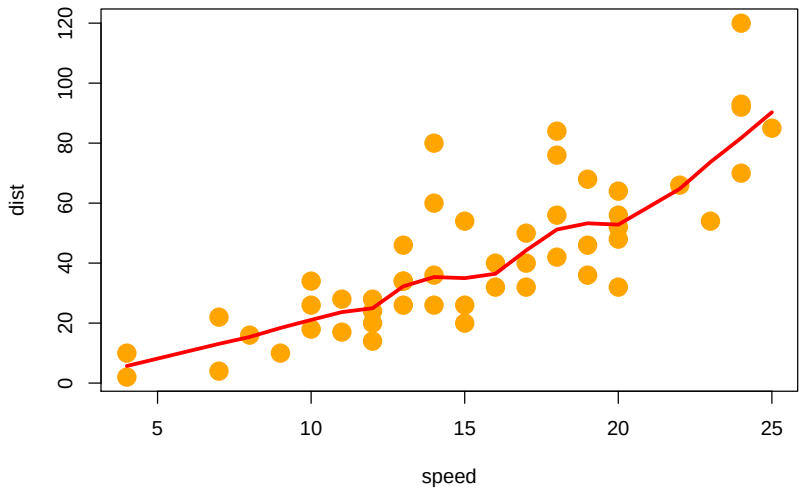
See all options in functions xTab, lTab, sTab

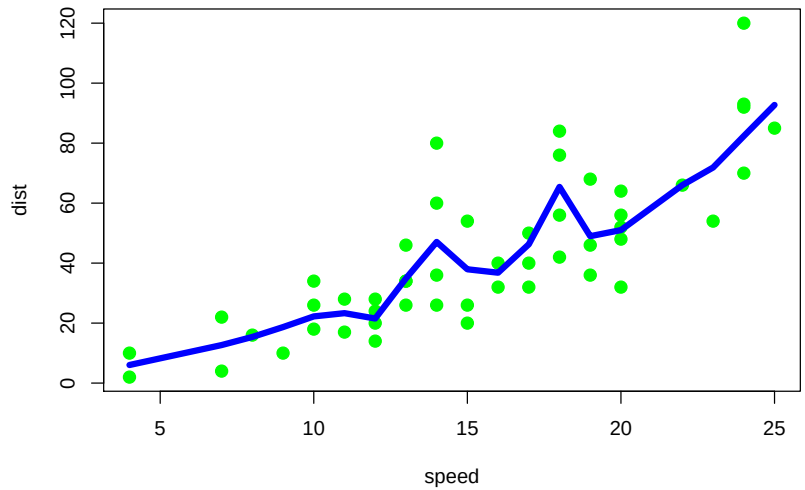
Índice

Simple Graphics

```
«cars,dev="jpeg", fig.height=5, out.height='.45\\linewidth'»=
```

```
plot(cars, pch = 19, col = "orange", cex = 2)
lines(lowess(cars, f = 0.3), col = "red", lwd = 3)
plot(cars, pch = 20, col = "green", cex = 2)
lines(lowess(cars, f = 0.2), col = "blue", lwd = 5)
```

