

# Reporting using Dynamic Documents

L. Torgo

ltorgo@fc.up.pt

Faculdade de Ciências / LIAAD-INESC TEC, LA  
Universidade do Porto

Sept, 2014



Introduction

## Reporting

### The Standard Process of Data Analysis

- Import the data into our favorite data analysis tool
- Carry out a set of data analysis steps
- Report the work by building some document (report and / or presentation)
  - Series of *copy+paste* steps from parts of the results of the analysis into some word processing and/or presentation software tool, adding some supporting text
- Frequently all process needs to be repeated / iterated!

## Some of the Dangers of this Standard Approach

- Too many manual steps  $\mapsto$  great potential for human error
- Too much human effort (time) in the process with many repetitive and boring tasks, like for instance the communication between different software tools
- All process is hardly recordable for future re-use, due to the extensive use of graphical user interfaces
- Small changes on the initial data require full repetition of all process!
- The Analysis and the Reporting are separated and thus great care is required to have both in “sync” avoiding reporting errors
- Very hard to share the work with other teams
- Very hard to re-use the work on similar tasks

List inspired by *Dynamic Documents with R and knitr* by Yihui Xie

## Dynamic Documents

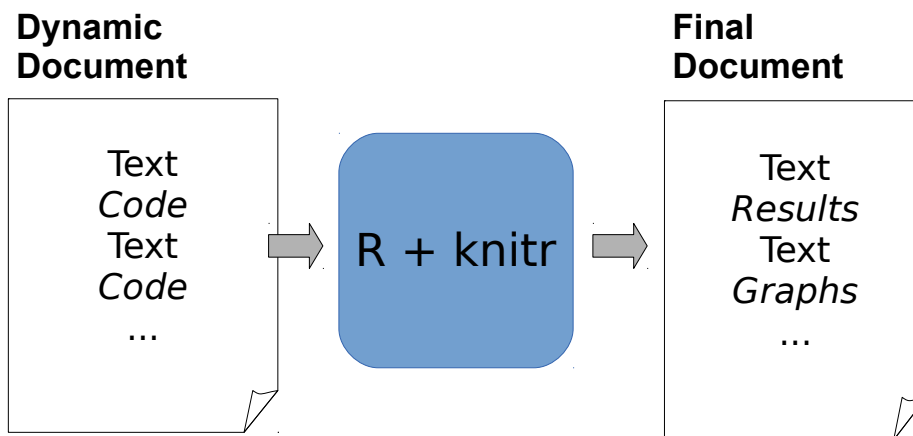
### What are Dynamic Documents

- Documents that mix data analysis steps with descriptive text
- Documents that are executable by a computer program to produce the final document
- This final document is produced by a computer program from the initial document created by the user containing data analysis steps and descriptive text

