PharmaSUG 2024 - Paper DV- 327

A R Markdown Structure for Automatically Generating Presentation Slides

Junze Zhang, Chuanhai Tang, and Xiaohui Wang, Merck & Co., Inc., Rahway, NJ, USA

ABSTRACT

The paper introduces a method to structure R Markdown, separating the code to define the output format and the code to conduct the analysis and compile the output using the predefined format. This paper will give an example of generating PowerPoint slides, though this method can be applied to various report formats. Traditionally, R Markdown is an authoring framework executing code and generating reports simultaneously. This method provides additional structure to the traditional method, which helps eradicate the work of manual data transfer to slides. For example, instead of regularly compiling slides for upper management to review the study result, statistical programmers can help stats utilize this method to compile the R markdown for stats to simply insert the code and output the result in the desired PowerPoint slides format. Consequently, this generates updated slides with the most recent data, requiring little to no modifications. The primary objective of this paper is not to provide more detailed R Markdown techniques for creating complex PowerPoint slides, but to propose a practice that enhances the structure of your R Markdown program for improved maintainability, user-friendliness, and repeatability, thereby easing your tasks.

INTRODUCTION

Statistical programmers often have the demand to create periodic presentations for sharing updates with upper management or collaborating functional teams (biostatistical, clinical, etc.). The manual creation of periodic reports can be tedious and time-consuming. Typically the standard process may include an output mockup provided or updated by the statistician, the creation or transferring of analysis datasets, and the programming efforts of the statistical programmer utilizing certain standard reporting programs (e.g. SAS macros) to generate the reports.

R Markdown has offered a potential solution for automatic report generation to reduce the amount of both the workload and time required. However, the traditional approach may not fully meet our needs, particularly for presentation reports. The limitations of traditional R Markdown will be discussed in detail in a later section. Following that we will introduce the Header Data Report (HDR) structure, a method that distinctly separates data manipulation from report writing and formatting for better code structuring and reusability.

R MARKDOWN

WHAT IS MARKDOWN

Markdown syntax defines notations for formatting, allowing you to add formatting to plaintext. Text files written in this syntax are referred to as Markdown files. Markdown Applications can render these files into well-formatted documents.

Following is an example:

Text in Markdown syntax:

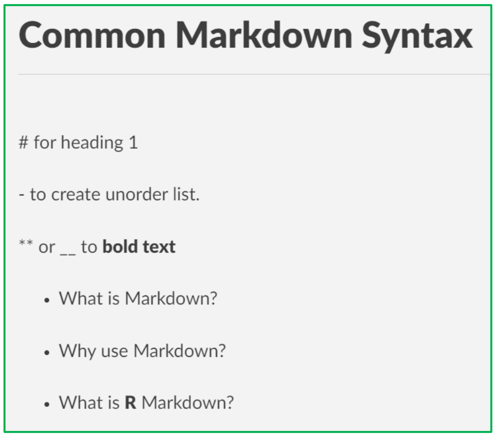
```
# Common Markdown Syntax
\# for heading 1

\- to create unorder list.

\*\* or \_\_ to **bold text**

- What is Markdown?
- Why use Markdown?
- What is **R** Markdown?
```

Rendered output:



Common Markdown Syntax

for heading 1

- to create unorder list.

** or __ to **bold text**

- What is Markdown?
- Why use Markdown?
- What is **R** Markdown?

