

Making Customized RTF Output with R Package r2rtf

Huei-Ling Chen, Heng Zhou, Madhusudhan Ginnaram, Yilong Zhang
Merck & Co., Inc., Kenilworth, NJ, USA

ABSTRACT

The Rich Text Format (RTF) file is a typical output file format for reporting, conventionally created with SAS procedures. However, with R's increasing popularity in the pharmaceutical industry, there is also a growing need to generate RTF files with R. Among many existing R packages to create RTF files, a newly developed open-source package, `r2rtf`, stands out, with easy-to-use features and powerful functions. It allows R users to produce various RTF outputs effortlessly. This paper uses three examples to demonstrate different methods of creating RTF tables using this package. We have papers presented in PharmaSUG 2019 (<https://www.lexjansen.com/pharmasug/2019/ST/PharmaSUG-2019-ST-149.pdf>) and 2020 (<https://www.pharmasug.org/proceedings/2020/DV/PharmaSUG-2020-DV-135.pdf>) that showed two of these same outputs produced using SAS macros.

KEYWORDS

R, r2rtf, mTPI, DLT, PAVA, ICH Listing, Rich Text Format

INTRODUCTION

The `r2rtf` is a new R package that can easily create presentation-ready RTF tables, figures and listings. The package resides in the open source community. A paper in the PharmaSUG 2020 conference [Wang et.al., 2020] introduced version 0.1.1 of this package. The package provides flexibility to customize table appearance for table title, subtitle, footnote, column header, table size, border format, text format, column width, row height, etc.

Since the publishing of the paper, `r2rtf` had released version 0.2.0 on CRAN to provide the enhanced features for pagination, section grouping, and multiple table concatenations. The table below summarizes the key changes from v0.1.1.

r2rtf0.2.0
<ul style="list-style-type: none">• Add <code>rtf_subline</code>, <code>rtf_page_header</code>, <code>rtf_page_footer</code>.• Add <code>rtf_page</code> to set page related attributes.• Introduce <code>text_convert</code> argument to allow fixed string.• Add <code>as_table</code> argument in <code>rtf_footnote</code> and <code>rtf_source</code> to allow footnote and datasource inside or outside of a table.• Refactor <code>pageby</code> feature to enable <code>group_by</code> feature. Add <code>vignette\example-pageby.Rmd</code> to illustrate new <code>pageby</code> features.• Define <code>obj_rtf_border</code> and <code>obj_rtf_text</code> objects to standardize border and text attributes.• Add example ADaM datasets.

Display 1: Summary of changes in r2rtf v0.2.0 compared with v0.1.1

We can install the `r2rtf` package from CRAN using the R syntax below.

```
install.packages("r2rtf")
```

The development version of `r2rtf` package is available at: <https://github.com/Merck/r2rtf>.

This paper presents three examples to demonstrate the features found in `r2rtf` v0.2.0 for creating customized RTF tables. Each example includes sample dataset, table content, table output, and R syntax.

Throughout the paper, we also use the `haven` package to read the SAS dataset and the `dplyr` R package to manipulate the data.

The first section displays a simple table which uses basic feature of `r2rtf`. The second example is a more advanced table which adds a feature grouping the data by columns. The final example is the most challenging one, which involves more features.

Note that all examples focus solely on the part from final data to RTF table.

TABLE 1: MTPITABLE

The first example is the modified Toxicity Probability Interval (mTPI) table. The mTPI design is used to identify the maximum tolerated dose (MTD) of the investigational drug in phase I dose-finding studies. The entries of the mTPI table are the dose-finding decisions: E, S, D, and DU represent escalating the dose, staying at the same dose, de-escalating the dose, and excluding the dose from the trial due to unacceptable toxicity, respectively.