**Dynamic Documents solve most of the problems we have described before!**

# Dynamic Documents

R+knitr - an implementation of the idea



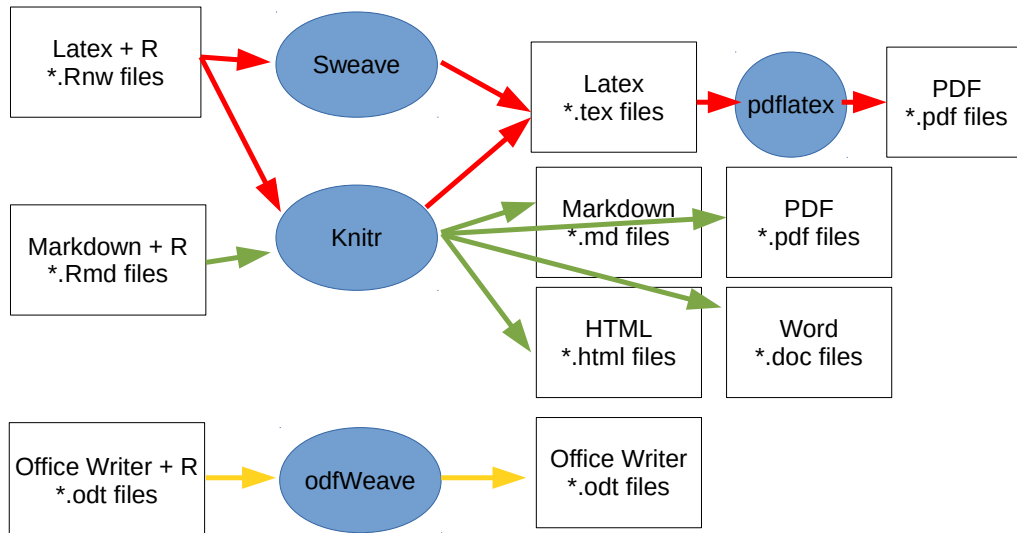
# Dynamic Documents

- The idea is related with the concept of *literate programming*
- It is implemented in 3 main steps:
  - 1 Inspect the dynamic report and separate the data analysis code from the descriptive text
  - 2 Execute the code and store the results of each code chunk
  - 3 Produce the final document replacing the data analysis code in the original document by the results
- NOTE: all process is carried out without human intervention!

Knuth, Donald E. (1984). "Literate Programming". The Computer Journal (British Computer Society) 27 (2): 97–111.

# Dynamic Documents

Different implementations of the concept



## R+knitr

- Knitr is a platform that allows:
  - Using different document formats
    - $\LaTeX$ , HTML and Markdown
  - Using different programming languages to implement the data analysis steps
    - R, Python, awk, C++ and shell scripts

# rmarkdown

- **rmarkdown** is an R package that allows integrating R with Markdown using **knitr** and **pandoc**
- **Pandoc** is a kind of swiss-army knife of document format conversion
- **R markdown** is a literate programming approach that allows embedding R code into markdown documents
- **Markdown** is a very simple markup language that was designed to easily produce internet content

## A (Very) Brief Introduction to Markdown

- Very simple language for creating Internet content
  - Much simpler than the native language - HTML
  - A tool with the same name can be used to convert markdown into HTML
- The format is a simple text file and you do not need any special software tool to create Markdown files

## Character formatting in Markdown

- Formatting is carried out with the help of small annotation tags

- Examples:

`**Note**` or `__Note__`

translated into **Note** (i.e. boldface)

## Sections and subsections

- An hash (#) character before a text line indicates a first level section heading
- If instead you use two or three hashes we get second and third level sections

Illustrations on the use of Markdow in Dynamic Reports

Introduction

R comes with a series of data sets. We may use the function `data` to load them, as show in the following example:

```
# Illustrations on the use of Markdow in Dynamic Reports
```

```
## Introduction
```

```
R comes with a series of data sets. We may use the function
**data** to load them, as show in the following example:
```

## Lists of Items

```
# OM prototype - Admin User
```

```
+ Stage 1
```

- Define topics of interest
- Define web sources
  - \* create crawlers (\*\*Potentially Challenging)
  - \* collect data

```
+ Stage 2
```

- Obtain tagged data (\*\*Potentially Challenging)
- Obtain, evaluate and select models

OM prototype - Admin User

- Stage 1
  - Define topics of interest
  - Define web sources
    - create crawlers (Potentially Challenging)
    - collect data
- Stage 2
  - Obtain tagged data (Potentially Challenging)
  - Obtain, evaluate and select models



## Embed images and links

```
# Monitoring and Forecasting Water Quality Parameters
```

- Collaboration with [Águas do Douro e Paiva, SA] (<http://addp.pt/pt/home.php>)
- FCT project [MORWAQ] (<http://liaad.inescporto.pt/modys/projects/morwaq>)
- Some of the main results:
  - Software prototype for monitoring and forecasting water quality parameters
  - Several publications
    - KDD'2011, ECAI'2010

```
![ADdP Network] (addpNet.png "The ADdP Network")
```