Display 1. Rendered Markdown output

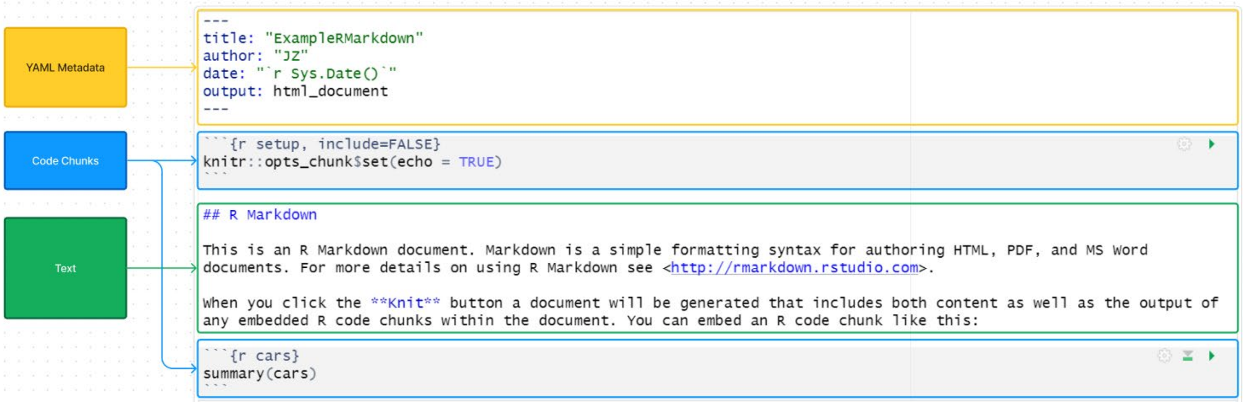
WHY USE MARKDOWN

You may wonder why we need to code for formatting when applications like Microsoft Word offer a graphical user interface and formatting options through buttons. The answer lies in the improved consistency and reproducibility that coded formatting achieves. When we program the format, the formatting for all outputs remains the same. For example, if you define how a heading should appear in a template and everyone in your company uses that template, every rendered output's heading will be exactly the same. This eliminates concerns about formatting inconsistencies.

WHAT IS "R MARKDOWN"

R Markdown, an extension of Markdown, introduces the ability to run code within the same file. Consequently, you can include the output of executed code in your final rendered documents.

A R Markdown file (.rmd) primarily consists of three parts: YAML metadata, which defines how we build and render the R Markdown file; code chunks, which run code for data manipulation and TLF generation; and text, which uses Markdown syntax to indicate context and formats.



~~~
---
title: "ExampleRMarkdown"
author: "JZ"
date: "`r Sys.Date()`"
output: html_document
---
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word
documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of
any embedded R code chunks within the document. You can embed an R code chunk like this:
```{r cars}
summary(cars)
```
~~~

Display 2. R Markdown Example Code

WHY USE "R MARKDOWN"

Demonstrating data derivation and result acquisition becomes simpler when code, outputs, and text are consolidated in one location. This approach provides readers with a straightforward view of the process, allowing them to rerun each step as needed.

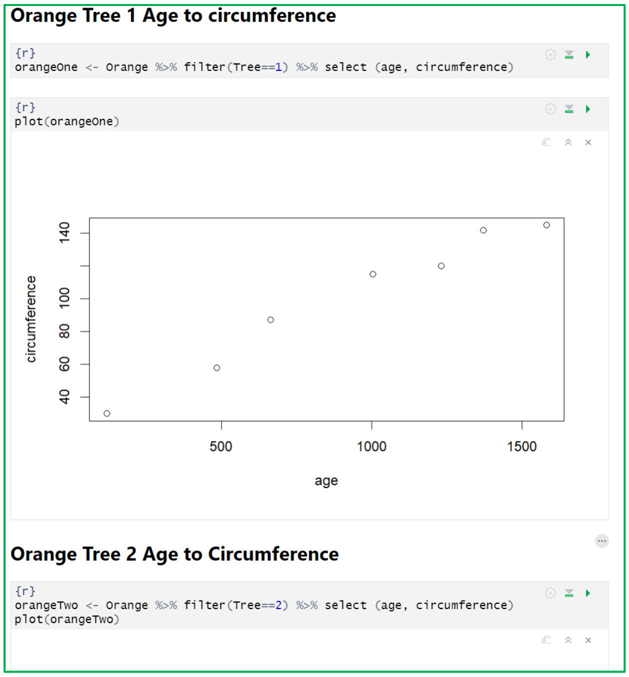
R Markdown takes reproducibility to the next level. It ensures consistency not only in formatting but also in data. For instance, if a R Markdown file is reading an ADEX dataset to generate documents and there are updates in your ADEX dataset that you want reflected in the output documents, all you need to do is to simply rerun your R Markdown. It will regenerate the documents with the updated ADEX data.