Sample Dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	0	E	E	E	E	E	E	E	E	E	E	E	E
2	1	S	S	S	E	E	E	E	E	E	E	E	E
3	2	D	S	S	S	S	S	S	S	E	E	E	E
4	3	DU	DU	D	S	S	S	S	S	S	S	S	S
5	4		DU	DU	DU	D	D	S	S	S	S	S	S
6	5			DU	DU	DU	DU	DU	D	S	S	S	S
7	6				DU	DU	DU	DU	DU	DU	D	S	S
8	7					DU	D						
9	8						DU						
10	9							DU	DU	DU	DU	DU	DU
11	10								DU	DU	DU	DU	DU
12	11									DU	DU	DU	DU
13	12										DU	DU	DU
14	13											DU	DU
15	14												DU

Display 2: mockup final dataset "final"

Table Contents

The columns of the table indicate the number of participants treated at the current dose level, and the rows indicate the number of participants experiencing dose-limiting toxicity (DLT).

Table Output

Dose-finding Rules per mTPI Design												
Number of Participants with at least 1 DLT	Number of Participants Evaluable for DLT at Current Dose											
	3	4	5	6	7	8	9	10	11	12	13	14
0	E	E	E	E	E	E	E	E	E	E	E	E
1	S	S	S	E	E	E	E	E	E	E	E	E
2	D	S	S	S	S	S	S	S	E	E	E	E
3	DU	DU	D	S	S	S	S	S	S	S	S	S
4		DU	DU	DU	D	D	S	S	S	S	S	S
5			DU	DU	DU	DU	DU	D	S	S	S	S
6				DU	DU	DU	DU	DU	DU	D	S	S
7					DU	D						
8						DU						
9							DU	DU	DU	DU	DU	DU
10								DU	DU	DU	DU	DU
11									DU	DU	DU	DU
12										DU	DU	DU
13											DU	DU
14												DU

Abbreviations: D = De-escalate to the next lower dose; DU= The current dose is unacceptably toxic; E = Escalate to the next higher dose; S = Stay at the current dose.
 Target toxicity rate= 30%
 Flat noninformative prior Beta (1,1) is used as a prior and $\epsilon_1=\epsilon_2=0.03$ [Ji Y, Li Y, Bekele BN 2007] [Ji, Y. and Wang, S.-J. 2013] [Ji, Y., et al 2010]

Table 1: mTPI Mock Table

R Syntax

By comparing with the mock table and final data, we can customize the items below:

- **Page orientation:** The `rtf_page` function is used to set all features related to page. The `orientation` argument is used to define the page orientation. i.e. `rtf_page(orientation = "landscape")`
- **Column header:** There are two column headers each with a different number of columns. In `r2rtf`, we can provide multiple column headers using the `rtf_colheader` function.
- **Column width:** The first column is wider compared with the rest. The `col_rel_width` argument in `rtf_colheader` and `rtf_body` function defines the relative width of each column. Only the ratio of each column will be used; in other words, `col_rel_width = c(1, 2, 3)` has the same effect of `col_rel_width = c(2, 4, 6)`.
- **Special character:** The Greek special character can be created by using latex code. For example, ϵ_1 is created by `\\epsilon{1}`.

```
table_rtf <- final %>%
  rtf_page(orientation = "landscape") %>%
  rtf_title(title = "Dose-finding Rules per mTPI Design" ) %>%
  rtf_colheader(colheader = " |Number of Participants Evaluable for DLT at Current Dose",
               col_rel_width = c(6,12)) %>%
  rtf_colheader(colheader = "Number of Participants with at least 1 DLT
                           |3|4|5|6|7|8|9|10|11|12|13|14",
               col_rel_width = c(6,rep(1,12))) %>%
  rtf_body(col_rel_width = c(6,rep(1, 12)),
          border_bottom = "single",
          text_justification = c(rep("c",13))) %>%
  rtf_footnote(footnote = c("Abbreviations: D = De-escalate to the next lower dose; DU= The
                             current dose is unacceptably toxic; E = Escalate to the next higher dose;
                             S = Stay at the current dose.",
                             "Target toxicity rate= 30%",
                             "Flat noninformative prior Beta (1,1) is used as a prior and
                             \\epsilon{1}=\\epsilon{2}=0.03 [Ji Y, Li Y, Bekele BN 2007] [Ji, Y. and
                             Wang, S.-J. 2013] [Ji, Y., et al 2010] "))

table_rtf %>% rtf_encode() %>% write_rtf("decision0mtpi0table.rtf")
```

TABLE 2: DOSE LIMITING TOXICITY SUMMARY TABLE

The second example is the dose-limiting toxicity (DLT) summary statistics table using the pool-adjacent-violators algorithm (PAVA) analysis method to estimate DLT rates in the table. There is no SAS procedure or macros available for PAVA analysis, but the R package `ISO` can do the analysis. One reporting solution is to embed R syntax inside the SAS code to create RTF files. A PharmaSUG 2019 conference paper [Chen and Zhen, 2019] provides details on features used to streamline the integration process between SAS and R. Here, we simplify the process using only the R code to calculate the PAVA DLT rate and create the final RTF table.