Monitoring and Forecasting Water Quality Parameters

- Collaboration with [Águas do Douro e Paiva, SA](#)
- FCT project [MORWAQ](#)
- Some of the main results:
  - Software prototype for monitoring and forecasting water quality parameters
  - Several publications
    - KDD'2011, ECAI'2010



# R Markdown v2

## Embedding R code into documents

```
# The Distribution of Sepal Length
```

An histogram of the variable can be obtained with:

```
```{r eval=FALSE}
hist(iris$Sepal.Length)
```
```

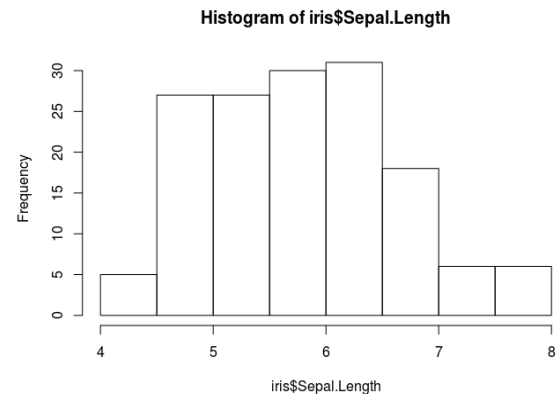
```
```{r echo=FALSE}
data(iris)
hist(iris$Sepal.Length)
```
```

Chunk+Insert Chunk in RStudio

## The Distribution of Sepal Length

An histogram of the variable can be obtained with:

```
hist(iris$Sepal.Length)
```



# Code Chunks in R Markdown

- Code chunks are parts of a document that should be executed by R
- In R Markdown they are indicated as follows:
 

```
```{r}
...
```
```
- Everything between these two lines will be executed by R
- The result of the execution will be part of the final document
- Inline code can be used like this: ``r 2+2``



## Code Chunk Options

- The way code chunks are interpreted by Knitr is controllable through a series of chunk options
- Without any option code chunks are:
  - 1 Executed
  - 2 A code block is added to the final document
  - 3 A results block is added to the final block
- Chunk options allow to adjust these defaults
- This can be done on each individual chunk or globally

## Some common chunk options

**eval** (TRUE) - allows to control whether the chunk code is to be executed or not by R. In the following example the code is inserted into the final document but it will not be executed by R

```
```{r eval=FALSE}
hist(iris$Sepal.Length)
```
%$
```

**echo** (TRUE) - allows to hide the R code from the final document (with the value FALSE), with only the results of the execution being included in that document (e.g. a figure)

- Note that each chunk may include several options separated by commas
- Many more options exist! See an exhaustive list with explanations at <http://yihui.name/knitr/options>

## Generating the Final Document

- Function `render()` from package **rmarkdown** receives as input a markdown file and produces the final document using R to execute the code chunks:

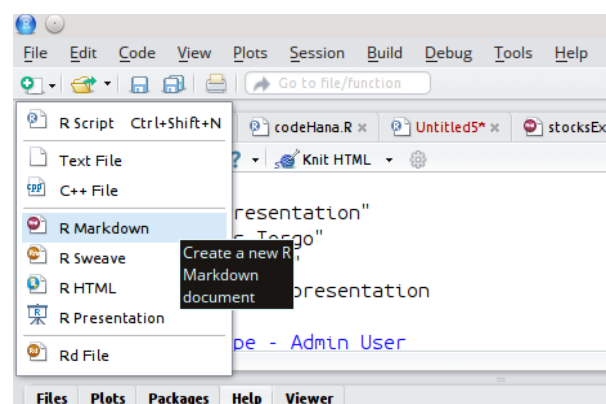
```
library(rmarkdown)
render("initialDoc.Rmd")
```

- The type of output document that is generated can be controlled through meta-data information that is included in the initial document,

