

Reporting Using R and Markdown

Part 1: Introduction

Niels Hagenbuch and Claude Renaux

February 2022

Introduction

- 1 Goal
- 2 An Early Precursor
- 3 WYSIWYG
- 4 Programmed Typesetting
 - PostScript
 - T_EX and L^AT_EX
 - Markdown

Goal

Goal

Integration of

- data manipulation
- generation of tables and plots
- statistical analyses
- report writing

in **one single file**.

Benefits

- Simplified workflow
- Reproducible analyses
- Automatically updated if data change
- No copy/paste
- No formatting by hand
- Flexible output format

Downside

- Everything needs to be coded.
- Managing three complex systems:
 - ▶ Markdown
 - ▶ knitr
 - ▶ R
- Learning curve is a bit steep in the beginning.

An Early Precursor

An Early Precursor: SwyftCard (1984)

Mixing programming and text editing can be very powerful. For example, to make a calendar for the month of January, you might write the following program

```
10 FOR I = 1 TO 31
20 PRINT "JANUARY "; I
30 NEXT I
RUN
```

If you then highlight it (including the RUN) and use the CALC command you will get a calendar that begins

```
JANUARY 1
JANUARY 2
JANUARY 3
JANUARY 4
JANUARY 5
JANUARY 6
```

You can even LIST the program by typing the word "LIST" and using the CALC command on it. Almost all BASIC commands work as usual. However, the size of the BASIC program is limited to about 900 bytes (in its compacted internal form). So SwyftCard is not suitable for developing large programs.

- SwyftCard™ for Apple IIe
- Developed in 1984 by Jef Raskin
- Integration of text and BASIC
- After running the code, the result was incorporated into the text
- Code could optionally remain in the text or be removed

Source:

SwyftCard™ User Manual
Information Appliance Inc., 1985

An Early Precursor: SwyftCard (1984)

To do a calculation, such as adding 34 and 78, think of asking SwyftCard,

How much is 34 plus 78?

Abbreviate "How much is?" to a question mark and type the formula (followed by a return or space so that the answer, when it appears, will be easy to read). Highlight the formula (including the question mark) and use the CALC command (USE FRONT G). The answer will appear in your text just after the problem:

```
? 34 + 78; 112
```

The problem will remain highlighted so that you can delete it, leaving only the answer in your text. A return character is put in after the answer. If you do not want a return after the answer, type a semicolon after the arithmetic expression. For example, ? 34 + 78; will not be followed by a return.

```
===== 0 =====
```

You can do a calculation anywhere you like. All you have to do is type it, highlight it, and use the CALC command - right in the ? 34 + 78; 112 middle of a sentence, if you like.

```
===== 1 =====
```

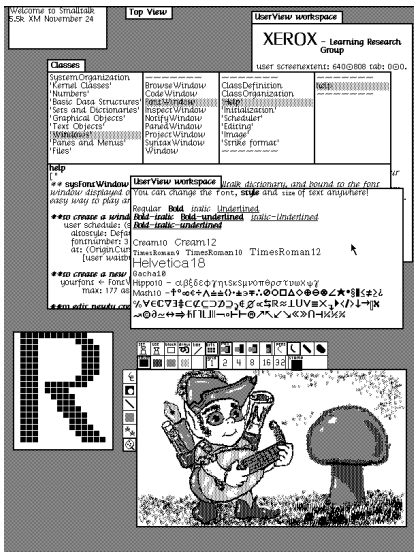
- In-line evaluation of expressions
- Operators: + * - / ^
- Code could optionally remain in the text or be removed

Source:

SwyftCard™ User Manual
Information Appliance Inc., 1985

WYSIWYG

WYSIWYG – What You See Is What You Get



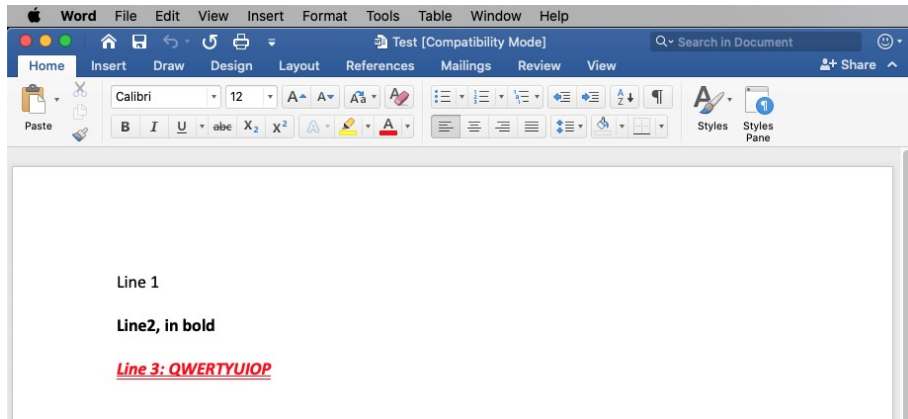
- Developed at Xerox PARC in the early 1970s
- First introduced in the Alto computer system (1973)
- Embedded in a bitmapped graphical user interface (GUI)
 - ▶ menus
 - ▶ overlapping windows
 - ▶ mouse-driven
- Steve Jobs visited Xerox PARC in 1979
 - ▶ Apple Lisa (1983)
 - ▶ Macintosh (1984)

Dynabook on a Xerox Alto, mid-1970s

https://ja.wikipedia.org/wiki/Bit_Block_Transfer (acc. 2020.01.16)

WYSIWYG – What You See Is What You Get

Word for Mac:



WYSIWYG – What You See Is What You Get

Since 2002, Word uses XML (.docx):

 < >  document.xml > No Selection

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <w:document xmlns:wpc="http://schemas.microsoft.com/office/word/2010/wordprocessingCanvas"
  xmlns:cx="http://schemas.microsoft.com/office/drawing/2014/chartex"
  xmlns:cx1="http://schemas.microsoft.com/office/drawing/2015/9/8/chartex"
  xmlns:cx2="http://schemas.microsoft.com/office/drawing/2015/10/21/chartex"
  xmlns:cx3="http://schemas.microsoft.com/office/drawing/2016/5/9/chartex"
  xmlns:cx4="http://schemas.microsoft.com/office/drawing/2016/5/10/chartex"
  xmlns:cx5="http://schemas.microsoft.com/office/drawing/2016/5/11/chartex"
  xmlns:cx6="http://schemas.microsoft.com/office/drawing/2016/5/12/chartex"
  xmlns:cx7="http://schemas.microsoft.com/office/drawing/2016/5/13/chartex"
  xmlns:cx8="http://schemas.microsoft.com/office/drawing/2016/5/14/chartex"
```

→ See example !

Programmed Typesetting

Programmed Textsetting – PostScript

- Page description language
- Developed by Adobe in 1980s
- Revolutionized printing industry
- PDF (1993) is a container file format that includes PS

Example:

```
%!PS
/Courier           % name the desired font
20 selectfont      % choose the size in points and establish
                   % the font as the current one
72 500 moveto       % position the current point at
                   % coordinates 72, 500 (the origin is at the
                   % lower-left corner of the page)
(Hello world!) show % stroke the text in parentheses
showpage           % print all on the page
```

Source: <https://en.wikipedia.org/wiki/PostScript> (2020.01.16)

Programmed Textsetting – T_EX and L^AT_EX

- Donald Knuth developed T_EX in 1978 to typeset (scientific) books with mathematical formulae
- L^AT_EX is an implementation of T_EX by Leslie Lamport, created in the early 1980s
- L^AT_EX provides extensive macros that facilitate the use of T_EX:
 - ▶ document class: article, presentation, book, letter
 - ▶ tables, figures, diagrams, along with captions
 - ▶ enumerations, itemized lists, references within the text, footnotes, links
 - ▶ automatic generation of tables of content and bibliographies
 - ▶ layout and typesetting elements like margins, spacing, use of fonts, font sizes, bold, italic, colors

Programmed Textsetting – L^AT_EX

```
\documentclass{article}
\usepackage{amsmath}
\title{\LaTeX}

\begin{document}
  \maketitle
  \LaTeX{} is a document preparation system for
  the \TeX{} typesetting program. It offers
  programmable desktop publishing features and
  extensive facilities for automating most
  aspects of typesetting and desktop publishing,
  including numbering and cross-referencing,
  tables and figures, page layout,
  bibliographies, and much more. \LaTeX{} was
  originally written in 1984 by Leslie Lamport
  and has become the dominant method for using
  \TeX; few people write in plain \TeX{} anymore.
  The current version is \LaTeXe.

