PharmaSUG China 2018 – Paper AD-38

Document Automation using R Markdown

Wenfang Li, Boehringer Ingelheim (China) Investment Co., Ltd, Shanghai, China

ABSTRACT

R markdown is an extensible markdown language under R Studio. It could provide an authoring framework for data science. Though nowadays SAS is the main program used in clinical research, smartly taking benefits from other software could greatly improve work efficiency as well. The scope of this paper is to explore how a little bit of R markdown can make work easier. The following example may be rather specific. Along with drug development and trial submission process there requires various supporting documents. These documents follow certain standards but are highly time consuming and require very high quality. To improve work efficiency and quality, we proposed a process to automate documents applying R markdown techniques. In addition, we developed a tool (program package dram_1.0) with user manual to automate the analysis data reviewer's guide, which is part of the trial documentation/ submission package. By running the tool, the guide will be generated automatically including required contents, tables, figures, formats etc. In this paper, we will briefly introduce R markdown, describe the document automation process and demonstrate the tool package which could automate an accurate document in "1-click".

INTRODUCTION

RStudio is a family of powerful, cost-effective disk recovery software. Originally developed by R-Tools Technology, Inc. RStudio provides popular open source and enterprise-ready professional software for the R statistical computing environment. R markdown is an extensible markdown language under RStudio. It provides an authoring framework for data science, which is widely used in generating professional and nice looking documents. The benefits and advantages are pretty attractive. First of all, it is easy to use. With a user friendly interface, one can easily learn and build an r markdown file in the R studio environment. Secondly, it is easy to learn. Training materials of all levels are available online and in store. Furthermore, R markdown is very flexible and powerful, one can use a single R markdown file to save and execute code and generate high quality reports. It is free, open source and has an incredible amount of libraries and functions available to install, expanding the language well beyond the base packages, which will providing people tools to get work done quickly. Worldwide, there is a large amount of R and R markdown users. For most problems that user might meet during program developing, solution or support could be easily found.

Document automation (also known as document assembly) is the design of workflows and programs that assist in the creation of electronic documents. Nowadays, along with drug development and trial submission process there requires various supporting documents. These documents follow certain standards but are highly time consuming and require very high quality. Without a document automation system the production of repetitive documents can lead to hours of unnecessary work. With documents are generated automatically, one can rest assured knowing that document automation and assembly processes are accurately and consistently followed, and that the right information is collected and validated. Automated document workflows optimize the document creation process, which increases workplace efficiency and compliance. It could greatly improve scalability, improve engagement, improve compliance, reduce costs, improve consistency and eliminate errors associated with manual document creation. Firstly, document automation will help greatly reduce risks by removing the human element from processes that are prone to mistakes. Secondly, in many cases, document automation has additional benefits, such as improved compliance with state and local regulations. Furthermore, important cost savings can be realized because overworked staff can concentrate on reviewing the most important pieces of communications without the need for additional workload. Also, it is a cumulative gaining process. Though, efforts are needed to create the file template at the very beginning, it would be better and better as more resources are cumulating. Per limit of resources, sometimes, the process is more like a combination of partial auto and partial manual.

The purpose of this paper is to demonstrate that with a little bit of background and a creative mind, you can have a large impact on daily work. To better illustrate r markdown techniques and document automation process, and show how the theory could be used in reality, we will briefly describe the generation of dram_1.0 package, which is used to automate the analysis data reviewer's guide. Analysis data reviewer's guide is part of the trial documentation/ submission package. It follows company standard operation procedure (SOP) and contains description information about datasets used for analysis. The basic elements of the document includes cover page, which includes basic information about the trial and statistician etc., table of contents (TOC), descriptive contents, supportive figures or tables, and company standard formats such as header, footer, logos, proprietary confidential information etc. In the following discussion, we will show how to automate the analysis data reviewer's guide with R and R markdown using R studio.