LIMITATIONS OF TRADITIONAL R MARKDOWN

Generating slides or other presentation-oriented documents can prove challenging with the traditional R Markdown approach. It integrates R code, R outputs, and markdown text, forming a narrative heavily dependent on code. It typically focuses on demonstrating the derivation of data and the process of generating graphs. Because these steps are not crucial for presentations, demonstrating these procedures becomes too burdensome and complicated.

Documents often become difficult to write and maintain as the programming process can take longer using the traditional method. Firstly, the code, being a significant part of the document, requires exceptional cleanliness and organization. However, in this instance, we are primarily concerned with the results, so we can arrange the code in a manner that is most conducive to implementation. Secondly, in the traditional method, data processing and visualization codes may get jumbled together, making the document hard to maintain. For statistical programmers, it is like creating datasets and TLFs in the same file, alternating back and forth.

As a result, a simpler structure that separates data manipulation from output presentation is more desirable. This structure will promote efficient coordination between the two components, reduce duplication, and facilitate reuse across various studies.



Orange Tree 1 Age to circumference

```
{r}
orangeOne <- Orange %>% filter(Tree==1) %>% select (age, circumference)
```

```
{r}
plot(orangeOne)
```

Orange Tree 2 Age to Circumference

```
{r}
orangeTwo <- Orange %>% filter(Tree==2) %>% select (age, circumference)
plot(orangeTwo)
```

Display 3. Traditional R Markdown Example Code

HEADER DATA REPORT STRUCTURE (HDR)

HDR structure is a new structure for organizing R Markdown, composed of three sections: YAML Header, Data Manipulation, and Report Writing. YAML Header contains metadata of the document like regular R Markdown, detailing elements such as the title, author, date, and output format. The Data Manipulation section encompasses the essential R code to produce the necessary data. The Report Writing and Editing section arranges outputs and texts with Markdown syntax.

A practical use case proceeds as follows:

Programmers obtain a PowerPoint slide design template from the company's branding department. Subsequently, they create a R Markdown file following the HDR structure, and specify in the header that the R Markdown will reference this template's style.

Statisticians then draft R code to access and derive the necessary data, and subsequently copy their code into the Data Manipulation section of the R Markdown file.

Programmers or statisticians then write slides in R Markdown syntax, which includes the insertion of TLFs and explanatory text.

In the end, they run the R Markdown file to produce the required slides. When the data is periodically updated, the statistician only needs to rerun the R Markdown file and the updated slides will be ready for presentation.

HEADER

The YAML Header contains the YAML metadata of the document. Information is written as key-value pairs, specifying details like output format. For example, to generate a PowerPoint presentation, the output format should be specified in YAML as follows:

```
---
output:
  powerpoint_presentation:
---
```

Additional information can be customized in the header, such as titles, subtitles, and reference templates, as shown in the following example:

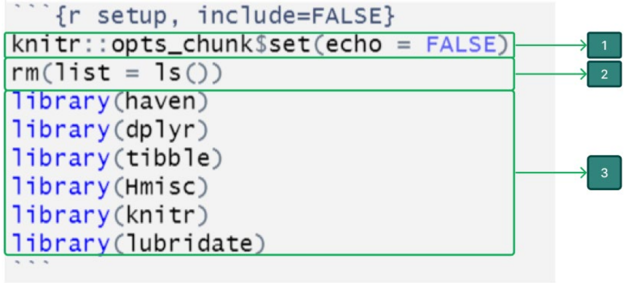
```
---
title: "Study AB8888 Monthly ORR Update"
output:
  powerpoint_presentation:
    reference_doc: custom-reference.pptx
subtitle: "Data Cutoff `r Sys.Date()`"
---
```

DATA MANIPULATION

This section includes all codes needed to generate the required data frame for presentation. It might involve environment setup, data extraction, data cleaning, and data processing. In practice, this process is seamless. Statisticians should insert their code here. They can use their preferred language, such as R, Python, or SQL, to manipulate and analyze data. For simplicity, we will concentrate on an example using R.

