PharmaSUG 2024 - Paper AP- 229

Create a Shift Summary of Laboratory Values in CTCAE Grade to the Worst Grade Abnormal Value Output using R and SASSY System

Vicky Yuan, Incyte Corporation;
David J. Bosak, Archytas Clinical Solutions

ABSTRACT

A shift summary of laboratory values in CTCAE grade to the worst grade abnormal value is often required for most laboratory data analysis and submission. The purpose of CTCAE grade shift table is to present how the results vary from the baseline to post-base visits in the study. This paper will illustrate how to report a shift table using R and packages from the SASSY system. It will start from an example and explain the anatomy, then a step-wise explanation of how to report the table in a PDF file. The example is interesting because the columns and "internal" footnotes change on every page.

INTRODUCTION

The **sassy** system simplifies the experience for SAS programmers working in R. These packages bring R coding closer to the robust reporting capabilities inherent in SAS. By incorporating these features, the **sassy** system enhances efficiency and overall satisfaction when writing programs in R. This paper is using shift laboratory in CTCAE grade to present how the **sassy** packages work.

The R product used in this paper is **sassy** system version 1.2.3 running in an RStudio environment.

WHAT IS SASSY?

The **sassy** system is an integrated set of packages to make programmers more productive in R, particularly those with a background in SAS® software. It includes several separate R packages. They were designed to complement each other, but can also be installed and used individually.

The system contains the following packages:

- **logr**: To create a traceable log
- **fmtr**: To create formats and format catalogs
- **libr**: To create a libname, a data dictionary, and perform a data step
- **reporter**: To create regulatory-style statistical reports. It specializes in producing Tables, Listings and Figures for the pharmaceutical, biotechnology, and medical-device industries.
- **procs**: Simulates several popular SAS procedures in R
- **common**: A collection of utility functions

Together, the above packages constitute a coherent and well-designed system for managing and reporting on data in R. This paper will use several functions from this system to accomplish the desired shift table.

SHIFT TABLE CHALLENGES

The table in question is a shift summary table of lab values with CTCAE Grade. The actual shift values have been pre-calculated by the sponsor. However, the challenge for this table is the layout and actual report generation. This table presented several difficulties while trying to produce it in R. The challenges encountered include:

(a) **Page Header**: Multiple lines of page header, aligned left and right, with a page number in the top right.

(b) **Page By**: A page by label above the table on both lab value and direction.

(c) **Columns Change**: The CTCAE grade can be different from one lab test to the next, and therefore the columns on each page can change.

(d) **Internal Footnotes**: Footnotes inside the table can change on every page, depending on the lab test and direction.

(e) **Page Footer**: The page footer has items aligned left and right, and is positioned above the page footnotes. Also, there is a border above the footer.

The above challenges are identified on the image below:

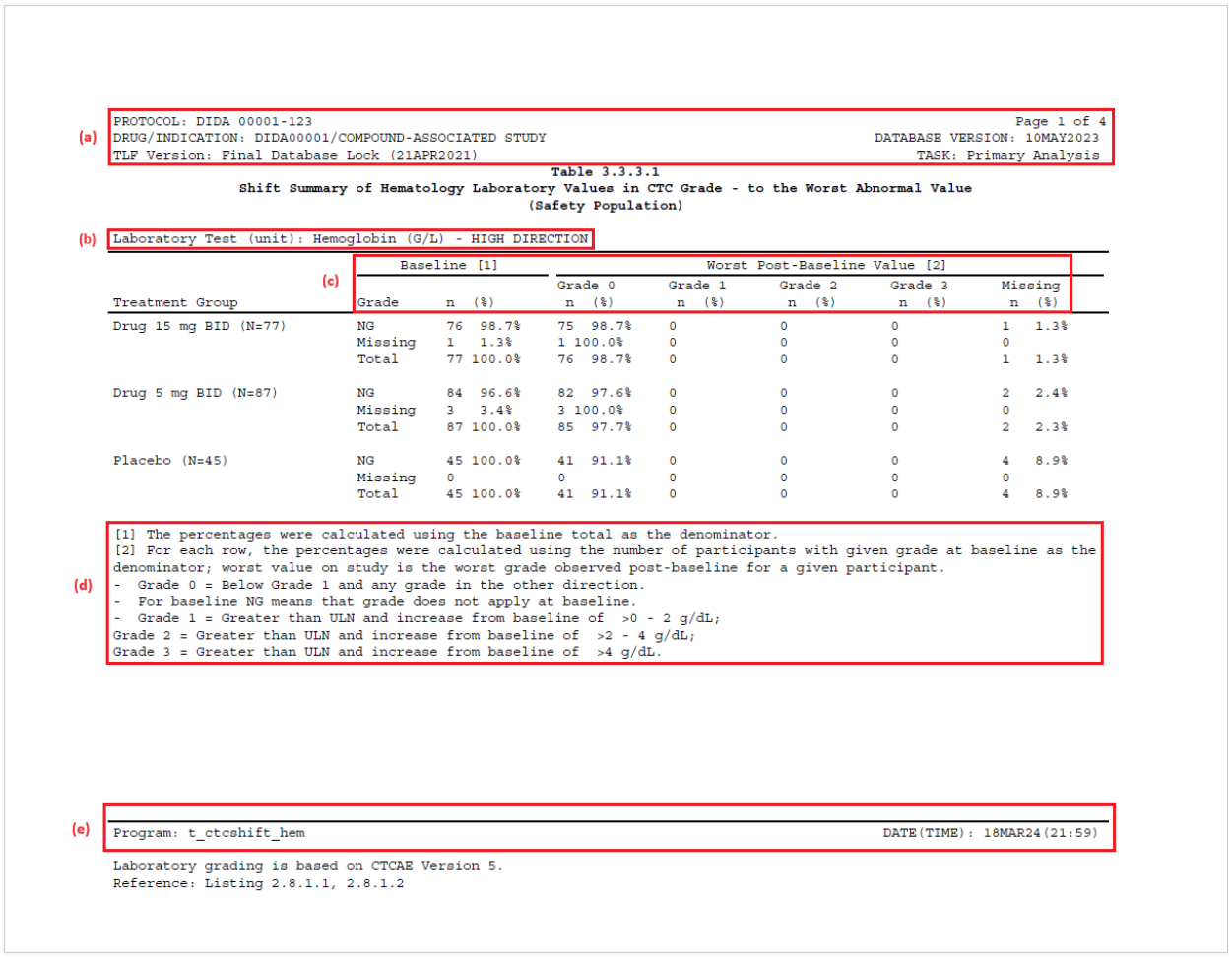

Figure 1. Shift table challenge areas

The **sassy** packages were able to accomplish this table using existing features, and dynamically generating each page of the output. The remainder of the paper will explain how it was accomplished.

CODE DEVELOPMENT

STEP 1: LOAD LIBRARIES AND SET UP VARIABLES

The first step in creating the desired shift table is to load the needed libraries and set some program variables. There are only two libraries needed for this table: **sassy** and **dplyr**. The **dplyr** package is used for data manipulation. The **sassy** package will load all of the other packages needed for reporting.

In addition, the program will set some variables needed in the program. The variables needed include the program name, output name, timestamp, and directory. These variables are set at the top of the program so they may be changed if needed. Here is the code:

```
library(sassy)
library(dplyr)

# Set variables
program.name <- "t_ctcshift_hem"
program.output <- "T_3_3_3_1_ctcshift_hem"
program.timestamp <- as.POSIXlt(Sys.time(), "UTC") %>%
  strftime("%d%b%y(%H:%M)") %>% toupper()
program.dir <- dirname(Sys.path())
```

STEP 2: OPEN LOG

The second step of the program is to open the log. The log is created using the **logr** package of the **sassy** system. This package can generate most of the log automatically. To engage the automatic log generation, you must set the "logr.autolog" option to TRUE, then open the log. The following code shows how to do this:

```
options("logr.autolog" = TRUE, "logr.notes" = FALSE)

# Open log
logpth <- log_open(file.path(program.dir, program.output))
```

Notice that the `program.dir` and `program.output` variables were set in Step 1 above.

STEP 3: PREPARE FORMATS

The third step is the program is to prepare some formats. These formats will be used to decode the lab value and the treatment groups. These formats are created using functions from the **fmtr** package:

```
fmt1 <- value(condition(x == "HEMO-HIGH",
                        "Hemoglobin (G/L) - HIGH DIRECTION"),
              condition(x == "HEMO-LOW",
                        "Hemoglobin (g/L) - LOW DIRECTION"),
              condition(x == "LEUK-HIGH",
                        "Leukocytes (GI/L) - HIGH DIRECTION"),
              condition(x == "LEUK-LOW",
                        "Leukocytes (GI/L) - LOW DIRECTION"))


fmt2 <- value(condition(x == "ARM A", "Drug 15 mg BID (N=77)"),
              condition(x == "ARM B", "Drug 5 mg BID (N=87)"),
              condition(x == "ARM C", "Placebo (N=45)"))
```

The above formats will be used later in the program. Note that the design of these functions is similar to how you would create formats in SAS using PROC FORMAT.

STEP 4: PREPARE LIST OF TABLE FOOTNOTES

The footnotes for this table are challenging because they can change on every page. Each lab value and direction have different footnotes. Further, those footnotes are included inside the table body, instead of at the bottom of the page. This footnote layout is difficult to achieve even in SAS.

To accomplish the desired footnotes in R, first we have to create a list. Then, we can populate the list with the needed footnotes for each lab value and direction. Here we are creating four sets of footnotes:

```
ftnts <- list()

base_ftnt <-
  c("[1] The percentages were calculated using the baseline total as the
denominator.",
    paste("[2] For each row, the percentages were calculated using the
number of participants",
    "with given grade at baseline as the denominator; worst value on study
is the worst",
    "grade observed post-baseline for a given participant."))

ftnts[["HEMO-HIGH"]] <-
  c(base_ftnt,
    "-  Grade 0 = Below Grade 1 and any grade in the other direction.",
    "-  For baseline NG means that grade does not apply at baseline.",
    "-  Grade 1 = Greater than ULN and increase from baseline of  >0 - 2
g/dL;",
    "Grade 2 = Greater than ULN and increase from baseline of  >2 - 4
g/dL;",
    "Grade 3 = Greater than ULN and increase from baseline of  >4 g/dL. ")


ftnts[["HEMO-LOW"]] <-
  c(base_ftnt,
    "-  Grade 0 = Below Grade 1 and any grade in the other direction.")


ftnts[["LEUK-HIGH"]] <-
  c(base_ftnt,
    "-  Grade 0 = Below Grade 1 and any grade in the other direction.",
    "-  For Leukocytes: Values between ULN and <= 100,000/mm3 are not
graded (NG). ")

ftnts[["LEUK-LOW"]] <-
  c(base_ftnt,
    "-  Grade 0 = Below Grade 1 and any grade in the other direction.")
```