  % This is a comment, not shown in final output.
  % The following shows typesetting power of LaTeX:
  \begin{align}
    E_0 &= mc^2 \\
    E &= \frac{mc^2}{\sqrt{1-\frac{v^2}{c^2}}}
  \end{align}
\end{document}
```

L^AT_EX

L^AT_EX is a document preparation system for the T_EX typesetting program. It offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing, including numbering and cross-referencing, tables and figures, page layout, bibliographies, and much more. L^AT_EX was originally written in 1984 by Leslie Lamport and has become the dominant method for using T_EX; few people write in plain T_EX anymore. The current version is L^AT_EX 2_ε.


$$E_0 = mc^2 \quad (1)$$

$$E = \frac{mc^2}{\sqrt{1 - \frac{v^2}{c^2}}} \quad (2)$$

Programmed Textsetting – Markdown

- A “lightweight markup language,” developed in 2004
- A bit like \LaTeX , but far more easy!
- The syntax is readable and unobtrusive
- Source code can be opened by any editor on any operating system
- Markdown documents are converted to HTML and can be displayed with any web browser
- Online editor: <https://dillinger.io/>
- Good introduction: <https://www.markdownguide.org/book>

Programmed Textsetting – Markdown

Text using Markdown syntax	Corresponding HTML produced by a Markdown processor	Text viewed in a browser
<p>Heading</p> <p>=====</p> <p>Sub-heading</p> <p>-----</p> <p>Paragraphs are separated by a blank line.</p> <p>Two spaces at the end of a line produces a line break.</p> <p>Text attributes <i>_italic_</i>, **bold**, <code>`monospace`</code>.</p> <p>Horizontal rule:</p> <p>---</p> <p>Strikethrough:</p> <p>--strikethrough--</p> <p>Bullet list:</p> <ul style="list-style-type: none">• apples• oranges• pears <p>Numbered list:</p> <ol style="list-style-type: none">1. lather2. rinse3. repeat <p>An <code>[example](http://example.com)</code>.</p> <p>!<code>[Image](Icon-pictures.png "icon")</code></p> <p>> Markdown uses email-style > characters for blockquoting.</p> <p>Inline <code><abbr title="Hypertext Markup Language">HTML</abbr></code> is supported.</p>	<pre><h1>Heading</h1> <h2>Sub-heading</h2> <p>Paragraphs are separated by a blank line.</p> <p>Two spaces at the end of a line
 produces a line break.</p> <p>Text attributes italic, bold, <code>monospace</code>.</p> <p>Horizontal rule:</p> <hr /> <p>Strikethrough:</p> strikethrough <p>Bullet list:</p> apples oranges pears <p>Numbered list:</p> lather rinse repeat <p>An example.</p> <p></p> <blockquote> <p>Markdown uses email-style > characters for blockquoting.</p> </blockquote> <p>Inline <abbr title="Hypertext Markup Language">HTML</abbr> is supported.</p></pre>	<h2>Heading</h2> <h3>Sub-heading</h3> <p>Paragraphs are separated by a blank line.</p> <p>Two spaces at the end of a line produces a line break.</p> <p>Text attributes <i>italic</i>, bold, monospace .</p> <hr/> <p>Strikethrough:</p> <p>strikethrough</p> <p>Bullet list:</p> <ul style="list-style-type: none">• apples• oranges• pears <p>Numbered list:</p> <ol style="list-style-type: none">1. lather2. rinse3. repeat <p>An example.</p>  <p>Markdown uses email-style > characters for blockquoting.</p> <p>Inline HTML is supported.</p>

Source: <https://en.wikipedia.org/wiki/Markdown> (2020.01.16)

References

- Cone, Matt (2018). *The Markdown Guide*. URL: <https://www.markdownguide.org/book>.
- Hiltzik, Michael A. (2014). *Dealers of Lightning. Xerox PARC and the Dawn of the Computer Age*. 23rd printing. Harper. ISBN: 978-0-88730-989-2.
- Lammers, Susan (1989). *Programmers at Work. Interviews with 19 Programmers Who Shaped the Computer Industry*. Tempus Books. ISBN: 1-55615-211-6.
- Seibel, Peter (2009). *Coders at Work. Reflections on the Craft of Programming*. Apress. ISBN: 978-1-4302-1948-4.
- Xie, Yihui, J. J. Allaire, and Garrett Grolemond (2018). *R Markdown. The Definitive Guide*. CRC Press. ISBN: 978-1-138-35933-8. URL: <https://bookdown.org/yihui/rmarkdown/>.