AUTOMATION PROCESS (GENERAL)

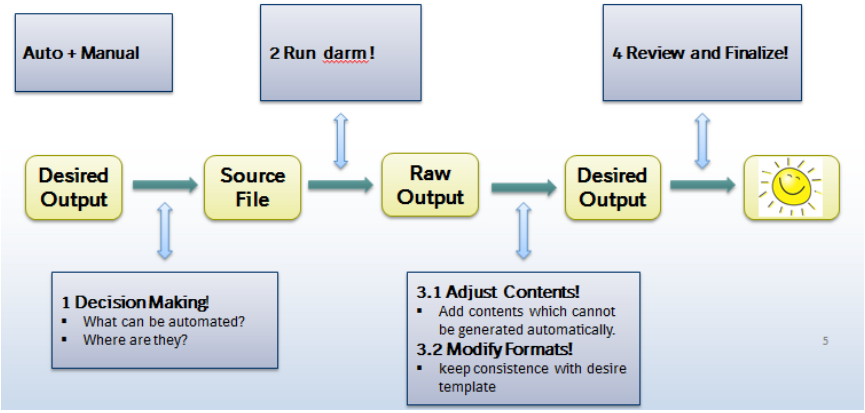

Figure 1. Automation Process

The motivation is really twofold: efficiency (maximizes the reusability of code, minimize coping and pasting errors) and reproducibility (maximize the number of people and computers that can reproduce findings). Overall, the automation process includes 4 key steps:

1. Decision Making
2. Generate File template and Run Program
3. Adjust Contents and Modify Formats
4. Review and Finalize

In general, prior to take the benefits of automation, people need to firstly generate the file template, if not already generated. Though some efforts are necessary in this step, it will be better use and save more in a long run. By taking an overview of the desired documents, people need to firstly identify information that can be obtained automatically. Generally speaking, there are several types: 1. Parts that remain the same or not often change, i.e. standard formats, header, footer, logo, proprietary confidential information; 2. Parts that can be stored and called from a data source i.e. Information saved in excels files/ figures such as study/ project description, cover page information and pre-defined tables; 3. Parts that can be inserted and executed from reproducible code such as Figures, tables, calculations reproducible by R; 4. Parts that could be summarized or generated by r markdown code, such as table of contents. We also need to identify what part still need to manually customize. After that, we need find source or location where those information stored, which we call them source files. Based on all these, we will apply the r markdown techniques and create a file template, with all necessary codes embedded. We run the code and a raw output will be generated. Then we customize the raw output, adjust necessary contents and formats. Finally, for accuracy purpose, we'd better do a review and the document is finalized.