STEP 5: PREPARE LIST OF COLUMNS

The CTCAE version 5 grade is different depending on the lab test performed. Since this report layout shows the grades in columns, we therefore need a way to dynamically create columns on each page. In step 5 we will create a list of column specifications for each lab test. The column specifications will be organized as a vector of column names and column labels:

```
cols <- list()

cols[["HEMO-HIGH"]] <- c(c1 = "Grade 0\n n  (%)", c2 = "Grade 1\n n  (%)",
                         c3 = "Grade 2\n n  (%)", c4 = "Grade 3\n n  (%)",
                         c7 = "Missing\n n  (%)")

cols[["HEMO-LOW"]] <- c(c1 = "Grade 0\n n  (%)", c2 = "Grade 1\n n  (%)",
                        c3 = "Grade 2\n n  (%)", c4 = "Grade 3\n n  (%)",
                        c7 = "Missing\n n  (%)")

cols[["LEUK-HIGH"]] <- c(c1 = "Grade 0\n n  (%)", c6 = "NG\n n  (%)",
                         c4 = "Grade 3\n n  (%)", c7 = "Missing\n n  (%)")
```

```
cols[["LEUK-LOW"]] <- c(c1 = "Grade 0\n n  (%)", c2 = "Grade 1\n n  (%)",
                        c3 = "Grade 2\n n  (%)", c4 = "Grade 3\n n  (%)",
                        c5 = "Grade 4\n n  (%)", c7 = "Missing\n n  (%)")
```

The above column specifications will be applied in step 7 below.

STEP 6: CREATE REPORT

Normally, when creating a report using the reporter package, you would create the content first. Then you would create the report and add content to the report. For the lab shift table, we are creating the report first, and then appending the content dynamically. Here is the code to create the report:

```
put("Create the report")
rpt <- create_report(pth, font = "Courier", font_size = 9)  %>%
  set_margins(top = 1.0, left = 1, right = 1, bottom = .5) %>%
  options_fixed(line_count = 51) %>%

  titles("Table 3.3.3.1",
         paste("Shift Summary of Hematology Laboratory Values",
         "in CTC Grade - to the Worst Abnormal Value"),
         "(Safety Population)", bold = TRUE, font_size = 9) %>%
  page_header(left = c("PROTOCOL: DIDA 00001-123",
              "DRUG/INDICATION: DIDA00001/COMPOUND-ASSOCIATED STUDY",
                       "TLF Version: Final Database Lock (21APR2021)"),
              right = c("Page [pg] of [tpg]",
                        "DATABASE VERSION: 10MAY2023",
                        "TASK: Primary Analysis")) %>%
  footnotes(paste0("Program: ", program.name, sep=""),
            "DATE(TIME): " %p% toupper(fapply(Sys.time(),
"%d%h%y(%H:%M)")),
            blank_row = "none", borders = "top", columns = 2,
            footer = TRUE)  %>%
  footnotes("Laboratory grading is based on CTCAE Version 5.",
            "Reference: Listing 2.8.1.1, 2.8.1.2", footer = TRUE)
```

Note that the above code also includes the functions necessary to produce the table header, page footer, and footnotes at the bottom of the page. These items are fixed for all pages. Therefore, they can be included here on the report object.

STEP 7: CREATE TABLES AND APPEND TO REPORT

The below code gets the unique lab tests from the data using `proc_sort()`. This function is similar to SAS PROC SORT. It is found in the **procs** package, which is also part of the **sassy** system.

Next, the code loops through the lab tests, creates a table for each test, and appends the appropriate footnotes for that test. Each table is then attached to the report object dynamically:

```
put("Get unique lab tests")
labtests <- proc_sort(hemo, by = labtest, options = nodupkey,
                      keep = labtest)

put("Loop through lab tests")
for (i in seq_len(nrow(labtests))) {

  put("Get lab test")
  lb <- labtests[[i, "labtest"]] %>% put()

  put("Get footnotes for this lab test")
```

```
    ftnt <- ftnts[[lb]]

    put("Filter data for this lab test")
    table_hemo <- hemo %>%
      dplyr::filter(labtest==lb) %>%
      mutate(labtest = fapply(labtest, fmt1),
             tmtnc = fapply(tmtnc, fmt2)) %>%
      select(labtest, tmtnc, tmtn, swmgrade,variable,
             c0,c1,c2,c3,c4,c5,c6,c7) %>%
      arrange(labtest, tmtn, swmgrade)

    put("Create table for this lab test")
    tbl <- create_table(table_hemo,
                        show_cols = "none",
                        borders = "top",
                        width = 9) %>%
      page_by(labtest, label = "Laboratory Test (unit):", borders = "none",
              blank_row = "none") %>%
      footnotes (ftnt, blank_row ="none" ) %>%
      spanning_header(variable, c0, label="Baseline [1]") %>%
      spanning_header(c1, c7, label="Worst Post-Baseline Value [2]") %>%
      define(labtest, visible = FALSE) %>%
      define(tmtnc, dedupe = TRUE, align = "left",
                    label = "Treatment Group") %>%
      define(tmtn, blank_after = TRUE, visible = FALSE) %>%
      define(variable, align="left", label="Grade", width=.8) %>%
      define(c0, align="left", label="n  (%)", width=1)

      # Get column specs
      cls <- cols[[lb]]

      # If more than 5 cols, reduce width
      wdth <- ifelse(length(cls) > 5, .85, 1)

      # Add column definitions dynamically for each test
      for (clnm in names(cls)) {
        tbl <- define(tbl, clnm, align = "left", label = cls[[clnm]],
               width = wdth, standard_eval = TRUE )
      }

    put("Add table to report dynamically")
    rpt <- rpt %>% add_content(tbl, blank_row = "none", page_break = TRUE)

  }
```

This method allows us to customize each page according to the requirements for the lab test. The **reporter** package offers a lot of flexibility. This technique can be used to create other challenging tables, such as patient profiles.

