Creating Documents with R - October 2022

Kevin O'Brien

Step 1

- The relevant datasets, such as Harvest Block Details, Screening Information, is held on a master spreadsheet in a dedicated folder on Citrix.
- This Dataset is maintained by Emma Benson, John Landy, Marie Therese Roche and more.
- Citrix is the optimal location because of ease of access for the R programming environment,

Step 2

- This data can be processed by the R programming language.
- In the first instance, the entire data set is loaded into the R environment using the ***{readxl}*** R package.
- Individual sheets from the master data spreadsheet are loaded as required.

```r
library(readxl)

HB_details <- read_excel("MasterData.xlsx", sheet = "HB_Details")

Screening <- read_excel("MasterData.xlsx", sheet = "Screening")
```

Step 3

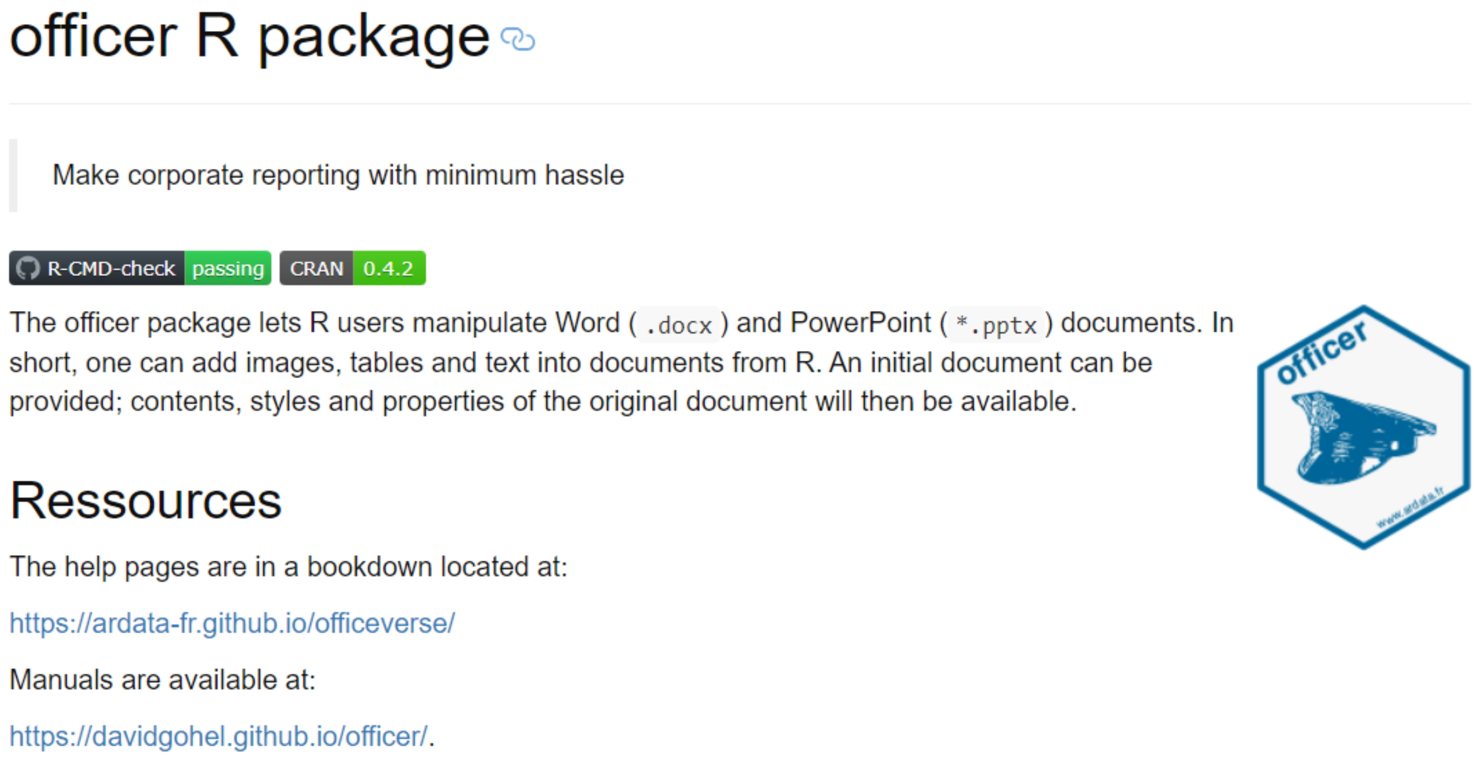
- The data is reduced to the particular felling licence applications required for the current run.
- There are multiple types of reporting requirements:
    - Single Screened In Application - NIS and Prescreening Report
    - Multiple FLs applied for jointly - NIS and Prescreening Reports
    - Single Screened out Applications - Prescreening Report Only
- The relevant type of report required is also recorded.
- The data undergoes some pre-processing.