Sample Dataset

treatment	N	DLTterm	n	BCI	PAVA	GRADE1	GRADE2	GRADE3	GRADE4	GRADE5
Dose Level 1	3	TOTAL	2	66.7 (36.5, 89.4)	33.3					
Dose Level 1		Anaemia	1	33.3		0	0	0	0	1
Dose Level 1		Fatigue	1	33.3		1	0	0	0	0
Dose Level 1		Insomnia	1	33.3		0	0	0	0	0
Dose Level 2	3	TOTAL	1	33.3 (10.6, 63.5)	33.3					
Dose Level 2		Cough	1	33.3		0	0	0	0	1
Dose Level 3	3	TOTAL	0	0.0 (0.0, 33.1)	33.3					
Dose Level 4	14	TOTAL	5	35.7 (21.5, 52.0)	35.7					
Dose Level 4		Anaemia	2	14.3		0	1	1	1	0
Dose Level 4		Cough	1	7.1		1	0	0	0	0
Dose Level 4		Decreased appetite	1	7.1		0	1	1	0	0
Dose Level 4		Fatigue	1	7.1		1	0	0	0	0
Dose Level 4		Vomiting	2	14.3		1	0	0	0	1

Display 3: mockup final dataset “final”

Table Contents

The meaning of each column is listed below:

- Treatment arms (**treatment**)
- The number of patients evaluable for DLT (**N**)
- Adverse event preferred term (**DLTterm**)
- The number of patients experiencing DLTs at each dose level by adverse event preferred term (**n**)
- Empirical DLT rate and Bayesian credible interval of DLT rate at each dose level (**BCI**)
- DLT rates estimated by PAVA for single arm dose-finding or two-dimensional PAVA for dose-finding with both monotherapy and combination therapy arms across all dose levels (**PAVA**)
- The number of patients experiencing DLTs at each dose level by adverse event preferred term and maximum toxicity grade (**GRADE1, GRADE2, GRADE3, GRADE4, GRADE5**)

Table Output

Summary of Dose Limiting Toxicity (DLT Evaluable Population) Mock Data PAVA - Mono only										
Treatment Group	N	Dose Limiting Toxicity	Participant with DLT			DLT by Maximum Grade ^a				
			n	% (80% CI) ^b	PAVA ^c Estimate (%)	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5
Dose Level 1	3	TOTAL	2	66.7 (36.5, 89.4)	33.3					
		Anaemia	1	33.3		0	0	0	0	1
		Fatigue	1	33.3		1	0	0	0	0
		Insomnia	1	33.3		0	0	0	0	0
Dose Level 2	3	TOTAL	1	33.3 (10.6, 63.5)	33.3					
		Cough	1	33.3		0	0	0	0	1
Dose Level 3	3	TOTAL	0	0.0 (0.0, 33.1)	33.3					
Dose Level 4	14	TOTAL	5	35.7 (21.5, 52.0)	35.7					
		Anaemia	2	14.3		0	1	1	1	0
		Cough	1	7.1		1	0	0	0	0
		Decreased appetite	1	7.1		0	1	1	0	0
		Fatigue	1	7.1		1	0	0	0	0
		Vomiting	2	14.3		1	0	0	0	1

DLT=Dose Limiting Toxicity
Adverse event terms are based on MedDRA version X.X.
Grades are based on NCI CTCAE version X.X.
Dose limiting toxicity (DLT) evaluable population consists of subjects XXXX.
^aOnly the highest reported grade for a given DLT is counted for the individual subject.
^b80% confidence interval is based on the Bayesian posterior credible interval with a prior distribution of Beta (1, 1).
^cPAVA: Pooled-adjacent-violator algorithm. For trials with both mono-therapy and combination-therapy arms, this is two-dimensional PAVA (2-D PAVA).