STEP 7: WRITE REPORT AND CLOSE LOG

The last step is to write out the report created in the previous two steps. This is the step that actually renders the report and writes it to a file. For this report, we are choosing a PDF output type. The reporter package can also generate RTF, DOCX, HTML, and TXT. Here is the code:

```
put("Write out the report")
res <- write_report(rpt, output_type = "PDF")


log_close()
```

At this point the program is complete, the output has been generated, and the log has been generated. We may view the output and log by running the following lines:

```
# View the report
file.show(res$modified_path)
file.show(logpth)
```

OUTPUT

Here are the PDF pages generated by the steps outlined above. Observe that the columns and internal footnotes are customized for each lab test:

PROTOCOL: DIDA 00001-123 Page 1 of 4
DRUG/INDICATION: DIDA00001/COMPOUND-ASSOCIATED STUDY DATABASE VERSION: 10MAY2023
TLF Version: Final Database Lock (21APR2021) TASK: Primary Analysis

Table 3.3.3.1
Shift Summary of Hematology Laboratory Values in CTC Grade - to the Worst Abnormal Value
(Safety Population)

Laboratory Test (unit): Hemoglobin (G/L) - HIGH DIRECTION

	Baseline [1]		Worst Post-Baseline Value [2]					
			Grade 0	Grade 1	Grade 2	Grade 3	Missing	
Treatment Group	Grade	n (%)	n (%)	n (%)	n (%)	n (%)	n	(%)
Drug 15 mg BID (N=77)	NG	76 98.7%	75 98.7%	0	0	0	1	1.3%
	Missing	1 1.3%	1 100.0%	0	0	0	0	
	Total	77 100.0%	76 98.7%	0	0	0	1	1.3%
Drug 5 mg BID (N=87)	NG	84 96.6%	82 97.6%	0	0	0	2	2.4%
	Missing	3 3.4%	3 100.0%	0	0	0	0	
	Total	87 100.0%	85 97.7%	0	0	0	2	2.3%
Placebo (N=45)	NG	45 100.0%	41 91.1%	0	0	0	4	8.9%
	Missing	0	0	0	0	0	0	
	Total	45 100.0%	41 91.1%	0	0	0	4	8.9%

[1] The percentages were calculated using the baseline total as the denominator.
[2] For each row, the percentages were calculated using the number of participants with given grade at baseline as the denominator; worst value on study is the worst grade observed post-baseline for a given participant.
- Grade 0 = Below Grade 1 and any grade in the other direction.
- For baseline NG means that grade does not apply at baseline.
- Grade 1 = Greater than ULN and increase from baseline of >0 - 2 g/dL;
Grade 2 = Greater than ULN and increase from baseline of >2 - 4 g/dL;
Grade 3 = Greater than ULN and increase from baseline of >4 g/dL.

Program: t_ctcshift_hem DATE(TIME): 18MAR24(21:59)

Laboratory grading is based on CTCAE Version 5.
Reference: Listing 2.8.1.1, 2.8.1.2

Figure 2: Shift table output page 1.

Table 3.3.3.1
Shift Summary of Hematology Laboratory Values in CTC Grade - to the Worst Abnormal Value
(Safety Population)

Laboratory Test (unit): Hemoglobin (g/L) - LOW DIRECTION

	Baseline [1]		Worst Post-Baseline Value [2]									
			Grade 0		Grade 1		Grade 2		Grade 3		Missing	
Treatment Group	Grade	n (%)	n	(%)	n	(%)	n	(%)	n	(%)	n	(%)
Drug 15 mg BID (N=77)	Grade 0	22 28.6%	2	9.1%	7	31.8%	6	27.3%	7	31.8%	0	
	Grade 1	38 49.4%	0		14	36.8%	14	36.8%	10	26.3%	0	
	Grade 2	13 16.9%	0		1	7.7%	0		11	84.6%	1	7.7%
	Grade 3	3 3.9%	0		0		1	33.3%	2	66.7%	0	
	Missing	1 1.3%	0		0		1	100.0%	0		0	
	Total	77 100.0%	2	2.6%	22	28.6%	22	28.6%	30	39.0%	1	1.3%
Drug 5 mg BID (N=87)	Grade 0	23 26.4%	2	8.7%	3	13.0%	11	47.8%	7	30.4%	0	
	Grade 1	41 47.1%	0		4	9.8%	18	43.9%	18	43.9%	1	2.4%
	Grade 2	17 19.5%	0		1	5.9%	2	11.8%	13	76.5%	1	5.9%
	Grade 3	3 3.4%	0		0		0		3	100.0%	0	
	Missing	3 3.4%	0		1	33.3%	0		2	66.7%	0	
	Total	87 100.0%	2	2.3%	9	10.3%	31	35.6%	43	49.4%	2	2.3%
Placebo (N=45)	Grade 0	14 31.1%	2	14.3%	4	28.6%	4	28.6%	2	14.3%	2	14.3%
	Grade 1	21 46.7%	1	4.8%	5	23.8%	4	19.0%	10	47.6%	1	4.8%
	Grade 2	6 13.3%	0		0		1	16.7%	4	66.7%	1	16.7%
	Grade 3	4 8.9%	0		0		1	25.0%	3	75.0%	0	
	Missing	0	0		0		0		0		0	
	Total	45 100.0%	3	6.7%	9	20.0%	10	22.2%	19	42.2%	4	8.9%

[1] The percentages were calculated using the baseline total as the denominator.
[2] For each row, the percentages were calculated using the number of participants with given grade at baseline as the denominator; worst value on study is the worst grade observed post-baseline for a given participant.
- Grade 0 = Below Grade 1 and any grade in the other direction.