Step 4

- One by one, data for each FL (or group of FLs) is extracted and processed by the main R programme.
- Using the ***officer*** R package, a word document is created and populated with information relevant to the each FL, along with formatted text.
- The ***officer*** R package can insert headings, images, tables and inserted sheets as appropriate. It can also specify landscape or portrait formats as required.
- Tables can be created using the ***flextable*** R package.

Officer R Package

Automated Generation of Word Documents with R



officer R package

> Make corporate reporting with minimum hassle

R-CMD-check passing CRAN 0.4.2

The officer package lets R users manipulate Word (`.docx`) and PowerPoint (`*.pptx`) documents. In short, one can add images, tables and text into documents from R. An initial document can be provided; contents, styles and properties of the original document will then be available.

Ressources

The help pages are in a bookdown located at:

https://ardata-fr.github.io/officeverse/

Manuals are available at:

https://davidgohel.github.io/officer/.

Officer R Package

Examples

```
my_doc <- read_docx() %>%
      body_add_par(value='Some text etc etc') %>%
      body_add_flextable(value=ft) %>%
      body_add_break()
```

- Formatted Text
- Add Images
- Replace Text in Existing Document

Officer R Package

Example

```
MyText_header <- fpar(ftext(MyText_header ,
        prop = fp_text(color = "black", font.family="Calibri",
        font.size = 14, bold = TRUE)))

MyText_1 <- fpar(ftext(MyText_1 ,
        prop = fp_text(color = "black", font.family="Calibri",
        font.size = 12, bold = FALSE)))

MyText_2  <- fpar(ftext(MyText_2 ,
        prop = fp_text(color = "black", font.family="Calibri",
        font.size = 12, bold = FALSE)))
```

Officer R Package

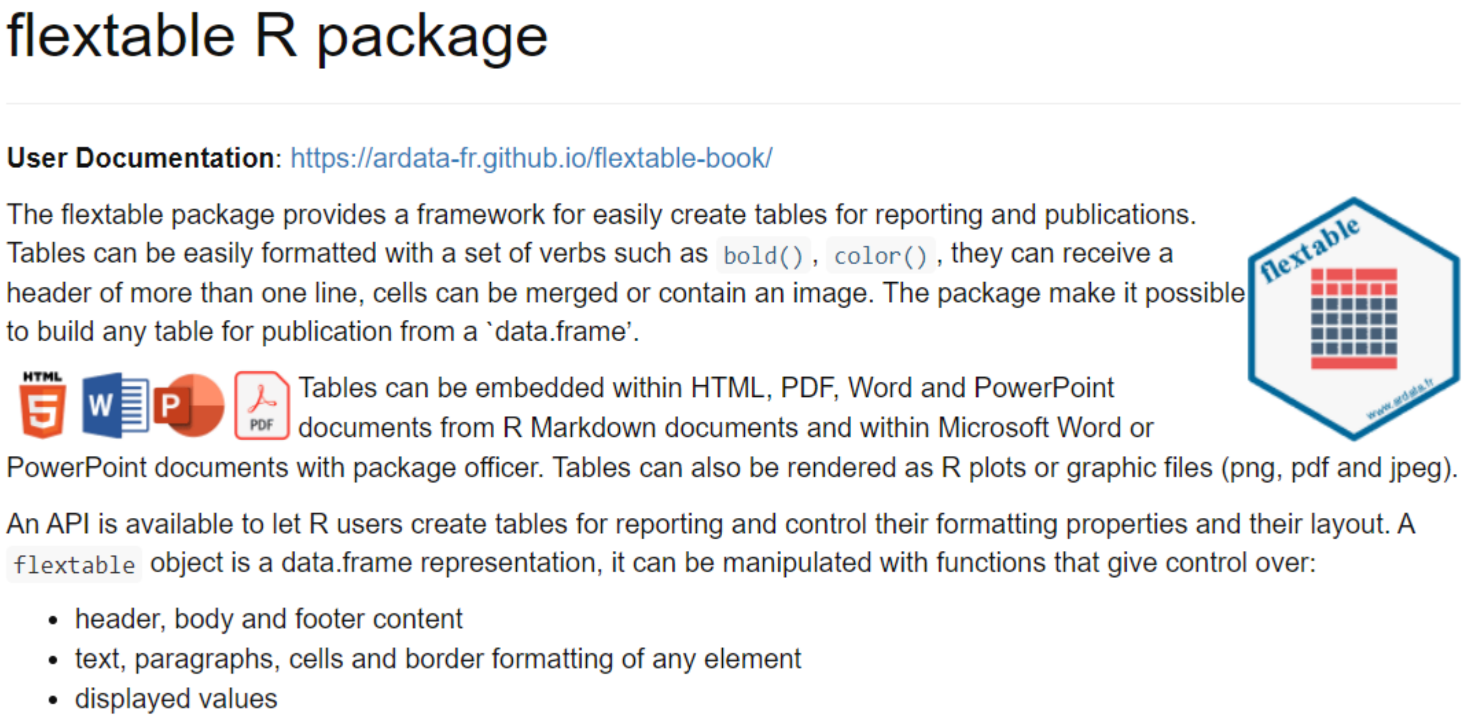
Example

```
my_doc <- my_doc  %>%
  body_add_fpar( MyText_header ) %>%
  body_add_par("", style = "Normal") %>%
  body_add_fpar( MyText_1 ) %>%
  body_add_par("", style = "Normal") %>%
  body_add_fpar( MyText_2) %>%
  body_add_par("", style = "Normal")


print(my_doc, target = Output_File_Name)
```