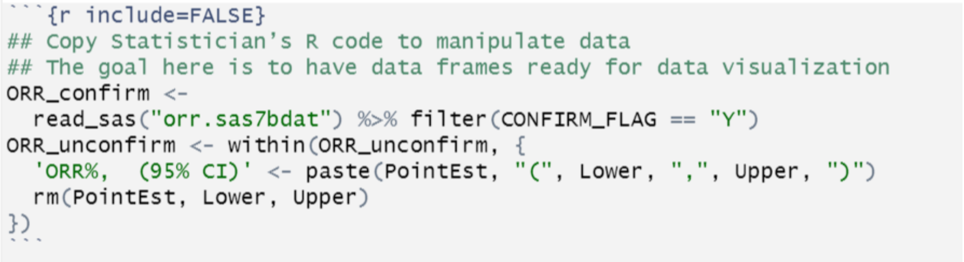
First, we usually need to set up the workspace. For this example, we have three steps.

1. Set the knitr global option. In this example, we set echo = FALSE, so the code won't be included in the final documents by default.
2. Clear the workspace.
3. Import necessary libraries.



~~~
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE)
rm(list = ls())
library(haven)
library(dplyr)
library(tibble)
library(Hmisc)
library(knitr)
library(lubridate)
```
~~~

Secondly, we add the statisticians' code that processes data. We can have multiple code chunks or everything in one code chunk, depending on statisticians’ preference.



~~~
```{r include=FALSE}
## Copy Statistician’s R code to manipulate data
## The goal here is to have data frames ready for data visualization
ORR_confirm <-
  read_sas("orr.sas7bdat") %>% filter(CONFIRM_FLAG == "Y")
ORR_unconfirm <- within(ORR_unconfirm, {
  'ORR%,  (95% CI)' <- paste(PointEst, "(", Lower, ",", Upper, ")")
  rm(PointEst, Lower, Upper)
})
```
~~~

REPORT WRITING AND EDITING

Once all necessary data frames are generated, they can be organized and presented using R Markdown syntax. For presentation slides, `#` is used to indicate a new slide. The following example demonstrates how to create a slide and insert the result of an R code block into the slide:

~~~
## AB8888

Some brief introduction about the study AB8888.

## AB8888 ORR (Confirmed)

```{r}
kable(ORR_confirm)
~~~

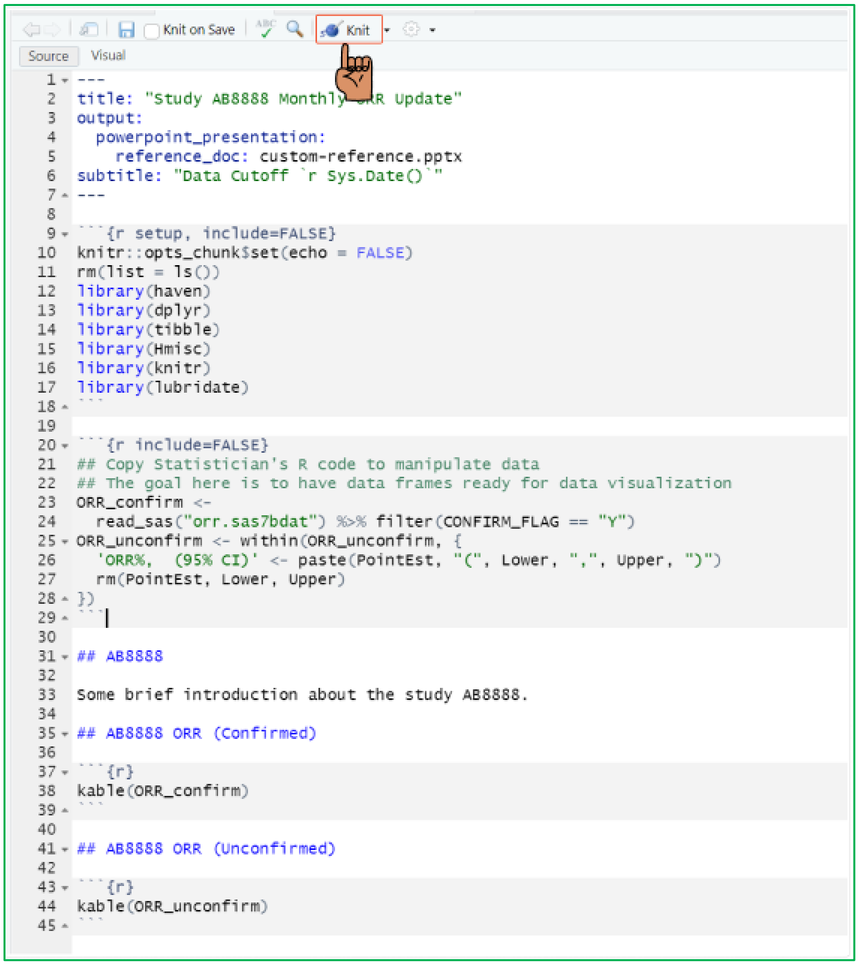
AB8888 ORR (Unconfirmed)

```{r}
kable(ORR_unconfirm)
```
```

Kable is a function for generating tables. It works perfectly with R Markdown, adapting the style of tables to the template we specified in the header. You can use any functions/packages you need to visualize data.

FINAL INTEGRATION

By integrating the header, data manipulation, and report writing components, you can knit and generate your report in the form of presentation slides.


Knit on Save
Knit
Source
Visual