Laboratory grading is based on CTCAE Version 5.
Reference: Listing 2.8.1.1, 2.8.1.2

Figure 3: Shift table output page 2.

Table 3.3.3.1
Shift Summary of Hematology Laboratory Values in CTC Grade - to the Worst Abnormal Value
(Safety Population)

Laboratory Test (unit): Leukocytes (GI/L) - HIGH DIRECTION

	Baseline [1]		Worst Post-Baseline Value [2]							
			Grade 0		NG		Grade 3		Missing	
Treatment Group	Grade	n (%)	n	(%)	n	(%)	n	(%)	n	(%)
Drug 15 mg BID (N=77)	Grade 0	31 40.3%	6	19.4%	25	80.6%	0		0	
	NG	45 58.4%	1	2.2%	43	95.6%	0		1	2.2%
	Grade 3	0	0		0		0		0	
	Missing	1 1.3%	0		1	100.0%	0		0	
	Total	77 100.0%	7	9.1%	69	89.6%	0		1	1.3%
Drug 5 mg BID (N=87)	Grade 0	36 41.4%	6	16.7%	29	80.6%	0		1	2.8%
	NG	49 56.3%	1	2.0%	47	95.9%	0		1	2.0%
	Grade 3	0	0		0		0		0	
	Missing	2 2.3%	0		2	100.0%	0		0	
	Total	87 100.0%	7	8.0%	78	89.7%	0		2	2.3%
Placebo (N=45)	Grade 0	14 31.1%	4	28.6%	8	57.1%	0		2	14.3%
	NG	31 68.9%	1	3.2%	28	90.3%	0		2	6.5%
	Grade 3	0	0		0		0		0	
	Missing	0	0		0		0		0	
	Total	45 100.0%	5	11.1%	36	80.0%	0		4	8.9%

[1] The percentages were calculated using the baseline total as the denominator.
[2] For each row, the percentages were calculated using the number of participants with given grade at baseline as the denominator; worst value on study is the worst grade observed post-baseline for a given participant.
- Grade 0 = Below Grade 1 and any grade in the other direction.
- For Leukocytes: Values between ULN and <= 100,000/mm3 are not graded (NG).

Laboratory grading is based on CTCAE Version 5.
Reference: Listing 2.8.1.1, 2.8.1.2

Figure 4: Shift table output page 3.

PROTOCOL: DIDA 00001-123 Page 4 of 4
DRUG/INDICATION: DIDA00001/COMPOUND-ASSOCIATED STUDY DATABASE VERSION: 10MAY2023
TLF Version: Final Database Lock (21APR2021) TASK: Primary Analysis

Table 3.3.3.1
Shift Summary of Hematology Laboratory Values in CTC Grade - to the Worst Abnormal Value
(Safety Population)

Laboratory Test (unit): Leukocytes (GI/L) - LOW DIRECTION

	Baseline [1]		Worst Post-Baseline Value [2]									
			Grade 0		Grade 1		Grade 2		Grade 3		Grade 4		Missing	
Treatment Group	Grade	n (%)	n	(%)	n	(%)	n	(%)	n	(%)	n	(%)	n	(%)
Drug 15 mg BID (N=77)	Grade 0	76 98.7%	70	92.1%	5	6.6%	0		0		0		1	1.3%
	Grade 1	0	0		0		0		0		0		0	
	Grade 2	0	0		0		0		0		0		0	
	Grade 3	0	0		0		0		0		0		0	
	Grade 4	0	0		0		0		0		0		0	
	Missing	1 1.3%	1	100.0%	0		0		0		0		0	
	Total	77 100.0%	71	92.2%	5	6.5%	0		0		0		1	1.3%
Drug 5 mg BID (N=87)	Grade 0	85 97.7%	77	90.6%	4	4.7%	1	1.2%	1	1.2%	0		2	2.4%
	Grade 1	0	0		0		0		0		0		0	
	Grade 2	0	0		0		0		0		0		0	
	Grade 3	0	0		0		0		0		0		0	
	Grade 4	0	0		0		0		0		0		0	
	Missing	2 2.3%	2	100.0%	0		0		0		0		0	
	Total	87 100.0%	79	90.8%	4	4.6%	1	1.1%	1	1.1%	0		2	2.3%
Placebo (N=45)	Grade 0	45 100.0%	41	91.1%	0		0		0		0		4	8.9%
	Grade 1	0	0		0		0		0		0		0	
	Grade 2	0	0		0		0		0		0		0	
	Grade 3	0	0		0		0		0		0		0	
	Grade 4	0	0		0		0		0		0		0	
	Missing	0	0		0		0		0		0		0	
	Total	45 100.0%	41	91.1%	0		0		0		0		4	8.9%

[1] The percentages were calculated using the baseline total as the denominator.
[2] For each row, the percentages were calculated using the number of participants with given grade at baseline as the denominator; worst value on study is the worst grade observed post-baseline for a given participant.
- Grade 0 = Below Grade 1 and any grade in the other direction.

Program: t_ctcshift_hem DATE(TIME): 18MAR24(21:59)

Laboratory grading is based on CTCAE Version 5.
Reference: Listing 2.8.1.1, 2.8.1.2

Figure 5: Shift table output page 4.

CONCLUSION

The **sassy** system simplifies the experience for SAS programmers working in R. This package brings R coding closer to the robust reporting capabilities inherent in SAS. By incorporating these features, the **sassy** package enhances efficiency and overall satisfaction when writing programs in R. Using these packages, the author was able to overcome several challenges when creating a lab shift table.