Source: [Study MK9999P001: adam-adsl,adae]

Table 2: Dose Limiting Toxicity Summary Mock Table

R Syntax

By comparing with the mock table and final data, we can customize additional features below:

- Group by variables: The values of these variables will only be printed once when they first appear. The `group_by` argument in the `rtf_body` function groups the variables defined in its value list.
- Cell justification: The first and third columns need to be left-aligned. The other columns are center-aligned. The `text_justification` argument in the `rtf_body` function controls the alignment of text in a column. Specifically, "l" is for left-align and "c" is for center-align.
- Superscript: The `{^a}` and `{^b}` is to add a superscript for "a" and "b". The curly braces define the pattern for superscript "^" only affect text within it.

```
table_rtf <- final %>%
  rtf_page(orientation = "landscape") %>%
  rtf_title(title = "Summary of Dose Limiting Toxicity",
           subtitle = c(" (DLT Evaluable Population)",
                       "Mock Data",
                       "PAVA - Mono only" ) ) %>%
  rtf_colheader(colheader = " | | |
                          Participant with DLT | DLT by Maximum Grade{^a} ",
                col_rel_width = c(3,1,3.5,6, 7.5)) %>%
  rtf_colheader(colheader = "Treatment Group| N | Dose Limiting Toxicity |
                          n | % (80% CI){^b} | PAVA{^c} Estimate (%) |
                          Grade 1|Grade 2|Grade 3|Grade 4|Grade5",
                col_rel_width = c(3,1,3.5,1,3,2,rep(1.5, 5))) %>%
  rtf_body(col_rel_width = c(3,1,3.5,1,3,2,rep(1.5, 5)),
          text_justification = c("l", "c", "l", rep("c",8)),
          group_by="treatment") %>%
  rtf_footnote(footnote = c("DLT=Dose Limiting Toxicity",
                            "Adverse event terms are based on MedDRA version X.X.",
                            "Grades are based on NCI CTCAE version X.X.",
                            "Dose limiting toxicity (DLT) evaluable population . . .",
                            "{^a}Only the highest reported grade for a given DLT is counted . . .",
                            "{^b}80% confidence interval is based on the Bayesian posterior . . .",
                            "{^c}PAVA: Pooled-adjacent-violator algorithm. For trials with . . .")) %>%
  rtf_source(source = "Source: [Study MK9999P001: adam-adsl,adae]",
            text_justification = "l",
            as_table=FALSE)

table_rtf %>% rtf_encode() %>% write_rtf("dlt0pava0table.rtf")
```

TABLE 3: ICH LAB LISTING FILES

The third example is an International Consortium on Harmonization (ICH) abnormal laboratory data listing table. ICH data listings are standard reports prepared by pharmaceutical companies in EU submission, following a guiding document, ICH Guideline E3, on the reports' content. A paper from PharmaSUG 2020 conference [Chen and Wei, 2020] provides details of ICH listing report standards. It demonstrated using ODS RTF with Proc Report procedure in SAS to generate the ICH abnormal laboratory data listing RTF file.

We will illustrate using the `r2rtf` R package to produce the same table.

Sample Dataset

Display 4 is a mockup final dataset. Due to the page size limit, we break columns into two screenshots.

B	C	D	E	F	G	H	I	J	K	L
SUBJID	LBCAT	LBTEST	EPOCH	VISIT	LBDC_CHAR	LBDC_CHAR	RESULT_WITH_UNIT	LBORNLO	LBORNHI	ABN_COL
1015	Hematology	RBC	Screening	SCREENING 1	26-Dec-13	-7	5.30 MILL/uL	3.9	5.5	
1015	Hematology	RBC	Treatment	WEEK 2	16-Jan-14	15	5.30 MILL/uL	3.9	5.5	
1015	Hematology	RBC	Treatment	WEEK 4	30-Jan-14	29	5.00 MILL/uL	3.9	5.5	
1015	Hematology	RBC	Treatment	WEEK 6	12-Feb-14	42	5.10 MILL/uL	3.9	5.5	
1015	Hematology	RBC	Treatment	WEEK 8	5-Mar-14	63	5.10 MILL/uL	3.9	5.5	
1015	Hematology	RBC	Treatment	WEEK 12	26-Mar-14	84	5.00 MILL/uL	3.9	5.5	