```
---
title: "Study AB8888 Monthly ORR Update"
output:
  powerpoint_presentation:
    reference_doc: custom-reference.pptx
subtitle: "Data Cutoff `r Sys.Date()`"
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE)
rm(list = ls())
library(haven)
library(dplyr)
library(tibble)
library(Hmisc)
library(knitr)
library(lubridate)
```

```{r include=FALSE}
## Copy Statistician's R code to manipulate data
## The goal here is to have data frames ready for data visualization
ORR_confirm <-
  read_sas("orr.sas7bdat") %>% filter(CONFIRM_FLAG == "Y")
ORR_unconfirm <- within(ORR_unconfirm, {
  'ORR%,  (95% CI)' <- paste(PointEst, "(", Lower, ",", Upper, ")")
  rm(PointEst, Lower, Upper)
})
```

AB8888

Some brief introduction about the study AB8888.

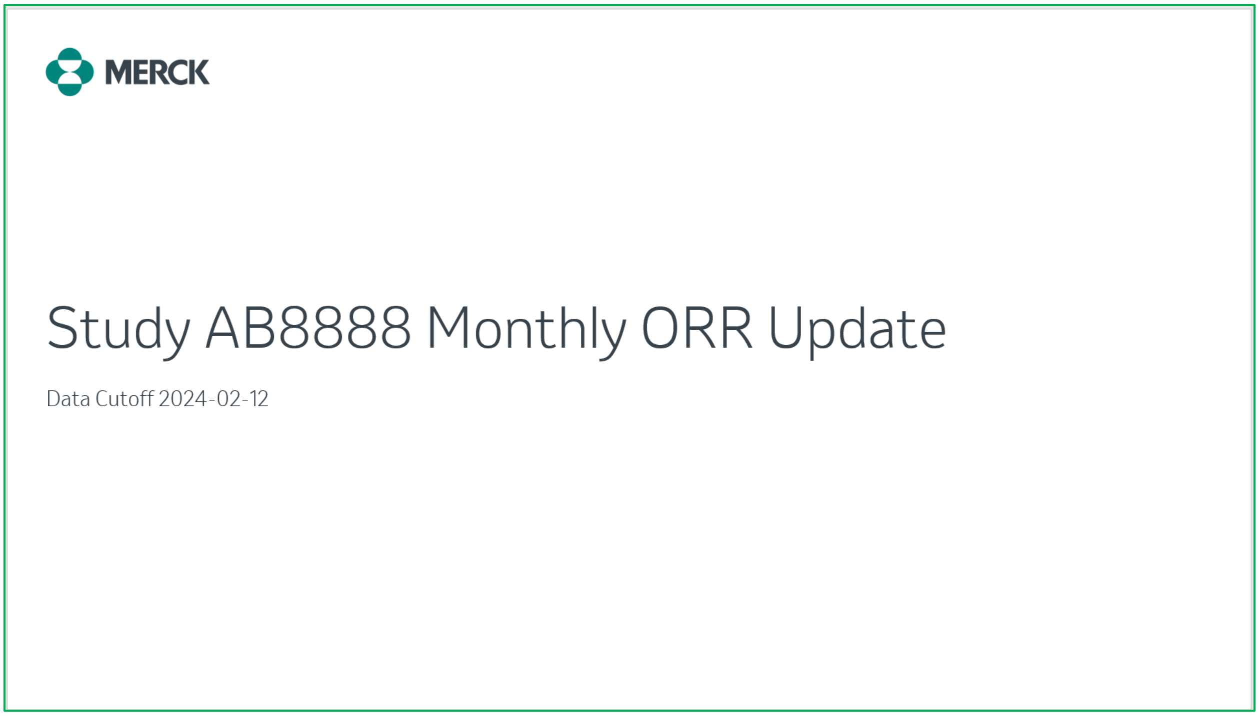
AB8888 ORR (Confirmed)

