Example Rmarkdown for data analysis

Dasapta Erwin Irawan & Willem Vervoort

10 January 2018

R Markdown demo

This is an R Markdown document. This is a useful tool to record workflows for open science, as you can show explanatory text, data, code for analyses and figures and tables. As a result the full workflow of the project can be described. You can use basic markdown coding, such as **bold** and *italic*, but you can also use more complex LateX scripting.

Lists are also possible:

- one
- two
  - three
  - four

Higher level headings

smaller headings are also possible

such as subheadings

and sub sub headings :-)

Can also be quite easily defined. This means you can organise your document in the way you want. The final extension is that you can actually write your whole scientfic article using Rmarkdown and there are a whole set of templates for popular publishers is available in the package **rticles**

Excercise

- Write a list of three of your favourite things, of which one is written in bold text
- Use a heading and a subheading to write: Methodology, Field Site as in a journal paper

An example of a workflow

We will use the example data set again, which we have described earlier. Here we are loading the particle size data, which is the most regular data to work with (the least missing data).

The code below is reproduced in the pdf document, which is the output of the markdown ("rmd") file. It is reading in the data and showing a snapshot of the data (as a tibble).

At the top of this document, I have specified the root directory where R should be looking for the data. Therefore, I only need to only indicate the specific folder and the file name.

```
ParticleSize <- read_csv("OriginalDataFolder/Willem/SoilParticles.csv")
```

```
## Parsed with column specification:
## cols(
##   Distance = col_integer(),
##   Depth_top = col_double(),
##   Depth_bottom = col_double(),
##   Clay_perc = col_double(),
##   Silt_perc = col_double(),
##   Fsand_perc = col_double(),
##   Csand_perc = col_double(),
##   Fgravel_perc = col_double(),
##   Cgravel_perc = col_double()
## )
```

```
print(ParticleSize)
```

