PharmaSUG 2024 - Paper AP-263

Creating Adverse Event Tables using R and SASSY

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

ABSTRACT

Currently, pharmaceutical industry is focusing on adopting R language for the creation of submission packages and TLFs generation. SAS programmers who come to R are often frustrated because there are many reporting packages in R. Selecting a right package for programmers became more and more important. If a certain R package can provide functions, which have an ability to create data libraries, traceable log, format catalogs and easy way to reports as DOCX, rich-text files like what SAS did. **sassy** system seems meet all the criteria.

This paper will provide on-hand overview of generating adverse event table by using **sassy** system.

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

INTRODUCTION

Adverse event by system organ class and preferred term tables plays a vital role in clinical trial studies, showcasing the number of subjects experiencing specific organ class and preferred term issues. While achieving this in SAS requires only a few lines of code. In R, it may demand dozens or even hundreds. Tasks like creating logs, loading data files, or generating reports can become cumbersome. The **sassy** system tackles these hurdles head-on, streamlining the R experience for SAS programmers. Leveraging specialized packages, it narrows the disparity between R coding and robust reporting capabilities of SAS. By seamlessly integrating these features, the **sassy** system enhances efficiency and satisfaction in R programming. With the **sassy** system, you can:

- Create a log
- Do a data step
- Generate frequency and summary statistics
- Write a report in a few lines of code

This paper utilizes an adverse event table to show case the efficacy of the **sassy** system.

INSTALLATION

The **sassy** meta-package is published on CRAN. You can install it with the following console command:

```
Install.packages("sassy")
Library(sassy)
```

The commands provided above will install and load a suite of packages designed to align your approach to programming in R closely with how you approached programming in SAS.

The **sassy** system contains the following packages:

• **logr**: To create a traceable log

• **libr**: To create a libname, a data dictionary, and perform a data step

• **fmtr**: To bring simplicity, flexibility, and power to data formatting

• **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.

ADVERSE EVENT TABLE CHALLENGES

The challenge for this table is the layout, percentage calculation and actual report generation. 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) **Spanning Headers**: Report requires spanning headers for baseline and post-baseline.

(c) **Percentage Calculation**: using **`fmt_cnt_pct`** function from **fmtr** package will make it easy.

(d) **Page Footer**: The page footer has items aligned left and right, and center. it is positioned at the bottom of the page or above the page footnotes

The above challenges are identified on the image below:

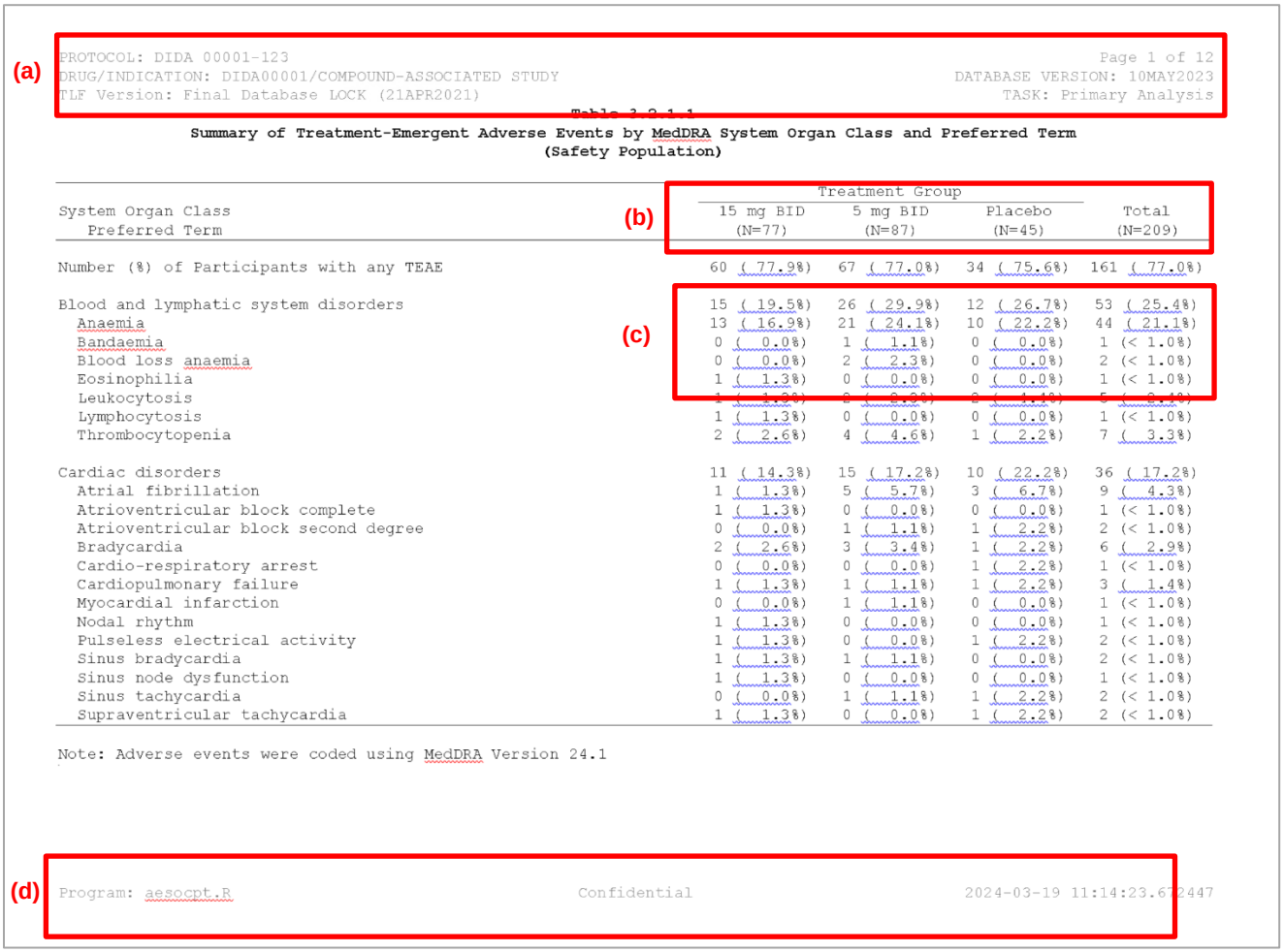


Figure 1. Adverse Event table challenge areas

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 a 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_aesocpt"
program.output <- "T_3_2_1_1_aesocpt"
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, paste0(program.output, ".log",sep='')))
```

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

STEP 3: PREPARE PERCENTAGE CALCULATION

The third step is the program is to prepare percentage calculation for "Any TEAE", "System Organ Class" and "Preferred Term". These calculations will ***fmt_cnt_pct*** function from the **fmtr** package.

1) **Population N count**:

```
adsl <- adsl %>%
  dplyr::filter(saffl == "Y")

adsl.total <- adsl %>%
  mutate(
    tmtn=4,
    tmt="Total"
  )
adsl.all <- rbind(adsl, adsl.total)

bigN2 <- adsl.all %>%
  select(usubjid, tmtn) %>%
  distinct()  %>%
```

```
  count(tmtn) %>%
  deframe() %>% put()
```

in thus bigN2 values for each treatment group is created.

2) Calculate “Number of Participants with any TEAE”

```
any_ae <- adae_all %>%
  group_by(usubjid, tmtn, tmt) %>%
  arrange(usubjid, tmtn, -atoxgrn) %>%
  slice_head(n=1)

any_ae1 <- any_ae %>%
  group_by(tmtn, tmt) %>%
  summarise(n=n(), .groups = "keep") %>%
  pivot_wider(names_from = c(tmtn, tmt),
              values_from = n,
              values_fill = 0) %>%
  mutate (
    aebodsys="Number (%) of Participants with any TEAE",
    order1 = 0,
    aedecod=NA
  )