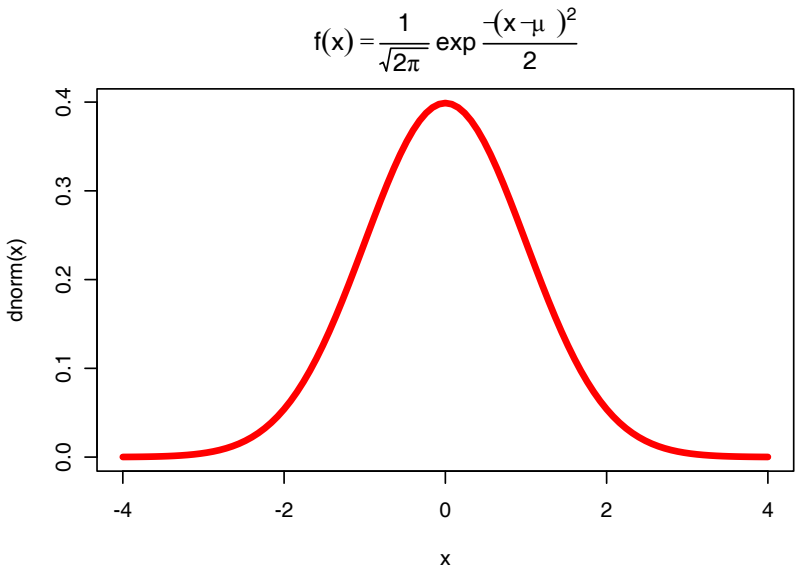


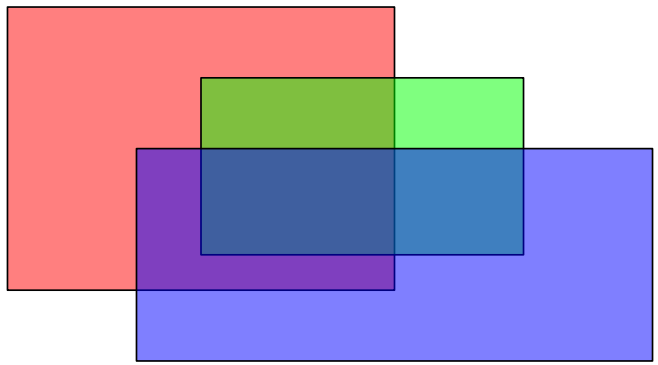


Simple Graphics

```
«cars2,dev="CairoPDF", fig.height=5, out.height='.45\\linewidth'»=
```

```
curve(dnorm(x), from = -4, to = 4, lwd = 4, col = 2, main = expression(paste(f(x) ==
    frac(1, sqrt(2 * pi)), " ", plain(exp), " ", {
    frac(-(x - mu)^2, 2)
})), cex = 2)
plot(c(0, 1), c(0, 1), type = "n", ann = F, axes = FALSE)
rect(c(0, 0.3, 0.2), c(1, 0.8, 0), c(0.6, 0.8, 1), c(0.2, 0.3, 0.6), col = rainbow(3,
    alpha = 0.5))
```