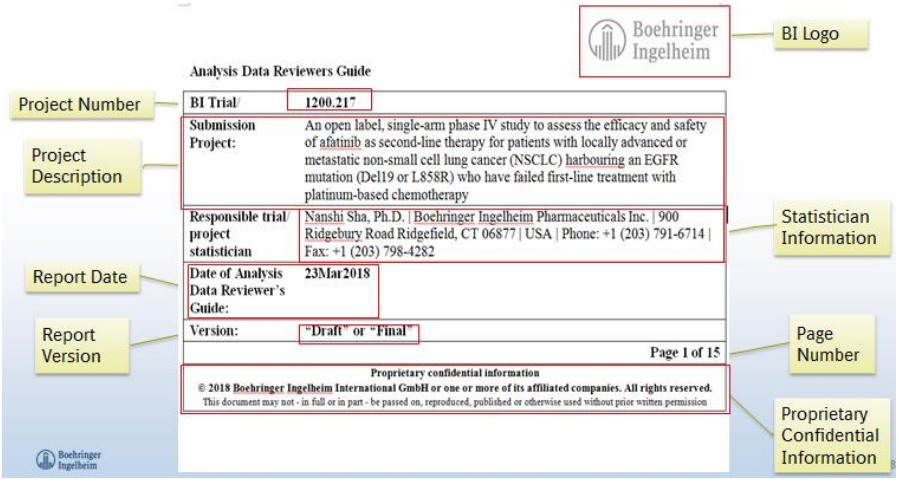

Figure 2. Sample Cover Page

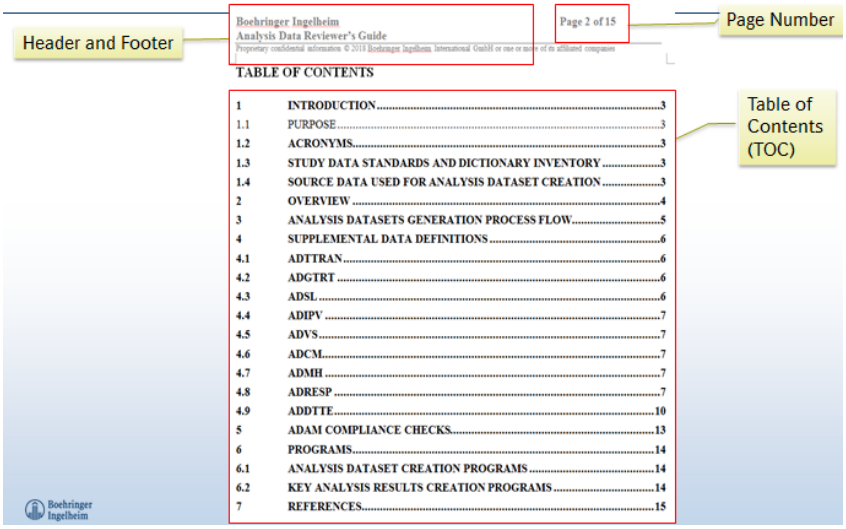

Figure 3. Sample Table of Contents

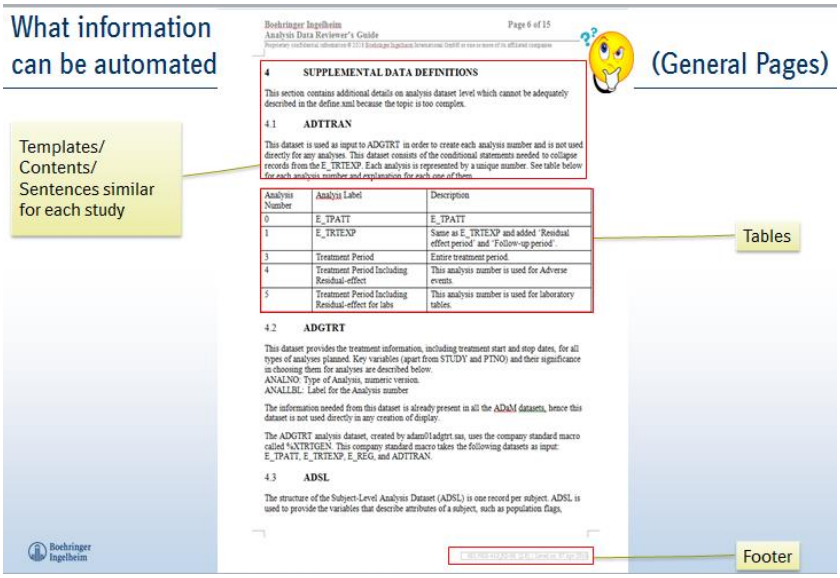

Figure 4. General Page with session, contents and tables

GENERATE FILE TEMPLATE WITH R MARKDOWN TECHNIQUES

The basic setup is to write an Rmd file that will serve as a template, and then a short R script that looks over each data file (using library (knitr)). The knit function then turns the Rmd into documents or slides (typically in pdf, html, or docs) by taking file metadata as a parameter.

Installation R studio's interface with library (rmarkdown) is evolving rapidly, installing the current version of R studio is highly recommended. The codes to install r-markdown packages and create a new R - markdown file are as follow:

```
In R studio: File -> New File -> R markdown
In r: install.packages("rmarkdown")
```

Install libraries One of the top benefits of using R markdown is there are various libraries packages with nice functions and features. To successfully generate a file template, we need to install necessary packages. Function "Install.packages ()" is used to install necessary packages. The readxl package makes it easy to get data out of Excel and into R. The R package knitr is a general-purpose literate programming engine, with lightweight API's designed to

give users full control of the output without heavy coding work. The goal of kableExtra package is to help you build common complex tables and manipulate table styles. It allows us to construct Complex Table with 'kable' and Pipe Syntax. Package cellranger will Translate Spreadsheet Cell Ranges to Rows and Columns

````
<!---------------------------------->
<!-- SECION A: INSTALL PACKAGES --->
<!---------------------------------->

```{r echo=FALSE, message=FALSE, warning=FALSE, error=FALSE}
## Remove the # in front to install at first time use
    # install.packages("readxl")
    # install.packages("knitr")
    # install.packages("kableExtra")
    #install.packages("cellranger")
```
````

Reference documents R markdown could generate various file types, including pdf, html, and word and so on. Basic knowledge and installation of Latex, which is a high quality typesetting system, is required to generating a pdf file, where various formats could be embedded directly though codes. For easier customization and later modification, here we will generate a word file. Unlike by using latex that formats could be embedded directly though codes, word documents are not plain-tex files, so you cannot create a template like most orther formats. However, the most notable feature of generating word documents is the word template, which is also known as the “style reference document”. You can specify a document to be used as a style reference in producing a *.docx file (a word document). This will allow you to customize formats of the output file. For best results, the reference document should be a modified version of a .docx file produced using r markdown or pandoc. One can open the word document and edit the styles in it. The path of such a document can be passed to the reference_doc argument of the word_document format. Pass “default” to use the default styles. [1] You customize the style of the word output applying this reference file.

```
---
fontsize: 12pt
geometry: margin = 1in
output:
  word_document:
    reference_docx: template1.docx
    toc: yes
---
```