#put("Format counts and percents for each column")
any_ae2 <- any_ae1 %>%
  transmute(aebodsys = aebodsys,
            order1 = order1,
            aedecod = str_to_title(aedecod),
            `1_15 mg BID` = fmt_cnt_pct(`1_15 mg BID`, bigN2["1"]),
            `2_5 mg BID` = fmt_cnt_pct(`2_5 mg BID`, bigN2["2"]),
            `3_Placebo` = fmt_cnt_pct(`3_Placebo`, bigN2["3"]),
            `4_Total` = fmt_cnt_pct(`4_Total`, bigN2["4"]),
           )%>%
  arrange(aebodsys, aedecod) %>%
  ungroup()
```

System Organ Class Preferred Term	Treatment Group 15 mg BID (N=77)	 5 mg BID (N=87)	 Placebo (N=45)	 Total (N=209)
Number (%) of Participants with any TEAE	60 (77.9%)	67 (77.0%)	34 (75.6%)	161 (77.0%)

3) Calculate AE SOC

```
any_soc <-adae_all %>%
  group_by(usubjid, tmtn, tmt, aebodsys) %>%
  arrange(usubjid,tmtnn, aebodsys, -atoxgrn) %>%
  slice_head(n=1)

any_soc1 <- any_soc %>%
  group_by(tmtn, tmt, aebodsys) %>%
  summarise(n=n(), .groups = "keep") %>%
  pivot_wider(names_from = c(tmtn, tmt),
              values_from = n,
              values_fill = 0) %>%

  mutate (
    aedecod=NA,
```

```
    order1 = 1,
  )

#put("Format counts and percents for each column")
any_soc2 <- any_soc1 %>%
  transmute(aebodsys = aebodsys,
            aedecod  = NA,
            order1 = 1,
            `1_15 mg BID` = fmt_cnt_pct(`1_15 mg BID`, bigN2["1"]),
            `2_5 mg BID` = fmt_cnt_pct(`2_5 mg BID`, bigN2["2"]),
            `3_Placebo` = fmt_cnt_pct(`3_Placebo`, bigN2["3"]),
            `4_Total` = fmt_cnt_pct(`4_Total`, bigN2["4"]),
  )%>%
  arrange(aebodsys) %>%
  ungroup()
```

System Organ Class Preferred Term	Treatment Group 15 mg BID (N=77)	 5 mg BID (N=87)	 Placebo (N=45)	 Total (N=209)
Number (%) of Participants with any TEAE	60 (77.9%)	67 (77.0%)	34 (75.6%)	161 (77.0%)
Blood and lymphatic system disorders	15 (19.5%)	26 (29.9%)	12 (26.7%)	53 (25.4%)

4) Calculate AE SOC and PT

```
any_socpt <-adae_all %>%
  group_by(usubjid, tmt, tmtn, aebodsys, aedecod) %>%
  arrange(usubjid, tmtn, aebodsys, aedecod, -atoxgrn) %>%
  slice_head(n=1) %>%
  select(usubjid, tmt, tmtn, aebodsys, aedecod) %>%
  arrange(tmtn, aebodsys, aedecod) %>%
  mutate(
    socpt = paste(aebodsys, aedecod, sep='/')
  )

any_socpt1 <- any_socpt %>%
  group_by(tmtn, tmt, aebodsys,aedecod) %>%
  summarise(n=n(), .groups = "keep") %>%
  pivot_wider(names_from = c(tmtn, tmt),
              values_from = n,
              values_fill = 0) %>%
  mutate (
    aedecod=aedecod,
    order1 = 2,

  )

#put("Format counts and percents for each column")
any_socpt2 <- any_socpt1 %>%
  transmute(aebodsys = aebodsys,
            aedecod  = aedecod,
            order1 = 2,
            `1_15 mg BID` = fmt_cnt_pct(`1_15 mg BID`, bigN2["1"]),
            `2_5 mg BID` = fmt_cnt_pct(`2_5 mg BID`, bigN2["2"]),
            `3_Placebo` = fmt_cnt_pct(`3_Placebo`, bigN2["3"]),
            `4_Total` = fmt_cnt_pct(`4_Total`, bigN2["4"], format='%5.1f'),
  )%>%
```