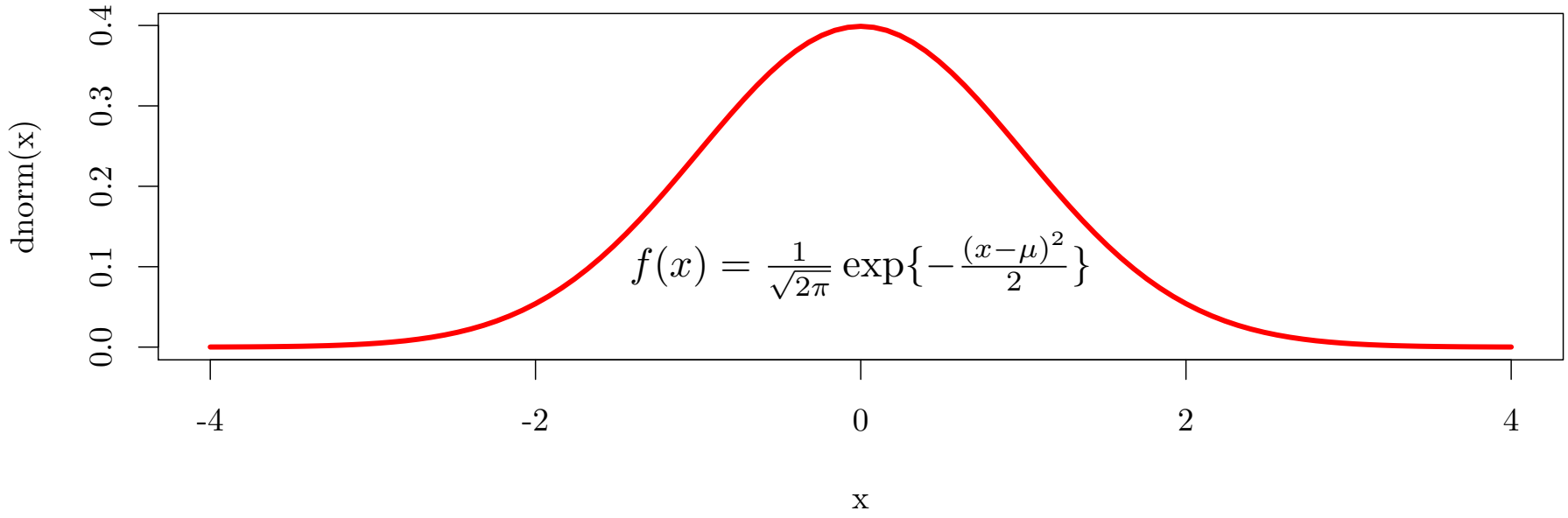


tikz Graphics

This device converts the graphics elements to LaTeX commands. Specially recommended for printing high quality math expressions inside graphs.

«normal,dev="tikz"»=

```
curve(dnorm(x), from = -4, to = 4, lwd = 4, col = 2, main = "")
text(0, 0.1, paste("$f(x)=\\frac{1}{\\sqrt{2\\pi}}", "\\exp\\{-\\frac{(x-\\mu)^2}{2}\\}$"),
    cex = 1.25)
```



Animations in PDF

Only works for Adobe Acrobat. In LaTeX, it must be included a call to \usepackage{animate}.
«clock,fig.show='animate',interval=1 »=

```
par(mar = rep(3, 4))
for (i in seq(pi/2, -4/3 * pi, length = 12)) {
    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()
}
```

Índice

Cache

When the code chunks is computationally heavy it is interesting to cache it. (only recompute when there are changes).
The cache is updated when:

- There is a change in a object inside the chunk. (in simulations use `set.seed`).
- There is a change in a `R` object defined in «»=.
- There is a change in a chunk that this one depends on. (include dependson)

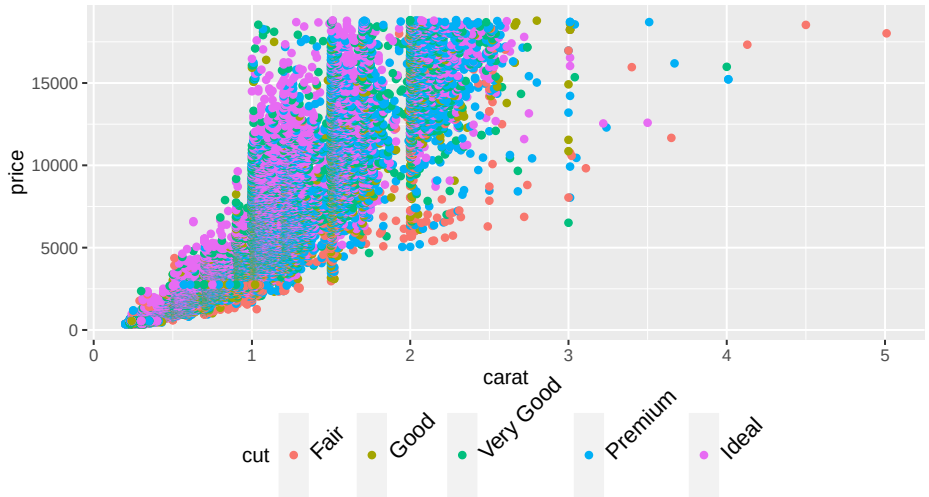
```
«map,cache=TRUE,vers=R.version.string,file.time=file.info('map_brazil.R')$mtime»=
source("map_brazil.R")
```

Cross reference I

The code written in one chunk can be inserted into another one. The syntax is simply «label»

```
«legend,eval=FALSE»=
theme(legend.text = element_text(size = 12, angle = 45)) + theme(legend.position = "bottom")
```

```
«plot1»=
qplot(carat,price,data=diamonds,color=cut)+ «legend»
@
```



Parents and children I

If the document is too long (for instance a book) it would be a good idea to split it in chapters.