B	M	N
SUBJID	DECODE_1	VAR_1CELL_DOSE
1015	Placebo	Trial Number=CDISCIPILOT01, Site Number=701, Treatment Group=Placebo, Unique Subject ID=01-701-1015, Sex=F, Age=63 Years, Weight=120.0 LB
1015	Placebo	Trial Number=CDISCIPILOT01, Site Number=701, Treatment Group=Placebo, Unique Subject ID=01-701-1015, Sex=F, Age=63 Years, Weight=120.0 LB
1015	Placebo	Trial Number=CDISCIPILOT01, Site Number=701, Treatment Group=Placebo, Unique Subject ID=01-701-1015, Sex=F, Age=63 Years, Weight=120.0 LB
1015	Placebo	Trial Number=CDISCIPILOT01, Site Number=701, Treatment Group=Placebo, Unique Subject ID=01-701-1015, Sex=F, Age=63 Years, Weight=120.0 LB
1015	Placebo	Trial Number=CDISCIPILOT01, Site Number=701, Treatment Group=Placebo, Unique Subject ID=01-701-1015, Sex=F, Age=63 Years, Weight=120.0 LB

Display 4: mockup final dataset

Table Contents

The meaning of each column is listed below:

- Data is grouped by site and investigator.
 - Site title row: study number (project and protocol number), site number, investigator name, country
- The listing has treatment arm and subject information in the header of every page.
 - Header row 1 - treatment arm description
 - Header row 2 - participant description: study number (project and protocol number), site number, treatment, subject identifier, sex, age, weight.
- Display the lab data in the order of importance of the tests and laboratory testing date.
 - Columns: category for lab test, lab test name, epoch, visit, date/time of specimen collection, relative day of specimen collection, results in original units, normal range values, and abnormal value indicator
- Page numbers are in the header on every page.
- "CONFIDENTIAL" is displayed on the footer of every page.
- Split the listing into multiple files smaller than 85 megabytes. For each file, the title and name include a letter (a, b, c...) to indicate the order of the files overall.

Table Output

16.2.8a Listing of Laboratory Values

Trial Number: CDISCPIL01, Site: 701, Investigator: Doe, John, Country: USA

Subject ID	Category	Test	Epoch	Visit	Date	Relative Day*	Result	Normal Range		Abnormal ^b
Placebo										
Trial Number=CDISCPIL01, Site Number=701, Treatment Group=Placebo, Unique Subject ID=01-701-1015, Sex=F, Age=63 Years, Weight=120.0 LB										
1015	Hematology	RBC	Screening	SCREENING 1	26DEC13	-7	5.30 MILL/uL	3.9	5.5	
			Treatment	WEEK 2	16JAN14	15	5.30 MILL/uL	3.9	5.5	
				WEEK 4	30JAN14	29	5.00 MILL/uL	3.9	5.5	
				WEEK 6	12FEB14	42	5.10 MILL/uL	3.9	5.5	
				WEEK 8	05MAR14	63	5.10 MILL/uL	3.9	5.5	
				WEEK 12	26MAR14	84	5.00 MILL/uL	3.9	5.5	

CONFIDENTIAL

4. Page numbers are displayed in the header on every page. Page 1 of 513

1. Data is grouped by site and investigator.

2. The listing has treatment arm and subject information in the header of every page.

3. Display the lab data in the order of importance of the tests and laboratory testing.

5. "CONFIDENTIAL" is displayed on the footer of every page.

6. For each file, the title and name include a letter (a, b, c...) to indicate the order of the files overall.

Table 3: ICH Lab Listing Mock Table

R Syntax

The `r2rtf` package prints the data points to table cells by their respective order in the dataset. Hence, there is no argument for a user to enter the variable name in the function. It is crucial to create an input dataset with the correct variable order. Table columns come before table header rows in the dataset.