Set up Source file location need to be specified to teach the program the location of the sources applying the following codes.

````
<!---------------------->
<!-- SECION B: SETUP --->
<!---------------------->

```{r echo=FALSE, message=FALSE, warning=FALSE, error=FALSE}
## varaible set up
    tsapFile<-"8-01-tsap-ads-plan.xls"          # inset source file name
    TrialVersion<- " \"Draft\" or \"Final\""    # inset version type such as
"Final"" or "Version" as defined
    TotalPN<-"17"                               # inset total page number
here
```
````

Set up code modules All codes are set up in modules for better modification. Each Section refers to a sub program related to the section. Thus one can easily add or delete a section. Contents could be edit directly in the main program. Session or sub program can be linked and run in the main program using {r child = '***.Rmd'}. These codes are used to nest child documents into the main documents. Figures, tables can be inserted and displayed applying the following sample codes

```
  a.          Insert figures: ![](BI LOGO.png)
  b.          Insert table: table1 <-read_excel(ExcelFile,range="A1:B6")
  c.          Display table: kable(table1)

<!--------------------------->
<!-- SECION C: MAIN BODY -->
<!--------------------------->

<!-- Section 0: COVER PAGE -->
```{r child = 'rmd/2-0-Analysis Data Reviewers Guide template.Rmd'}
```

<!-- SECTION 1: INTRODUCTION -->
```{r child = 'rmd/2-1-INTRODUCTION.Rmd'}
```

<!-- SECTION 2: OVERVIEW -->
```{r child = 'rmd/2-2-OVERVIEW.Rmd'}
```

<!-- SECTION 3: ANALYSIS DATASETS GENERATION PROCESS FLOW-->
```{r child = 'rmd/2-3-ANALYSIS DATASETS GENERATION PROCESS FLOW.Rmd'}
```

<!-- SECTION 4: SUPPLEMENTAL DATA DEFINITIONS -->
```{r child = 'rmd/2-4-0-SUPPLEMENTAL DATA DEFINITIONS.Rmd'}
```
```{r child = 'rmd/2-4-1-ADTTRAN.Rmd'}
```
```{r child = 'rmd/2-4-2-ADGTRT.Rmd'}
```
```{r child = 'rmd/2-4-3-ADSL.Rmd'}
```
```{r child = 'rmd/2-4-4-ADIPV.Rmd'}
```
```{r child = 'rmd/2-4-5-ADVS.Rmd'}
```
```{r child = 'rmd/2-4-6-ADCM.Rmd'}
```
```{r child = 'rmd/2-4-7-ADMH.Rmd'}
```
```{r child = 'rmd/2-4-8-ADRESP.Rmd'}
```
```{r child = 'rmd/2-4-9-ADDTTE.Rmd'}
```

<!-- SECTION 5: ADAM COMPLIANCE CHECKS -->
```{r child = 'rmd/2-5-ADAM COMPLIANCE CHECKS.Rmd'}
```

<!-- SECTION 6: PROGRAMS -->
```{r child = 'rmd/2-6-PROGRAMS.Rmd'}
```

<!-- SECTION 7: REFERENCES -->
```

```
```{r child = 'rmd/2-7-REFERENCES.Rmd'}
```

<!------------------>
<!-- PROGRAM END -->
<!------------------>
```