Listing 2: parent.Rnw

```
\documentclass{article}
\begin{document}
\title{Parent example}
<<A>>=
x=rnorm(12)
y="Hello"
@
<<B,child='child.Rnw'>>=
@
The value of $y$ is \Sexpr{y}. (Not Hello!!).
<<C>>=
x <- rnorm(8)
@
Include the child document again.
<<D, child='child.Rnw'>>=
@
The value of $y$ is \Sexpr{y}. (Not Hello!! nor the previous value.).
\end{document}
```

Parents and children II

Listing 3: child.Rnw

```
<<include=FALSE>>=
set_parent('parent.Rnw')
@

This is the child document

<<>>=
y=sum(x^2)
@
```

See the result: parent.pdf

Hooks: Extensions to knitr I

A hook is a function (stored in `knit_hooks`) that modifies the behavior of a chunk. A hook always has three arguments: `before`, `options` and `envir`.

```
names(knit_hooks$get(default = TRUE))  #default hooks
```

```
knit_hooks$set(inline = function(x) {
    if (is.numeric(x))
        round(x, 3)
})  #Change the default hook
knit_hooks$set(margin = function(before, options, envir) {
    if (before) {
        m <- options$margin
        if (is.numeric(m) && length(m) == 4L) {
            par(mar = m)
        }
    }
})
knit_hooks$set(title = function(before, options, envir) {
    if (before) {
        paste(options$title, "\\hrulefill", sep = "")
    } else {
        paste("\\hrulefill", sep = "")
    }
})
```

«title="Code example"»=

Hooks: Extensions to knitr II

Code example

```
1 + 1  #Simple example of a hook with title
```

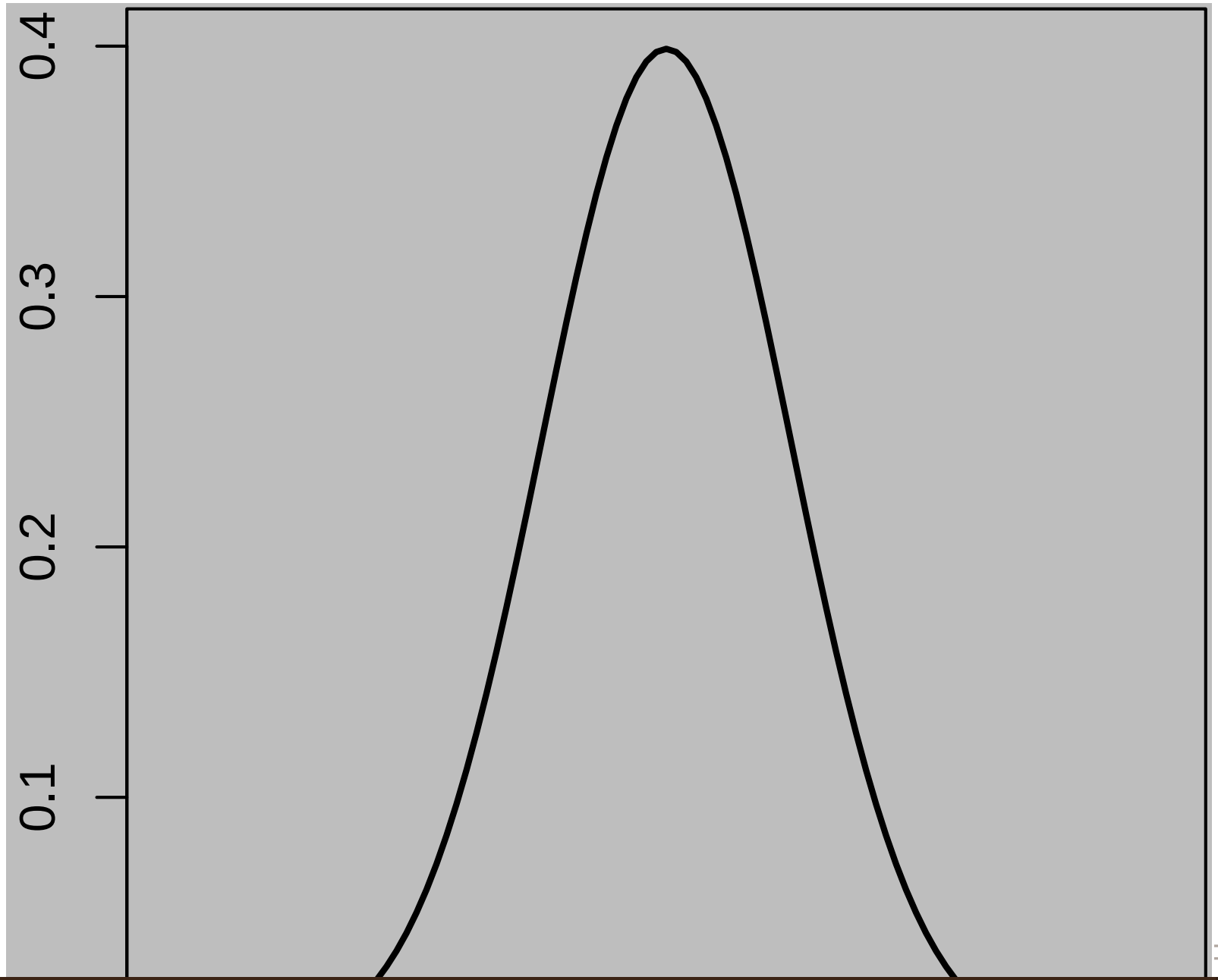
The options of a chunk can also be modified by other options with the function `opts_hooks`. The function `opts_template` allows to group a list of options.

```
opts_hooks$set(fig.height = function(options) {
    if (options$fig.height < options$fig.width) {
        options$fig.height = options$fig.width
    }
    options  #A plot with at least the same height as width
})
opts_template$set(fig.large = list(fig.width = 6, fig.height = 5, out.width = ".8\\linewidth"),
    fig.small = list(fig.width = 4, fig.heigth = 3, out.width = "4in"))
```

```
«margin=c(2,2,.1,.1),opts.label='fig.small'»=
```

```
curve(dnorm(x), from = -4, to = 4, lwd = 2)
```