- Columns
- Header rows

The `dplyr::select()` function in R is for select variables in order. In Table 3, the first column is `SUBJID` for subject identifier, the second column is `LBCAT` for lab category, and hence the variable `select` first would be `SUBJID`, and next is `LBCAT`, etc. The `dplyr::arrange()` function in R sort the data. The data is sorted by header rows first and group by columns next.

One essential step before using the `r2rtf` package is to ensure the number of entries in the `rtf_body` matches the number of variables in the dataset. Table 3 has 11 columns and 2 header rows; hence, the final dataset contains 13 variables.

```
final <- datain %>%
select(SUBJID,LBCAT,LBTEST,EPOCH,VISIT,LBDC_CHAR,LBDY_CHAR,RESULT_WITH_UNIT,
LBORNRO,LBORNRI,ABN_COL,DECODE_1,VAR_1CELL_DOSE)
arrange(DECODE_1, VAR_1CELL_DOSE, SUBJID, LBCAT, LBTEST, EPOCH)
```

- Text format: The `text_format` argument in the `rtf_body` function defines the format of the text. The valid entry of the `text_format` argument is " " for normal text, "b" for bold, "i" for italics, "u" for underline, "s" for strike, "^" for superscript, and "_" for subscript text.
- Border lines: You can control borderlines using the `border_top`, `border_bottom`, `border_left`, and `border_right` arguments in the `rtf_body` function. Specifically, "single" is single line, "double" is double line, and blank " " is no borderline.

- The arguments (`col_rel_width`, `text_justification`, `text_format`, `border_top`, ..., etc) in the function `rtf_body` all contain 13 entries in the value list to define the features of each column.

```
db <- final %>%
  rtf_page(orientation = "landscape",
          col_width = 10,
          nrow = nrow) %>%

  rtf_page_header() %>%
  rtf_page_footer(text = "CONFIDENTIAL") %>%

  rtf_title(title="16.2.8a Listing of Laboratory Values") %>%

  rtf_subline(text= subtitle) %>%

  rtf_colheader(colheader = "Subject| | | | |Relative| |Normal Range| ",
                col_rel_width = c(2,4,5,3,3,3,2,4,3,3,3,1,1),
                text_font_size=8 ) %>%

  rtf_colheader(colheader = "ID|Category|Test|Epoch|Visit|Date|Day{^a}|
  Result|Low|High|Abnormal{^b}",
                col_rel_width = c(2,4,5,3,3,3,2,4,3,3,3),
                border_top = c(rep("", 11)),
                border_left = c(rep(c("single"), 9), "", "single"),
                text_font_size=8) %>%

  rtf_body(col_rel_width = c(2,4,5,3,3,3,2,4,3,3,3,1,1),
           text_justification = c("l", rep("c",10), "l", "l"),
           text_format = c(rep("",11), "b", ""),
           border_top = c(rep("", 11), "single","single"),
           border_bottom = c(rep("", 11), rep("single", 2)),
           border_left = c(rep(c("single"), 9), "", rep(c("single"), 3)),
           page_by= c("DECODE_1", "VAR_1CELL_DOSE"),
           group_by= c("SUBJID", "LBCAT", "LBTEST", "EPOCH"),
           text_font_size=8)

db %>% rtf_encode()
```

1. Data is grouped by site and investigator.

The `rtf_subline()` adds subtitle attributes to the table. The subtitle appears on top of every page and changes for a different site. We slice the data to subsets by site, and each site has a variable `SUBJLINE` which contains trial information, site number, investigator name, and country. Each site has its rtf object, the subtitle then can be updated by site.

```
subtitle <- unique(db$SUBJLINE)
rtf_subline(subtitle)
```

2. The listing has treatment arm and subject information in the header of every page.

The two header rows, treatment arm description (`DECODE_1`) and participant description (`VAR_1CELL_DOSE`), are controlled by argument `page_by` in the function `rtf_body`. The two variables are displayed as nested rows instead of displaying the two variables as a column in the table. Those nested rows repeated at each page for the same group to ensure self-contained within a page.