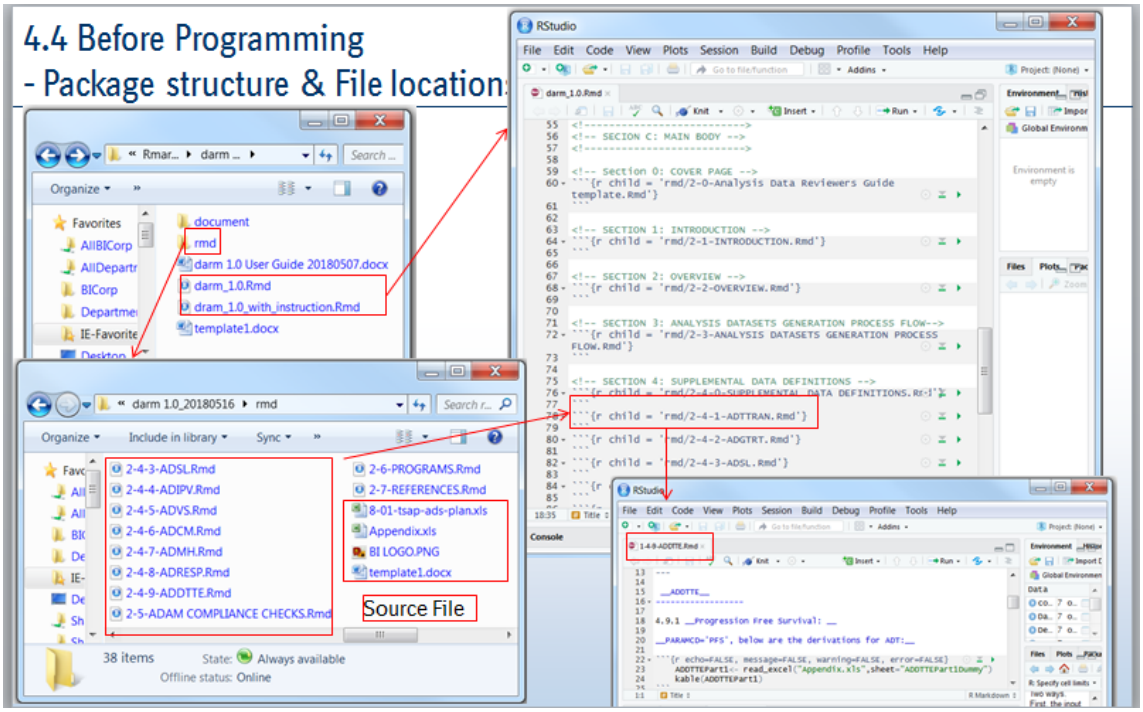

Figure.5 folder structure

The created program package is called dram 1.0. By running the program, a draft documents will be generated automatically.

Post Modification Further modification and review is recommended to assure the file quality. Applying all these steps, a professional analysis data reviewer's guide as specified earlier will be generated.

OTHER APPLICATION OF R MARKDOWN

In the previous generation processes, we only showed a tinny area of applying the R markdown techniques. There also various other applications.

It could be used to **build websites and create html widgets**, render multiple R Markdown documents to a website of multiple pages using "rmarkdown::render_site". The minimal requirements is index.(R)md and _site.yml. RStudio (currently preview version, e.g. 0.99.1242) can automatically detect R Markdown website projects, and you can use the Build button to build the website using demo site.Rproj.

Interactive Shiny documents you can embed shiny apps in R Markdown. You can write UI/server logic code directly in R code chunks, instead of separate R scripts. The output document is essentially a Shiny app, so requires a running R session (unlike typical R Markdown output documents, which are static HTML/PDF files).

Include code chunks of other languages. Although knitr was designed primarily for executing R code in dynamic documents, it also has limited support for other languages, such as C, Fortran, C++/Rcpp, Python, and so on.

CONCLUSION AND DISCUSSION

R is one of the top tools for programming. Though nowadays SAS is the main software used in clinical research, smartly take advantages of other software such as R can reduce the repetitive tasks and save time.

For this tool, little efforts to learn the basic enough to start taking the benefits, and one doesn't need to master R before taking the benefits. We introduced basic knowledge about r markdown, document automation process and provided an application example - package dram 1.0 for generating analysis data reviewer's guide. The process is a combination of auto and manual.

The advantages are obvious. It is efficient and could minimize the replicated work. It is accurate and could minimize manual edition errors. It is user friendly, that the draft document could be generated in 1 click. It is modifiable. As the program is modularized and output is in word formats, people don't need too much advanced programming skills to do modification. Also, the generated report has a professional appearance. As the library of

Well, in reality there are still limitations that need further explore to make it better. The power of the tool is highly reply on data source availability. By software default, table of contents will be generated on the first page. If cover page is required and need to be in the first page, the position of table of contents and cover page need to be manually shifted. But, it only takes seconds and should not be a big burden.

Future work could be done to find more application of R markdown and other techniques to improve efficacy and solve problems

REFERENCES

1. R Markdown: the Definitive Guide https://bookdown.org/yihui/rmarkdown/word-document.html

ACKNOWLEDGMENTS

The author is thankful to Jingwei Gao, Yun Ma, and Xuebo Pi at Boehringer Ingelheim (China) Investment Co., Ltd for their encouragement and support.

RECOMMENDED READING

- R Markdown from R Studio
- R Markdown Reference Guide
- darm 1.0 User Guide

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Wenfang Li, M.S., M.S., BS, BS
Boehringer Ingelheim (China) Investment Co., Ltd
Wenfang.Li@boehringer-ingelheim.com