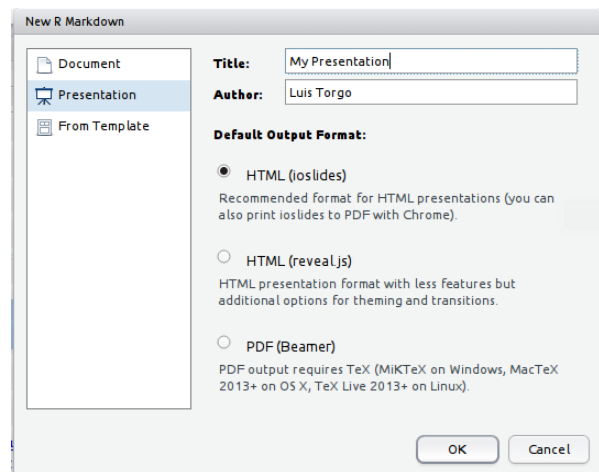
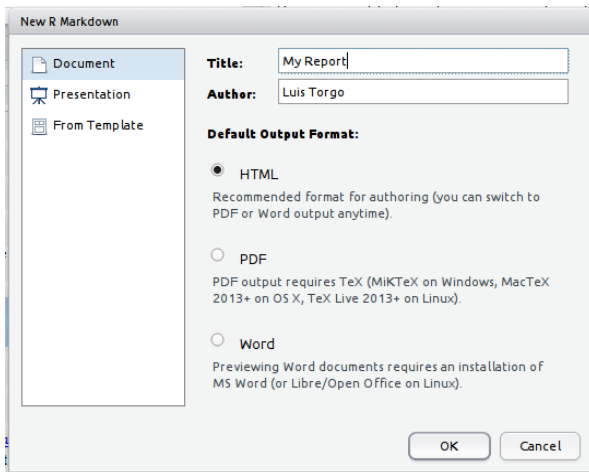
```
---
title: "My First Dynamic Report"
output:
  pdf_document:
    toc: true
    highlight: zenburn
---
```

## Using RStudio interface

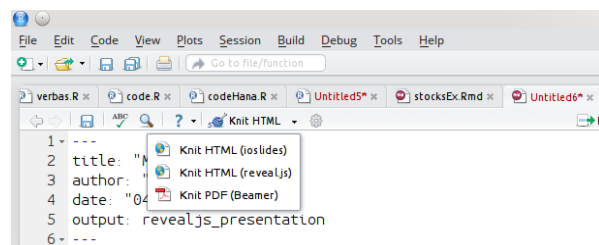
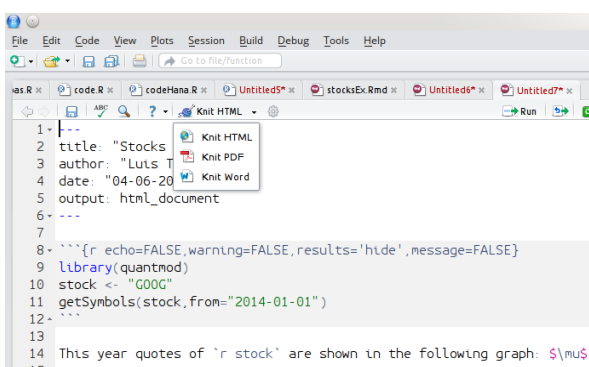
Recent versions of RStudio facilitate the creation of dynamic reports and presentations using R Markdown v2



# Selecting the Type of Document Using RStudio interface



# Generating the Output Document Using RStudio interface



## Learning more

Much more information is available at  
<http://rmarkdown.rstudio.com/>

## Hands On Dynamic Reports

Using the **algae** data set create a small data analysis report containing some exploratory analysis of this data. Include the following aspects in your report:

- Data summarization
- Data visualization

Try to produce a credible document. Take advantage of what was shown before in terms of data summarization and visualization, joining this with the current topic of dynamic reports using R+Markdown.