REFERENCES

Bosak D (2024). *The SASSY System*. R package version 1.2.3, https://github.com/dbosak01/sassy, https://www.r-sassy.org

Bosak D (2023). *An Overview of the SASSY System*, WUSS Paper 185-2023, https://www.lexjansen.com/wuss/2023/WUSS-2023-Paper-185.pdf

CONTACT INFORMATION

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

Vicky Yuan
vyuan@incyte.com
Incyte Corporation
1815 Augustine-cutoff, Wilmington, DE 19801
(302) 498-6947

David J. Bosak
dbosak01@gmail.com
Archytas Clinical Solutions, LLC
Kalamazoo, MI

APPENDIX A – SAMPLE DATA

```
# Sample Data
hemo <- read.table(header = TRUE, sep = ",", text = '
labtest,tmtnc,tmtn,swmgrade,variable,c0,c1,c2,c3,c4,c5,c6,c7
HEMO-HIGH,ARM A,1,7,NG,76  98.7%,75  98.7%,0,0,0,0,0,1   1.3%
HEMO-HIGH,ARM A,1,8,Missing,1   1.3%,1 100.0%,0,0,0,0,0,0
HEMO-HIGH,ARM A,1,9,Total,77 100.0%,76  98.7%,0,0,0,0,0,1   1.3%
HEMO-HIGH,ARM B,2,7,NG,84  96.6%,82  97.6%,0,0,0,0,0,2   2.4%
HEMO-HIGH,ARM B,2,8,Missing,3   3.4%,3 100.0%,0,0,0,0,0,0
HEMO-HIGH,ARM B,2,9,Total,87 100.0%,85  97.7%,0,0,0,0,0,2   2.3%
HEMO-HIGH,ARM C,3,7,NG,45 100.0%,41  91.1%,0,0,0,0,0,4   8.9%
HEMO-HIGH,ARM C,3,8,Missing,0,0,0,0,0,0,0,0
HEMO-HIGH,ARM C,3,9,Total,45 100.0%,41  91.1%,0,0,0,0,0,4   8.9%
HEMO-LOW,ARM A,1,1,Grade 0,22  28.6%,2   9.1%,7  31.8%,6  27.3%,7
31.8%,0,0,0
HEMO-LOW,ARM A,1,3,Grade 1,38  49.4%,0,14  36.8%,14  36.8%,10  26.3%,0,0,0
HEMO-LOW,ARM A,1,4,Grade 2,13  16.9%,0,1   7.7%,0,11  84.6%,0,0,1   7.7%
HEMO-LOW,ARM A,1,5,Grade 3,3   3.9%,0,0,1  33.3%,2  66.7%,0,0,0
HEMO-LOW,ARM A,1,8,Missing,1   1.3%,0,0,1 100.0%,0,0,0,0
HEMO-LOW,ARM A,1,9,Total,77 100.0%,2   2.6%,22  28.6%,22  28.6%,30
39.0%,0,0,1   1.3%
HEMO-LOW,ARM B,2,1,Grade 0,23  26.4%,2   8.7%,3  13.0%,11  47.8%,7
30.4%,0,0,0
HEMO-LOW,ARM B,2,3,Grade 1,41  47.1%,0,4   9.8%,18  43.9%,18  43.9%,0,0,1
2.4%
HEMO-LOW,ARM B,2,4,Grade 2,17  19.5%,0,1   5.9%,2  11.8%,13  76.5%,0,0,1
5.9%
HEMO-LOW,ARM B,2,5,Grade 3,3   3.4%,0,0,0,3 100.0%,0,0,0
HEMO-LOW,ARM B,2,8,Missing,3   3.4%,0,1  33.3%,0,2  66.7%,0,0,0
HEMO-LOW,ARM B,2,9,Total,87 100.0%,2   2.3%,9  10.3%,31  35.6%,43
49.4%,0,0,2   2.3%
HEMO-LOW,ARM C,3,1,Grade 0,14  31.1%,2  14.3%,4  28.6%,4  28.6%,2
14.3%,0,0,2  14.3%
HEMO-LOW,ARM C,3,3,Grade 1,21  46.7%,1   4.8%,5  23.8%,4  19.0%,10
47.6%,0,0,1   4.8%