Flextable R Package

Automated Generation of Tables with R



flextable R package

User Documentation: https://ardata-fr.github.io/flextable-book/

The flextable package provides a framework for easily create tables for reporting and publications. Tables can be easily formatted with a set of verbs such as `bold()`, `color()`, they can receive a header of more than one line, cells can be merged or contain an image. The package make it possible to build any table for publication from a `data.frame'.

Tables can be embedded within HTML, PDF, Word and PowerPoint documents from R Markdown documents and within Microsoft Word or PowerPoint documents with package officer. Tables can also be rendered as R plots or graphic files (png, pdf and jpeg).

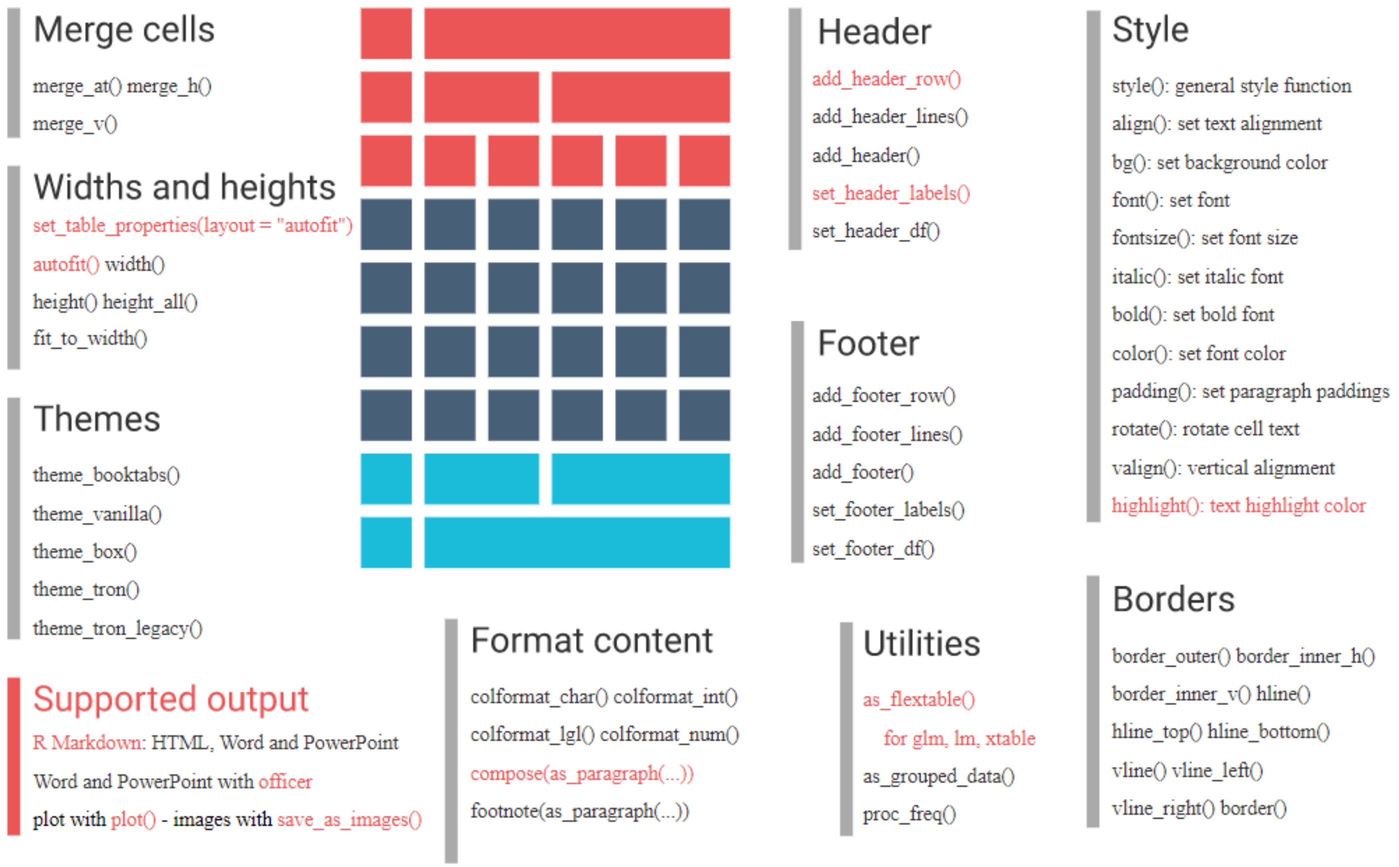
An API is available to let R users create tables for reporting and control their formatting properties and their layout. A `flextable` object is a data.frame representation, it can be manipulated with functions that give control over:

- header, body and footer content
- text, paragraphs, cells and border formatting of any element
- displayed values

(***Author: David Gohel***)

Flextable R Package

Automated Generation of Tables with R



Merge cells

merge_at() merge_h()

merge_v()

Widths and heights

set_table_properties(layout = "autofit")

autofit() width()

height() height_all()

fit_to_width()

Themes

theme_booktabs()

theme_vanilla()

theme_box()

theme_tron()

theme_tron_legacy()

Supported output

R Markdown: HTML, Word and PowerPoint

Word and PowerPoint with officer

plot with plot() - images with save_as_images()

Format content

colformat_char() colformat_int()

colformat_lgl() colformat_num()

compose(as_paragraph(...))

footnote(as_paragraph(...))

Header

add_header_row()

add_header_lines()

add_header()

set_header_labels()

set_header_df()

Footer

add_footer_row()

add_footer_lines()

add_footer()

set_footer_labels()

set_footer_df()

Utilities

as_flextable()

for glm, lm, xtable

as_grouped_data()

proc_freq()

Style

style(): general style function

align(): set text alignment

bg(): set background color

font(): set font

fontsize(): set font size

italic(): set italic font

bold(): set bold font

color(): set font color

padding(): set paragraph paddings

rotate(): rotate cell text

valign(): vertical alignment

highlight(): text highlight color

Borders

border_outer() border_inner_h()

border_inner_v() hline()

hline_top() hline_bottom()

vline() vline_left()

vline_right() border()

(*Author: David Gohel*)