Display 5 below is a screenshot to highlight multiple pages. The treatment arm and subject information are in the header of every page.

Page 1 of 513

16.2.8a Listing of Laboratory Values

Trial Number: CDISCPLOT01, Site: 701, Investigator: Doe, John, Country: USA

Subject ID	Category	Test	Epoch	Visit	Date	Relative Day ^a	Result	Normal Range		Abnormal ^b
								Low	High	
Placebo										
Trial Number=CDISCPLOT01, Site Number=701, Treatment Group=Placebo, Unique Subject ID=01-701-1015, Sex=F, Age=63 Years, Weight=120.0 LB										
1015	Hematology	RBC	Screening	SCREENING 1	26DEC13	-7	5.30 MILL <u>u</u> L	3.9	5.5	
			Treatment	WEEK 2	16JAN14	15	5.30 MILL <u>u</u> L	3.9	5.5	

Page 2 of 513

16.2.8a Listing of Laboratory Values

Trial Number: CDISCPLOT01, Site: 701, Investigator: Doe, John, Country: USA

Subject ID	Category	Test	Epoch	Visit	Date	Relative Day ^a	Result	Normal Range		Abnormal ^b
								Low	High	
Placebo										
Trial Number=CDISCPLOT01, Site Number=701, Treatment Group=Placebo, Unique Subject ID=01-701-1015, Sex=F, Age=63 Years, Weight=120.0 LB										
1015	Hematology	RBC	Treatment	WEEK 16	07MAY14	126	5.20 MILL <u>u</u> L	3.9	5.5	
				WEEK 20	21MAY14	140	4.90 MILL <u>u</u> L	3.9	5.5	
				WEEK 24	18JUN14	168	5.30 MILL <u>u</u> L	3.9	5.5	

Display 5: Multiple Pages screen shot

Note that there are two entries in the arguments (col_rel_width, text_justification, text_format, border_top, ..., etc) in the function specifically defined for these two variables.

```
rtf_body (... ,
  page_by= c ("DECODE_1", "VAR_1CELL_DOSE")
  ... )
```

3. Display the lab data in the order of importance of the tests and laboratory testing date.

The factor() function in R with levels argument can determine a specific order of a categorical variable values.

```
datain <- datain %>%
  mutate (LBCAT = factor (LBCAT,
    levels=c ("Hematology", "Chemistry", "Urinalysis", "Other", "")))
```

The rtf_body() function in r2rtf with group_by argument can make the data grouped by and displayed in order of subject id (SUBJID), lab test category (LBCAT), lab test (LBTEST), and epoch (EPOCH). The sorting follows the specific order determined by the factor function on variables (LBCAT, LBTEST, EPOCH).

```
rtf_body (... ,
  group_by= c ("SUBJID", "LBCAT", "LBTEST", "EPOCH"),
  ... )
```

4. Page numbers are displayed in the header on every page.

The rtf_page_header() adds RTF page header information.

```
rtf_page_header ()
```

5. "CONFIDENTIAL" is displayed on the footer of every page.

The `rtf_page_footer()` adds RTF page footer information.

```
rtf_page_footer(text = "CONFIDENTIAL")
```

6. Split the listing into multiple files smaller than 85 megabytes. For each file, the title and name include a letter (a, b, c...) to indicate the order of the files overall.

The detailed process to carry out this task is below:

Step 1: Create a temp ICH listing rtf object for one site in R.
Step 2: Calculate megabyte size per record from temp ICH listing R object.
Step 3: Assign group ID to data by SITE / SUBJID that most close to total rows.
Step 4: Split data by group ID
Step 5: One RTF object per group ID. Create multiple RTF files. For each file, the title and name include a letter (a, b, c...) to indicate the order of the files overall.

Step 1: Create a temp ICH listing rtf table for one site.

The `split()` function divides the input data in different groups within a list. A list is an R structure that may contain objects of any other types. Here, we use a list to specify datasets with different site information (variable `SUBJLINE`) to different objects.

```
lab_t1_list<- split(lab_t1,lab_t1$SUBJLINE)
```

Next, retrieve the first site data for a temp ICH listing rtf table.

```
db <- lab_t1_list[[1]]
```

Step 2: Calculate megabyte size per record from temp ICH listing file.