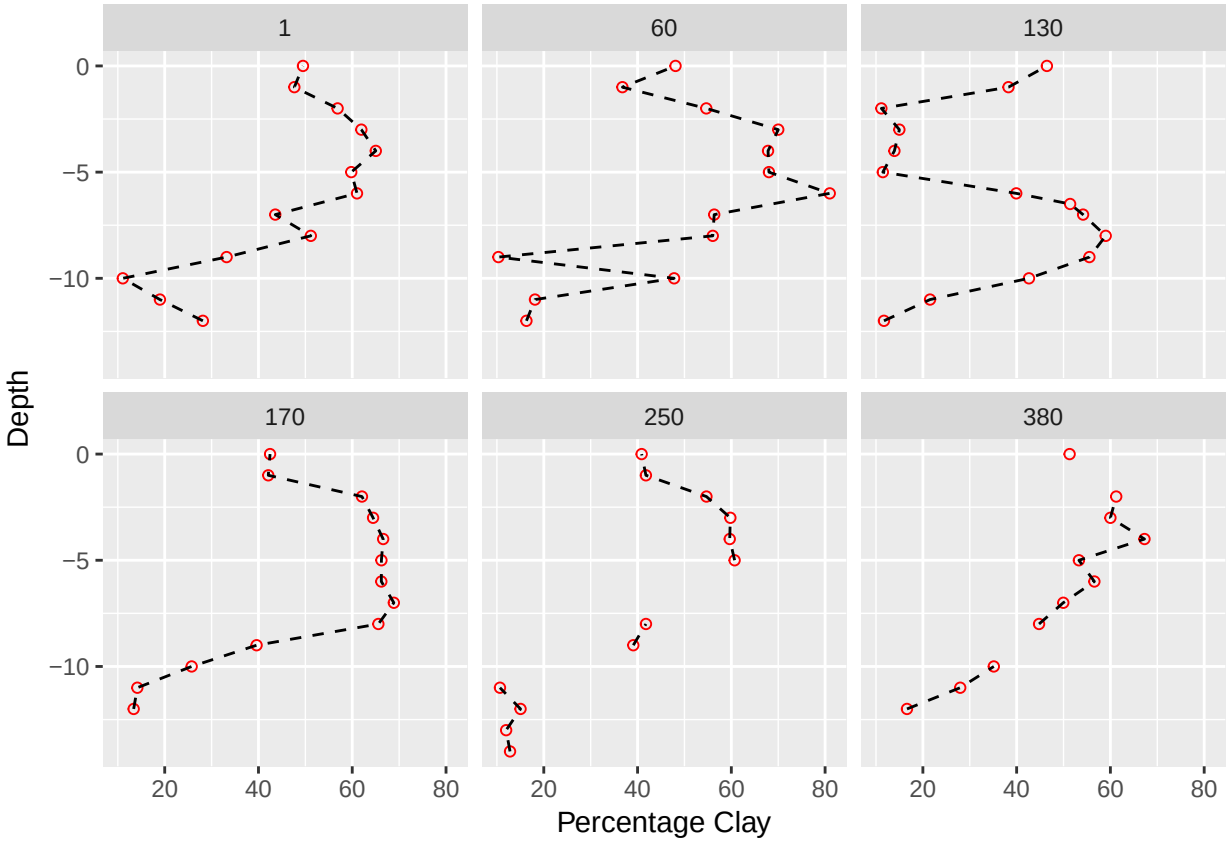
```
## # A tibble: 81 x 9
##    Distance Depth_top Depth_bottom Clay_perc Silt_perc Fsand_perc
##       <int>     <dbl>        <dbl>     <dbl>     <dbl>      <dbl>
##  1        1         0            1     49.49     15.92      28.27
##  2        1         1            2     47.65     23.19      25.06
##  3        1         2            3     56.88     16.31      18.78
##  4        1         3            4     61.94     12.67      16.33
##  5        1         4            5     65.02     12.83      12.62
##  6        1         5            6     59.79     15.20      13.97
##  7        1         6            7     61.01     16.86      12.38
##  8        1         7            8     43.58      3.72       8.54
##  9        1         8            9     51.16      0.33       7.38
## 10        1         9           10     33.20      3.39       6.01
## # ... with 71 more rows, and 3 more variables: Csand_perc <dbl>,
## #   Fgravel_perc <dbl>, Cgravel_perc <dbl>
```

Including Plots

The nice thing about Rmarkdown is that you can also include plots. For example we can repeat part of the plot that we have in the slides, showing the distribution of the clay percentage at each location.

Here, I have added to the code the statement "echo=FALSE" for the chunk. This means that the code is actually not visible in the pdf document that you are reading, but it is in the original "rmd" file.

```
## Warning: Removed 5 rows containing missing values (geom_point).
```



Doing some calculations

We could now aggregate the data across depths or across locations and show the results in a table. Using the package `pander` in the Rmarkdown document allows you to make nice looking tables.

```
ByDepth <- ParticleSize %>%
  group_by(Depth_top) %>%
    summarise(Clay = mean(Clay_perc, na.rm=T)) %>%
  arrange(desc(Clay))

pander(ByDepth)
```

Depth_top	Clay
6	60.94
4	56.72
3	55.2
7	54.58
5	53.24
8	53.06
6.5	51.41
2	50.12
0	46.45
1	41.31
9	35.55
10	32.47

Depth_top	Clay
11	18.57
12	16.86
14	12.8
13	11.97

```
ByLoc <- ParticleSize %>%
  group_by(Distance) %>%
    summarise(Clay = mean(Clay_perc, na.rm=T)) %>%
  arrange(desc(Clay))

pander(ByLoc)
```

Distance	Clay
170	49.03
60	48.56
380	47.65
1	45.22
250	37.4
130	33.72

As you can see the whole workflow is completely clear from the code and the results that are presented, thus producing fully reproducible research.

Activity

- Create your own document to make a plot of the fine sand percentage and table with fine sand by depth and location from the data.