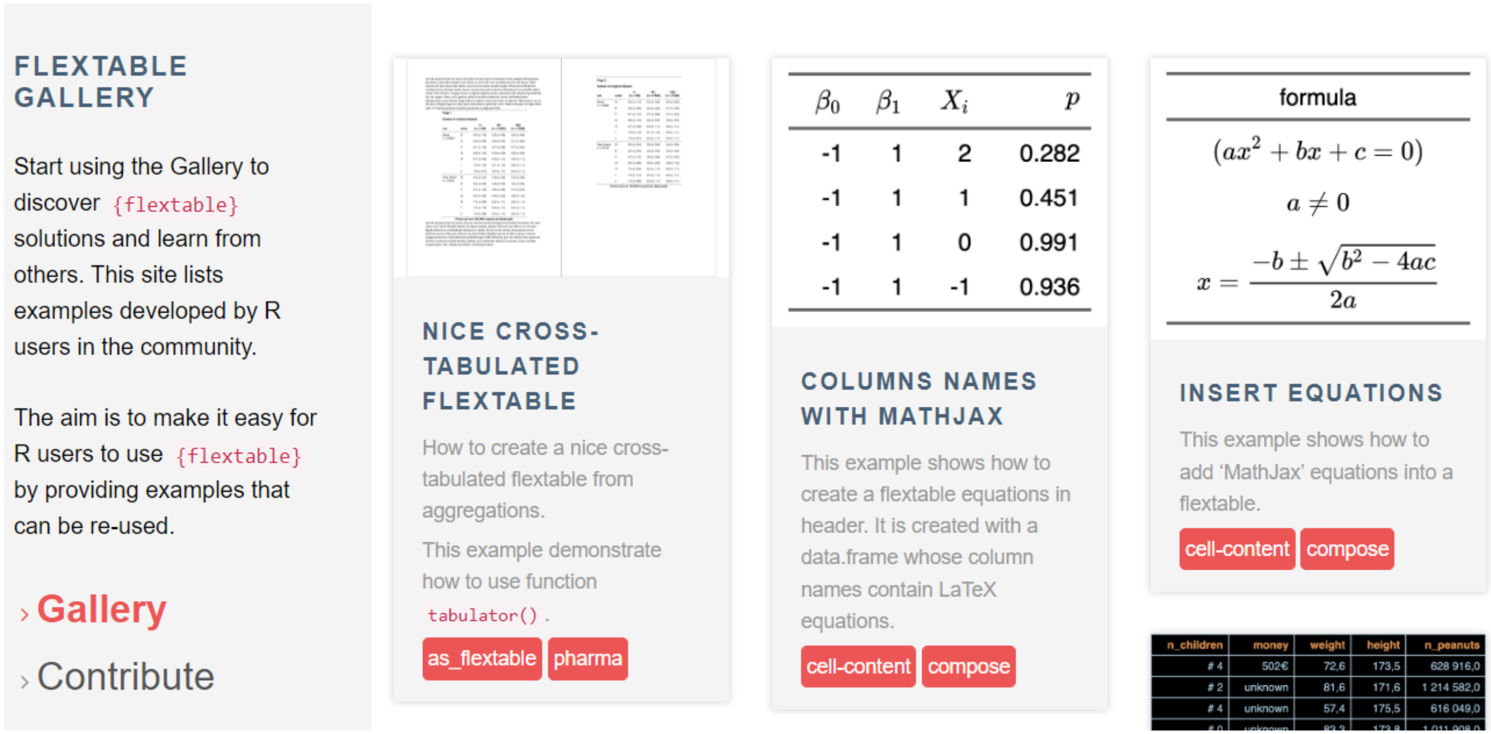
Flextable R Package

```r
This_OutPut_Table  <- This_Input_DF %>%
  flextable() %>%
  width(width=c(2.25,4.75)) %>%
  align( align = "left", part = "all" ) %>%
  font(fontname = "Calibri",part="all") %>%
  fontsize(size = 12, part = "body") %>%
  padding(padding = 3, part = "all" ) %>%
  delete_part(part="header") %>%
  border_remove() %>%
  border_outer( part="all", border = big_border ) %>%
  border_inner_h(part="all", border = big_border ) %>%
  border_inner_v(part="all", border = big_border )


my_doc <- my_doc %>%
  body_add_flextable(This_OutPut_Table,align="left") %>%
  body_add_par("", style = "Normal")
```

Flextable Gallery

`ardata-fr.github.io/flextable-gallery/gallery/`



FLEXTABLE GALLERY

Start using the Gallery to discover `{flextable}` solutions and learn from others. This site lists examples developed by R users in the community.

The aim is to make it easy for R users to use `{flextable}` by providing examples that can be re-used.

› Gallery

› Contribute

NICE CROSS-TABULATED FLEXTABLE

How to create a nice cross-tabulated flextable from aggregations.

This example demonstrate how to use function `tabulator()`.

as_flextable pharma

β_0	β_1	X_i	p
-1	1	2	0.282
-1	1	1	0.451
-1	1	0	0.991
-1	1	-1	0.936

COLUMNS NAMES WITH MATHJAX

This example shows how to create a flextable equations in header. It is created with a data.frame whose column names contain LaTeX equations.

cell-content compose

formula
$(ax^2 + bx + c = 0)$
$a \neq 0$
$x = \dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

INSERT EQUATIONS

This example shows how to add 'MathJax' equations into a flextable.

cell-content compose

n_children	money	weight	height	n_peanuts
# 4	502€	72,6	173,5	628 916,0
# 2	unknown	81,6	171,6	1 214 582,0
# 4	unknown	57,4	175,5	616 049,0
# 0	unknown	83,3	173,8	1 011 908,0

(Author: David Gohel)