The `object.size()` function provides the memory size that is being used to store an R object. The `nrow()` function returns the number of records of the dataset. The calculation formula below returns the megabyte size per record from the temp ICH listing file.

```
file_size <- as.numeric(object.size(lab_t1_encode_temp[[2]]) / (nrow(db) * 1024^2))
```

Step 3: Assign group ID to data by SITE / SUBJID that most close to total rows.

Calculate the number of record thresholds.

```
record_limit <- 85 / file_size
```

Assign group ID (a temp variable `filegroup`) to data by `SITE` and `SUBJID` combination that is most close to the number of record thresholds (`record_limit`).

Step 4: Split data by group ID

```
lab_t2_list<- split(lab_t2,lab_t2$filegroup)
```

Step 5: Create one RTF object per group ID. Create multiple RTF files. For each file, the title and name include a letter (a, b, c...) to indicate the order of the files overall.

The `seq_along()` function creates a sequence from 1 to the length of the input. It is useful for loops.

Use `letters()` function to create a sequence of letters and stick the letter postfix to the title and file name. This is to indicate the order of the files overall.

```
for(i in seq_along(lab_t2_list)){
  db <- lab_t2_list[[i]]
  subline <- unique(db$SUBJLINE)
  abc <- letters[unique(db$filegroup)]
  title <- paste0("16.2.8", abc, ' Listing of Laboratory Values')
  . . .
  encode %>% write_rtf(paste0("combine_ichlisting-", letters[i], ".rtf"))
}
```

CONCLUSION

The `r2rtf` package version 0.2.0 has more powerful functions than version 0.1.1. It has some new features allowing R users to produce various RTF outputs effortlessly. This paper uses three examples to present the key R syntax, input datasets, and output tables. With `r2rtf` package available, the clinical trial reporting is much easier to produce using R. It helps straighten out a path for carrying out both data preparation and table generating in R.

REFERENCES

R packages

- `r2rtf`: <https://github.com/Merck/r2rtf>
- `haven`: <https://www.rdocumentation.org/packages/haven/versions/2.3.1>
- `dplyr`: <https://www.rdocumentation.org/packages/dplyr/versions/0.7.8>

Proceedings

- Wang, S., Ye, S., Anderson, K., Zhang, Y., 2020. "r2rtf – an R Package to Produce Rich Text Format (RTF) Tables and Figures" Proceedings of PharmaSUG 2020. (<https://www.pharmasug.org/proceedings/2020/DV/PharmaSUG-2020-DV-198.pdf>)
- Chen, H., Wei, W., 2020. "Making Customized ICH Listings with ODS RTF" Proceedings of PharmaSUG 2020. (<https://www.pharmasug.org/proceedings/2020/DV/PharmaSUG-2020-DV-135.pdf>)
- Chen, H., Zeng, Z., 2019. "Application of R@ Functions in SAS@ to Estimate Dose Limiting Toxicity Rates for Early Oncology Dose Finding" Proceedings of PharmaSUG 2019. (<https://www.lexiansen.com/pharmasug/2019/ST/PharmaSUG-2019-ST-149.pdf>)

ACKNOWLEDGMENTS

The authors would like to thank Xiaohui Wang from Merck & Co., Inc., Kenilworth, NJ, USA, for her advice on this paper/presentation.

CONTACT INFORMATION

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

Huei-Ling Chen, Ph.D.
Merck & Co., Inc., Kenilworth, NJ, USA
e-mail: huei-ling_chen@merck.com

Heng Zhou, Ph.D.
Merck & Co., Inc., Kenilworth, NJ, USA
e-mail: heng.zhou@merck.com

Madhusudhan Ginnaram
Merck & Co., Inc., Kenilworth, NJ, USA
e-mail: madhusudhan.ginnaram.reddy@merck.com

Yilong Zhang, Ph.D.
Merck & Co., Inc., Kenilworth, NJ, USA
e-mail: yilong.zhang@merck.com

TRADEMARK

SAS and all other SAS Institute Inc. products or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.