HEMO-LOW,ARM C,3,4,Grade 2,6  13.3%,0,0,1  16.7%,4  66.7%,0,0,1  16.7%
HEMO-LOW,ARM C,3,5,Grade 3,4   8.9%,0,0,1  25.0%,3  75.0%,0,0,0
HEMO-LOW,ARM C,3,8,Missing,0,0,0,0,0,0,0,0
HEMO-LOW,ARM C,3,9,Total,45 100.0%,3   6.7%,9  20.0%,10  22.2%,19
42.2%,0,0,4   8.9%
LEUK-HIGH,ARM A,1,1,Grade 0,31  40.3%,6  19.4%,0,0,0,0,25  80.6%,0
LEUK-HIGH,ARM A,1,3,NG,45  58.4%,1   2.2%,0,0,0,0,43  95.6%,1   2.2%
LEUK-HIGH,ARM A,1,5,Grade 3,0,0,0,0,0,0,0,0
LEUK-HIGH,ARM A,1,8,Missing,1   1.3%,0,0,0,0,0,1 100.0%,0
LEUK-HIGH,ARM A,1,9,Total,77 100.0%,7   9.1%,0,0,0,0,69  89.6%,1   1.3%
LEUK-HIGH,ARM B,2,1,Grade 0,36  41.4%,6  16.7%,0,0,0,0,29  80.6%,1   2.8%
LEUK-HIGH,ARM B,2,3,NG,49  56.3%,1   2.0%,0,0,0,0,47  95.9%,1   2.0%
LEUK-HIGH,ARM B,2,5,Grade 3,0,0,0,0,0,0,0,0
LEUK-HIGH,ARM B,2,8,Missing,2   2.3%,0,0,0,0,0,2 100.0%,0
LEUK-HIGH,ARM B,2,9,Total,87 100.0%,7   8.0%,0,0,0,0,78  89.7%,2   2.3%
LEUK-HIGH,ARM C,3,1,Grade 0,14  31.1%,4  28.6%,0,0,0,0,8  57.1%,2  14.3%
LEUK-HIGH,ARM C,3,3,NG,31  68.9%,1   3.2%,0,0,0,0,28  90.3%,2   6.5%
LEUK-HIGH,ARM C,3,5,Grade 3,0,0,0,0,0,0,0,0
LEUK-HIGH,ARM C,3,8,Missing,0,0,0,0,0,0,0,0
LEUK-HIGH,ARM C,3,9,Total,45 100.0%,5  11.1%,0,0,0,0,36  80.0%,4   8.9%
LEUK-LOW,ARM A,1,1,Grade 0,76  98.7%,70  92.1%,5   6.6%,0,0,0,0,1   1.3%
```

```
LEUK-LOW,ARM A,1,3,Grade 1,0,0,0,0,0,0,0,0
LEUK-LOW,ARM A,1,4,Grade 2,0,0,0,0,0,0,0,0
LEUK-LOW,ARM A,1,5,Grade 3,0,0,0,0,0,0,0,0
LEUK-LOW,ARM A,1,6,Grade 4,0,0,0,0,0,0,0,0
LEUK-LOW,ARM A,1,8,Missing,1   1.3%,1 100.0%,0,0,0,0,0,0
LEUK-LOW,ARM A,1,9,Total,77 100.0%,71  92.2%,5   6.5%,0,0,0,0,1   1.3%
LEUK-LOW,ARM B,2,1,Grade 0,85  97.7%,77  90.6%,4   4.7%,1   1.2%,1
1.2%,0,0,2   2.4%
LEUK-LOW,ARM B,2,3,Grade 1,0,0,0,0,0,0,0,0
LEUK-LOW,ARM B,2,4,Grade 2,0,0,0,0,0,0,0,0
LEUK-LOW,ARM B,2,5,Grade 3,0,0,0,0,0,0,0,0
LEUK-LOW,ARM B,2,6,Grade 4,0,0,0,0,0,0,0,0
LEUK-LOW,ARM B,2,8,Missing,2   2.3%,2 100.0%,0,0,0,0,0,0
LEUK-LOW,ARM B,2,9,Total,87 100.0%,79  90.8%,4   4.6%,1   1.1%,1   1.1%,0,0,2
2.3%
LEUK-LOW,ARM C,3,1,Grade 0,45 100.0%,41  91.1%,0,0,0,0,0,4   8.9%
LEUK-LOW,ARM C,3,3,Grade 1,0,0,0,0,0,0,0,0
LEUK-LOW,ARM C,3,4,Grade 2,0,0,0,0,0,0,0,0
LEUK-LOW,ARM C,3,5,Grade 3,0,0,0,0,0,0,0,0
LEUK-LOW,ARM C,3,6,Grade 4,0,0,0,0,0,0,0,0
LEUK-LOW,ARM C,3,8,Missing,0,0,0,0,0,0,0,0
LEUK-LOW,ARM C,3,9,Total,45 100.0%,41  91.1%,0,0,0,0,0,4   8.9%
')
```