Hooks: Extensions to knitr III



A couple of tricks I

- The chunk options can be shortened with `set_alias`, e.g., `set_alias(w="fig.width",h="fig.height")`
- With `beamer`, always use `[fragile]` in each frame with code chunks.
- Every LaTeX commands included in the chunk options or as characters must be written with the double backslash, e.g. `'\\linewidth'`, `fig.cap='$\\beta=0.5$'`
- If you want reproducible simulations, remember the use of `set.seed()`.

Índice

R Markdown v2 I

`R Markdown` is a simple plain text format that can be converted to valid `XHTML` or `HTML`. This allows to convert a simple document to many other formats: PDF, Word, HTML5,...
The structure is: `Metadata` (at the begining between `---`) + `Contents` (plain text).
The code chunks begin with ```` ```{r label, opts} ```` and ends with three backsticks ```` ``` ````

Listing 4: example.Rmd

```
---
title: "A simple Markdown example for use with RStudio"
author: "Manuel Febrero"
date: "25/04/2016"
output:
  word_document: default
  pdf_document:
    template: null
  html_document:
    fig_caption: yes
    number_sections: yes
    theme: readable
bibliography: Rmd.bib
---
```

R Markdown v2 II

```
# Header 1
## Header 2
### Third level
This is an usual paragraph with **bold** and _italic_

- list item 1
        1. item 1.1
        2. item 1.2
- list item 2


The code is include in `backticks`. The links are written using `[name](url)` like,
for example, [Markdown](http://en.wikipedia.org/wiki/Markdown).
For including images, use `![image](url)`

The R code is include in three backticks.'``{r␣name,␣options}␣...␣``'
```{r plot, fig.cap='Example␣of␣a␣plot'}
x=rnorm(10)
y=rnorm(10)
plot(x,y)
```

Inline math is like LaTeX: $\alpha + \beta = 1$. Inline R is done with a single backstick
`'r␣mean(x)'`: Mean=`r mean(x)`. For center equations use double $$.
$$f(x)=\sin(x^2)+1$$.

Also the references are quite simple and the syntax is `[@ref]`
See [@Baumer2014], [@Allaire2015] and [@Udwin2015].
```