```{r}
kable(ORR_confirm)
```

AB8888 ORR (Unconfirmed)

```{r}
kable(ORR_unconfirm)
```
```

Display 4. Knit button in RStudio
```



MERCK

Study AB8888 Monthly ORR Update

Data Cutoff 2024-02-12

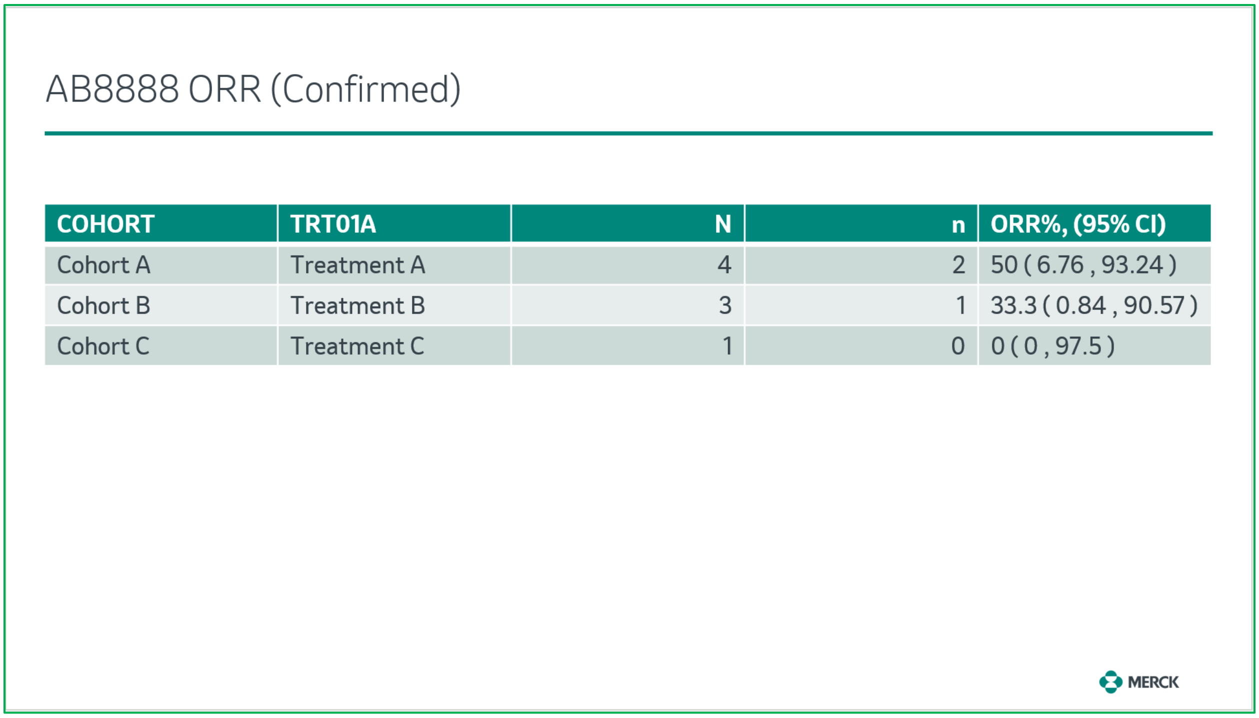
**Output 1.1 Output from example R Markdown file**

AB8888

Some brief introduction about the study AB8888.

MERCK

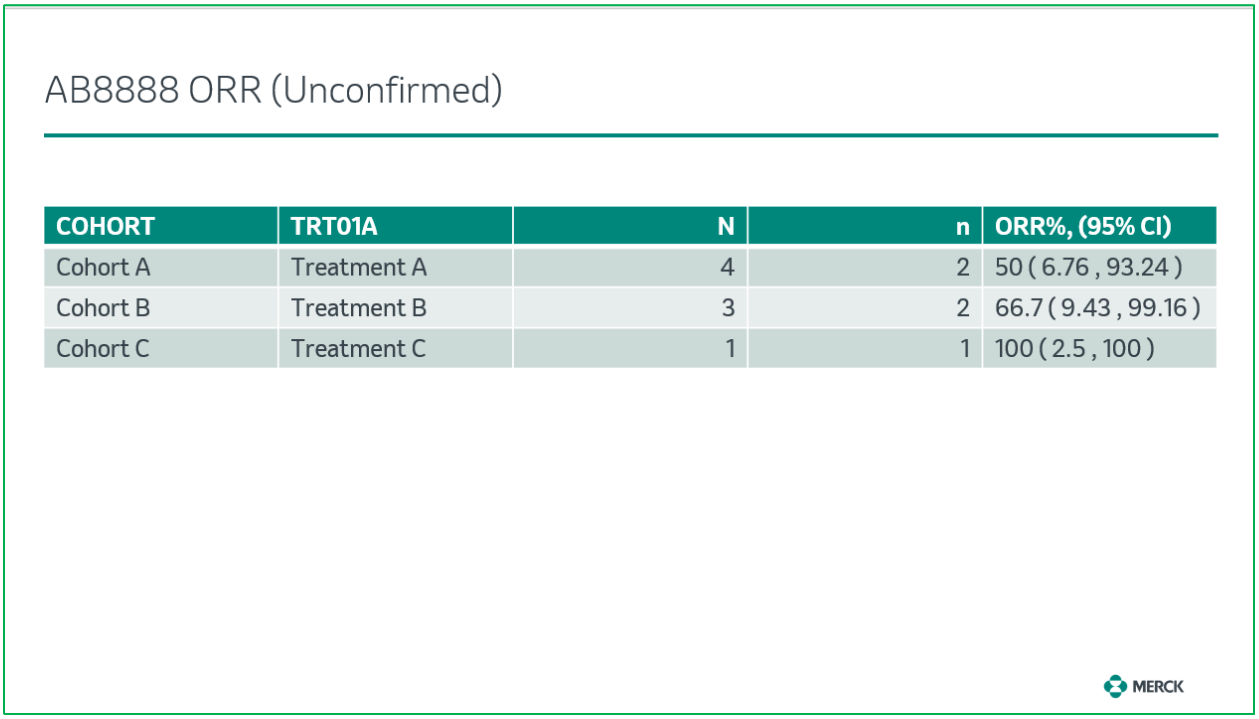
**Output 2.2 Output from example R Markdown file**



## AB8888 ORR (Confirmed)

| COHORT | TRT01A | N | n | ORR%, (95% CI) |
|---|---|---:|---:|---|
| Cohort A | Treatment A | 4 | 2 | 50 ( 6.76 , 93.24 ) |
| Cohort B | Treatment B | 3 | 1 | 33.3 ( 0.84 , 90.57 ) |
| Cohort C | Treatment C | 1 | 0 | 0 ( 0 , 97.5 ) |

**Output 3.3 Output from example R Markdown file**



## AB8888 ORR (Unconfirmed)

| COHORT | TRT01A | N | n | ORR%, (95% CI) |
|---|---|---:|---:|---|
| Cohort A | Treatment A | 4 | 2 | 50 ( 6.76 , 93.24 ) |
| Cohort B | Treatment B | 3 | 2 | 66.7 ( 9.43 , 99.16 ) |
| Cohort C | Treatment C | 1 | 1 | 100 ( 2.5 , 100 ) |

**Output 1.4 Output from example R Markdown file**

## CONCLUSION

In summary, this paper introduces the HDR method for generating periodic reports using R Markdown, with a specific focus on PPTX presentation slides.  This approach distinctly separates data manipulation from output presentation, reducing code complexity and enhancing reusability.

## REFERENCES

Markdown https://rmarkdown.rstudio.com/ Accessed January 7, 2024.

## CONTACT INFORMATION

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

Junze Zhang
junze.zhang@merck.com

Chuanhai Tang
chuanhai.tang@merck.com