APPENDIX B – COMPLETE CODE

```
library(sassy)
library(dplyr)

# Set variables
program.name <- "t_ctcshift_hem"
program.output <- "T_3_3_3_1_ctcshift_hem"
program.timestamp <- as.POSIXlt(Sys.time(), "UTC") %>%
  strftime("%d%b%y(%H:%M)") %>% toupper()
program.dir <- dirname(Sys.path())

options("logr.autolog" = TRUE, "logr.notes" = FALSE)

# Open log
logpth <- log_open(file.path(program.dir, program.output))


# Prepare Formats ---------------------------------------------------------

sep("Prepare Formats")

fmt1 <- value(condition(x == "HEMO-HIGH", "Hemoglobin (G/L) - HIGH
DIRECTION"),
              condition(x == "HEMO-LOW", "Hemoglobin (g/L) - LOW
DIRECTION"),
              condition(x == "LEUK-HIGH", "Leukocytes (GI/L) - HIGH
DIRECTION"),
              condition(x == "LEUK-LOW", "Leukocytes (GI/L) - LOW
DIRECTION"))


fmt2 <- value(condition(x == "ARM A", "Drug 15 mg BID (N=77)"),
              condition(x == "ARM B", "Drug 5 mg BID (N=87)"),
              condition(x == "ARM C", "Placebo (N=45)"))

# Prepare Footnotes -------------------------------------------------------

sep("Prepare Footnotes")

ftnts <- list()

base_ftnt <-
  c("[1] The percentages were calculated using the baseline total as the
denominator.",
    paste("[2] For each row, the percentages were calculated using the
number of participants",
    "with given grade at baseline as the denominator; worst value on study
is the worst",
    "grade observed post-baseline for a given participant."))

ftnts[["HEMO-HIGH"]] <-
  c(base_ftnt,
    "-  Grade 0 = Below Grade 1 and any grade in the other direction.",
    "-  For baseline NG means that grade does not apply at baseline.",
```

```
    "-  Grade 1 = Greater than ULN and increase from baseline of  >0 - 2
g/dL;",
    "Grade 2 = Greater than ULN and increase from baseline of  >2 - 4
g/dL;",
    "Grade 3 = Greater than ULN and increase from baseline of  >4 g/dL. ")


ftnts[["HEMO-LOW"]] <-
  c(base_ftnt,
    "-  Grade 0 = Below Grade 1 and any grade in the other direction.")


ftnts[["LEUK-HIGH"]] <-
  c(base_ftnt,
    "-  Grade 0 = Below Grade 1 and any grade in the other direction.",
    "-  For Leukocytes: Values between ULN and <= 100,000/mm3 are not
graded (NG). ")

ftnts[["LEUK-LOW"]] <-
  c(base_ftnt,
    "-  Grade 0 = Below Grade 1 and any grade in the other direction.")

put(ftnts)

# Prepare Columns -----------------------------------------------------

sep("Prepare Columns")

cols <- list()

cols[["HEMO-HIGH"]] <- c(c1 = "Grade 0\n n  (%)", c2 = "Grade 1\n n  (%)",
                         c3 = "Grade 2\n n  (%)", c4 = "Grade 3\n n  (%)",
                         c7 = "Missing\n n  (%)")

cols[["HEMO-LOW"]] <- c(c1 = "Grade 0\n n  (%)", c2 = "Grade 1\n n  (%)",
                        c3 = "Grade 2\n n  (%)", c4 = "Grade 3\n n  (%)",
                        c7 = "Missing\n n  (%)")

cols[["LEUK-HIGH"]] <- c(c1 = "Grade 0\n n  (%)", c6 = "NG\n n  (%)",
                        c4 = "Grade 3\n n  (%)", c7 = "Missing\n n  (%)")

cols[["LEUK-LOW"]] <- c(c1 = "Grade 0\n n  (%)", c2 = "Grade 1\n n  (%)",
                        c3 = "Grade 2\n n  (%)", c4 = "Grade 3\n n  (%)",
                        c5 = "Grade 4\n n  (%)", c7 = "Missing\n n  (%)")

# Print Report --------------------------------------------------------

sep("Print Report")

pth <- file.path(program.dir, "output", program.output)

put("Create the report")
rpt <- create_report(pth, font = "Courier", font_size = 9)  %>%
  set_margins(top = 1.0, left = 1, right = 1, bottom = .5) %>%
  options_fixed(line_count = 51) %>%

  titles("Table 3.3.3.1",
```

```
         paste("Shift Summary of Hematology Laboratory Values",
         "in CTC Grade - to the Worst Abnormal Value"),
         "(Safety Population)", bold = TRUE, font_size = 9) %>%
  page_header(left = c("PROTOCOL: DIDA 00001-123",
                       "DRUG/INDICATION: DIDA00001/COMPOUND-ASSOCIATED
STUDY",
                       "TLF Version: Final Database Lock (21APR2021)"),
              right = c("Page [pg] of [tpg]",  "DATABASE VERSION:
10MAY2023",
                        "TASK: Primary Analysis")) %>%
  footnotes(paste0("Program: ", program.name, sep=""),
            "DATE(TIME): " %p% toupper(fapply(Sys.time(),
"%d%h%y(%H:%M)")),
            blank_row = "none", borders = "top", columns = 2, footer =
TRUE)  %>%
  footnotes("Laboratory grading is based on CTCAE Version 5.",
            "Reference: Listing 2.8.1.1, 2.8.1.2", footer = TRUE )

put("Get unique lab tests")
labtests <- proc_sort(hemo, by = labtest, options = nodupkey,
                      keep = labtest)

put("Loop through lab tests")
for (i in seq_len(nrow(labtests))) {

  put("Get lab test")
  lb <- labtests[[i, "labtest"]] %>% put()

  put("Get footnotes for this lab test")
  ftnt <- ftnts[[lb]]

  put("Filter data for this lab test")
  table_hemo <- hemo %>%
    dplyr::filter(labtest==lb) %>%
    mutate(labtest = fapply(labtest, fmt1),
           tmtnc = fapply(tmtnc, fmt2)) %>%
    select(labtest, tmtnc, tmtn, swmgrade,variable,
           c0,c1,c2,c3,c4,c5,c6,c7) %>%
    arrange(labtest, tmtn, swmgrade)

  put("Create table for this lab test")
  tbl <- create_table(table_hemo,
                      show_cols = "none",
                      borders = "top",
                      width = 9) %>%
    page_by(labtest, label = "Laboratory Test (unit):", borders = "none",
            blank_row = "none") %>%
    footnotes (ftnt, blank_row ="none" ) %>%
    spanning_header(variable, c0, label="Baseline [1]") %>%
    spanning_header(c1, c7, label="Worst Post-Baseline Value [2]") %>%
    define(labtest, visible = FALSE) %>%
    define(tmtnc, dedupe = TRUE, align = "left",
           label = "Treatment Group") %>%
    define(tmtn, blank_after = TRUE, visible = FALSE) %>%
    define(variable, align="left", label="Grade", width=.8) %>%
    define(c0, align="left", label="n  (%)", width=1)
```

```
    # Get column specs
    cls <- cols[[lb]]

    # If more than 5 cols, reduce width
    wdth <- ifelse(length(cls) > 5, .85, 1)

    # Add column definitions dynamically for each test
    for (clnm in names(cls)) {
      tbl <- define(tbl, clnm, align = "left", label = cls[[clnm]],
                    width = wdth,
                    standard_eval = TRUE)
    }

  put("Add table to report dynamically")
  rpt <- rpt %>% add_content(tbl, blank_row = "none", page_break = TRUE)

}

put("Write out the report")
res <- write_report(rpt, output_type = "PDF")


log_close()


# View the report
file.show(res$modified_path)
file.show(logpth)
```