

# Markup languages and the terminal—Power at your fingertips

Thomas de Graaff & Daniel Arribas-Bel

September 5, 2014

# Introduction

# Recap

Our main goal:

*To make our research as reproducible and visible as possible*

This entails:

1. Sharing of code
2. Sharing of data (if possible and not proprietary nor privacy sensitive)
3. Sharing of output (presentation, article, website)

# The power of plain text

1. Ubiquitous
2. Usually small in size
3. Portable across platforms (and versions)
  - ▶ it will not be obsolete soon
  - ▶ everyone can read it everywhere
4. It is scriptable (both as input as output)
  - ▶ code is almost **always** in text format
  - ▶ **usually** data is in text format as well
  - ▶ but underlying format for output (presentation, website, tables, articles, books) **can** be text as well

# Manipulation of text

- ▶ Most output is based on simple text file; applications only change appearance, such as:
  - ▶ browsers
  - ▶ pdf
- ▶ **How** to change appearance require markup-languages
  - ▶ HTML
  - ▶ LaTeX
  - ▶ Markdown

# Latex and friends

# LaTeX

## 1. What?

- ▶ A set of macros around Tex, a markup language invented by Donald Knuth

## 2. How?

- ▶ Latex is a document preparation system and document markup language. Source: Wikipedia

## 3. Why?

- ▶ Defacto standard in academic publishing
- ▶ Formulae used in HTML pages (e.g., Wikipedia)
- ▶ Macro's thus scriptable (whoohoo)

## 4. But...

- ▶ Notation a bit cumbersome
- ▶ For small texts a bit too much and not geared for HTML (see also)

## A minimal example

```
\documentclass[12pt]{article}
\begin{document}
\section{My Paper}
I just discovered that:
\begin{equation}
e=mc^2
\end{equation}
\end{document}
```



# Bibtex

- ▶ Basically a free reference manager (actually more a style of managing references)
- ▶ Very versatile and very powerful (most other markup languages work with bibtex as well)
- ▶ Free managers, such as bibdesk or mendeley, are now ubiquitous

# Markdown

# Why markdown?

1. Easy to learn

`http://daringfireball.net/projects/markdown/`

2. Much less notation than Latex . Originally,

- ▶ LaTeX is for paper (aka dead trees)
- ▶ Markdown is for HTML (blogs, wikipedia and so)
- ▶ but sneakily uses some Latex when needed

3. Focus on text

4. Nowadays:

- ▶ “easily” change it in html or pdf (via Latex)—even in Word if needed
- ▶ can be extended with code or—much better—its results

# Language syntax

Emphasis:

```
*italic* **bold**  
_italic_ __bold__
```

Headers:

```
# Header 1  
## Header 2  
### Header 3
```

# Language syntax (cont.)

## Unordered lists

- \* Item 1
- \* Item 2
  - + Item 2a
  - + Item 2b

## Ordered List

1. Item 1
2. Item 2
3. Item 3
  - + Item 3a
  - + Item 3b

# Language syntax (cont.)

Links:

```
http://assemble.io/docs/Cheatsheet-Markdown.html
```

```
[Cheatsheet] (http://assemble.io/docs/Cheatsheet-Markdown.ht
```

Images:

```
![alt text] (http://example.com/logo.png)
```

```
![alt text] (figures/img.png)
```

## Language syntax (cont.)

Code blocks:

```
```python
    s = "Python syntax highlighting"
    print s
```
```

which renders as:

```
s = "Python syntax highlighting"
print s
```

## Language syntax (cont.)

To embed mathematics just use Latex notation:

```
$$e=mc^2$$
```

which surprisingly looks as excel type of formulae and renders as:

$$e = mc^2$$



## Language syntax (cont.)

Inline equations just require  $\$$   $\$$ , e.g.:

In economics it is well known that:

```

$$\frac{dx}{dy} = -\frac{\partial u(x,y)/\partial y}{\partial u(x,y)/\partial x}.$$

```

which renders as

In economics it is well known that:  $\frac{dx}{dy} = -\frac{\partial u(x,y)/\partial y}{\partial u(x,y)/\partial x}.$

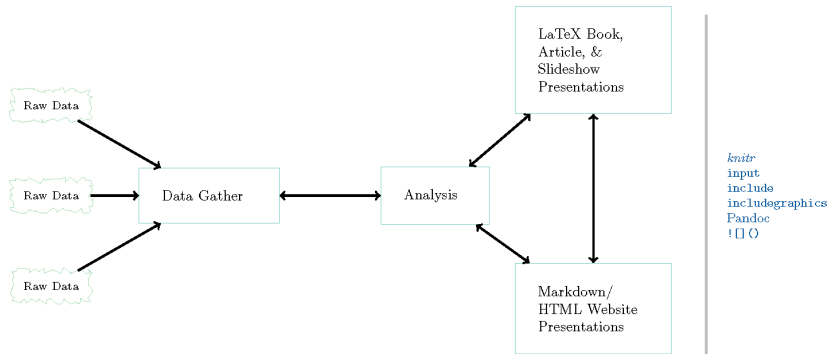
# Pandoc

# The swiss knife of formats

So how do we glue everything together and produce wonderful htmls and pdfs out of thin air? With **pandoc**

- ▶ Pandoc can convert from (not extensive):
  - ▶ markdown (whoohoo), Latex, HTML, DocBook, Org-mode, and ... Words docx
- ▶ To (and here we go...)
  - ▶ HTML formats (including those very cool and nerdy HTML(5) slides)
  - ▶ via Latex to pdf
  - ▶ Word (but support somewhat limited) and OpenOffice formats
  - ▶ various markup formats
  - ▶ and much more

# So, a typical workflow



---

```
Make  
download.file  
source.data  
source.DropboxData  
read.table  
merge  
getURL  
API-based  
packages
```

```
source.data  
source.DropboxData  
read.table  
getURL
```

```
knitr  
source  
source_url  
print(xtable())
```

# The terminal

# The terminal

- ▶ What?
  - ▶ Unix kind of application that is completely working without GUI's and governing all processes. . . (btw, powershell for Windows works now really well as well)
- ▶ Do we really need it?
  - ▶ not really, but it makes you faster, more versatile, nerdier, and you have to know what you are doing.
  - ▶ and there are some applications that only work in the terminal (such as make and pandoc)
- ▶ So, why is it so cool again?
  - ▶ you can make everything **scriptable**