R Markdown v2 III

````
Table:
A simple table

| id   | age | sex |
|:----|----:|:---:|
| a    | 49  | M   |
| b    | 32  | F   |


## A couple of Pandoc extensions.
(@) Think what is `0.3+0.4-0.7`=`r 0.3+0.4-0.7`.
(@weird) Now think what is `0.3-0.7+0.4`=`r 0.3-0.7+0.4`.

A footnote here[^1]. The result in (@weird) is quite surprinsingly.


```{r, echo=FALSE}
options(digits=2)
knitr::kable(cbind(x,y))
```


[^1]: This is the footnote
````

Simply edit and compile this file in RStudio and see the result: example.html
With Pandoc (http://www.pandoc.org) this file can be converted to Word Documents (.docx), PDF, Shiny, HTML5 and much more.

See the help included in RStudio.

R Markdown and Shiny I

To produce simple Shiny documents using `R Markdown` change Metadata to include `output: html_document` and `runtime: shiny`.

Listing 5: example2.Rmd

```
---
title: "A simple Markdown--Shiny example"
author: "Manuel Febrero"
date: "25/04/2016"
runtime: shiny
output:  html_document
---

# Effect of a bandwidth in Nonparametric Curve Estimation

Select the value of the bandwidth

```{r}
library(shiny)
x=faithful[,2] #Old Faithful Geyser Data
r=c(40,100)
sliderInput("h","Bandwidth:",min=diff(r)/30,max=diff(r)/5,value=diff(r)/12)
selectInput("kernel","Kernel:",choices=list("gaussian", "rectangular", "triangular",
                               "epanechnikov", "biweight", "cosine", "optcosine"))

renderPlot({
h=seq(diff(r)/30, diff(r)/5,len=101)
hist(x,freq=FALSE,col='darkgray',border='white')
lines(density(x,bw=input$h,kernel=input$kernel),col=2,lwd=3)
```
```

R Markdown and Shiny II

````
})
```


# Historic currency

```{r, echo=FALSE,message=FALSE}
library(quantmod)
today=Sys.Date()
dateRangeInput("dates", label = h3("Date range"),start=today-200,end=today-1,
               min=today-1000,max=today,format="yyyy-mm-dd")

selectInput("cur1","Currency 1:",choices=list("Euro"="EUR", "US Dolar"="USD",
                  "British Pound"="GBP", "Brasil Real"="BRL"))
selectInput("cur2","Currency 2:",choices=list("Euro"="EUR", "US Dolar"="USD",
                  "British Pound"="GBP", "Brasil Real"="BRL"),selected="USD")

renderPlot({
  from=as.Date(input$dates[1],"%Y-%m-%d")
  to=as.Date(input$dates[2],"%Y-%m-%d")
  getFX(paste(input$cur1,"/",input$cur2,sep=""),from=from,to=to)
  name=paste(input$cur1,input$cur2,sep="")
  name2=paste(input$cur1,input$cur2,sep="/")
  aa=get(name)
plot(aa,main=name2)
})
```
````

Simply edit and compile this file in `RStudio`.

Alternatively, run `rmarkdown::render()`. (pandoc must be installed).
````

Índice

References

[Pandoc] John McFarlane (2016). Pandoc: A Universal Document Converter. http://pandoc.org.

[rmarkdown] JJ Allaire et al (2016). R Markdown. Dynamic Documents for R. http://rmarkdown.rstudio.com/.

[knitr] Yihui Xie (2013). knitr: A General-Purpose Tool for Dynamic Report Generation in R. http://yihui.name/knitr.

[Rknitr] Yihui Xie (2015). Dynamic Documents with R and knitr, second edition. CRC Press.