```
  arrange(aebodsys,aedecod) %>%
  ungroup()
```

System Organ Class Preferred Term	Treatment Group 15 mg BID (N=77)	 5 mg BID (N=87)	 Placebo (N=45)	Total (N=209)
Number (%) of Participants with any TEAE	60 (77.9%)	67 (77.0%)	34 (75.6%)	161 (77.0%)
Blood and lymphatic system disorders	15 (19.5%)	26 (29.9%)	12 (26.7%)	53 (25.4%)
Anaemia	13 (16.9%)	21 (24.1%)	10 (22.2%)	44 (21.1%)
Bandaemia	0 (0.0%)	1 (1.1%)	0 (0.0%)	1 (< 1.0%)
Blood loss anaemia	0 (0.0%)	2 (2.3%)	0 (0.0%)	2 (< 1.0%)
Eosinophilia	1 (1.3%)	0 (0.0%)	0 (0.0%)	1 (< 1.0%)
Leukocytosis	1 (1.3%)	2 (2.3%)	2 (4.4%)	5 (2.4%)
Lymphocytosis	1 (1.3%)	0 (0.0%)	0 (0.0%)	1 (< 1.0%)
Thrombocytopenia	2 (2.6%)	4 (4.6%)	1 (2.2%)	7 (3.3%)

By employing the **fmt_cnt_pct** function, you can compute xx (xx.x%). This format is specified as “%5.1f”, presenting the value with one decimal place. Should the calculated percentage fall between 0% and 1%, the function will exhibit “(<1.0%)” as the percentage value. Zero values will appear as “(0.0%)”.

If you prefer not to append “%” as the end of the value, the **sprint** function can be utilized for this objective. The R code would be structured as follows:

```
any_socpt2 <- any_socpt1 %>%
  transmute(aebodsys = aebodsys,
            aedecod  = aedecod,
            order1 = 2,
            `1_15 mg BID` = sprintf('%d (%5.1f)', `1_15 mg BID`, `1_15 mg
BID`/bigN2["1"]*100),
            `2_5 mg BID` = sprintf('%d (%5.1f)', `2_5 mg BID`, `2_5 mg
BID`/bigN2["2"]*100),
            `3_Placebo` = sprintf('%d (%5.1f)', `3_Placebo`,
`3_Placebo`/bigN2["3"]*100),
            `4_Total` = sprintf('%d (%5.1f)', `4_Total`,
`4_Total`/bigN2["4"]*100),
  )%>%
  arrange(aebodsys,aedecod) %>%
  ungroup()
```

System Organ Class Preferred Term	Treatment Group 15 mg BID (N=77)	 5 mg BID (N=87)	 Placebo (N=45)	Total (N=209)
Number (%) of Participants with any TEAE	60 (77.9%)	67 (77.0%)	34 (75.6%)	161 (77.0%)
Blood and lymphatic system disorders	15 (19.5%)	26 (29.9%)	12 (26.7%)	53 (25.4%)
Anaemia	13 (16.9)	21 (24.1)	10 (22.2)	44 (21.1)
Bandaemia	0 (0.0)	1 (1.1)	0 (0.0)	1 (0.5)
Blood loss anaemia	0 (0.0)	2 (2.3)	0 (0.0)	2 (1.0)
Eosinophilia	1 (1.3)	0 (0.0)	0 (0.0)	1 (0.5)
Leukocytosis	1 (1.3)	2 (2.3)	2 (4.4)	5 (2.4)
Lymphocytosis	1 (1.3)	0 (0.0)	0 (0.0)	1 (0.5)
Thrombocytopenia	2 (2.6)	4 (4.6)	1 (2.2)	7 (3.3)

5) Create final data frame

```
final <- bind_rows(any_soc2, any_socpt2) %>%
  arrange(aebodsys,order1, aedecod)
final <- bind_rows(any_ae2, final)
```

STEP 4: CREATE REPORT

Typically, when generating a report with the reporter package, you'd begin by crafting the content. Subsequently, you'd proceed to construct the report and incorporate the content within it. The **sassy** system introduces a function enabling users to generate custom page breaks. Below is the code to implement a page break:

```
datastep(final, by = c('order2', 'aebodsys','order1', 'aedecod', 'order3'),
         retain = list(ptcnt = 0, PG = 1),
         {
           if(first.){
             ptcnt <- ptcnt +1
           }

           if (ptcnt == 13){
             ptcnt <- 1
             PG <- PG+1
           }

         }) -> final
```

Alternatively, you can employ the automated page break function within the **sassy** system. Below is the code to generate the report:

```
  put("Create the table")
  tbl <- create_table(final, first_row_blank=TRUE, borders = c("bottom", "top"),
  width=9)%>%
    column_defaults(from = '1_15 mg BID', to = '4_Total', width=1) %>%
    spanning_header("1_15 mg BID", "3_Placebo", label="Treatment Group") %>%
    stub(vars = v(aebodsys, aedecod), label = "System Organ Class\n   Preferred
  Term", width = 5) %>%
    define(aebodsys, blank_after = TRUE) %>%
    define(aedecod, indent = .25) %>%
    define(order1, visible=FALSE) %>%
    define("1_15 mg BID", align = "center", label = "15 mg BID", n= bigN2["1"])%>%
    define("2_5 mg BID", align = "center", label = "5 mg BID", n= bigN2["2"])%>%
    define("3_Placebo", align = "center", label = "Placebo",   n= bigN2["3"])%>%
    define("4_Total", align = "center", label = "Total",       n= bigN2["4"])
```

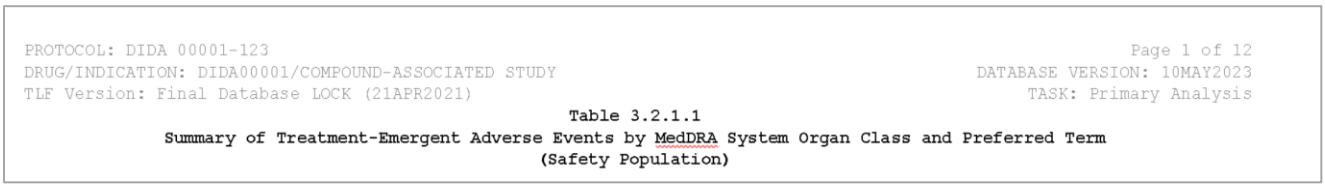
System Organ Class Preferred Term	Treatment Group 15 mg BID (N=77)	 5 mg BID (N=87)	 Placebo (N=45)	 Total (N=209)

Note that the preceding code encompasses the instructions to generate the header line. Which spans both the header and variables labels. The "indent" parameter enables you to establish an indentation in the output.

After using `create_table` function to set up content then we need to use `create_report` function to create the report. Below is the code:

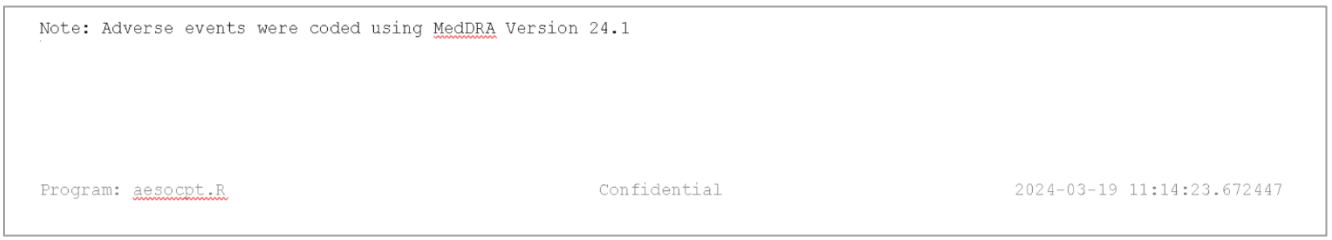
```
  put("Create the report")
  rpt <- create_report(pth, font = "Courier", font_size = 9)  %>%
```

```
  set_margins(top = 1.0, left = 1, right = 1, bottom = .5) %>%
  options_fixed(line_count = 40) %>%

  titles("Table 3.2.1.1", "Summary of Treatment-Emergent Adverse Events by MedDRA
System Organ Class and Preferred Term",
         "(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")) %>%

  add_content(tbl)  %>%
  page_footer("Program: aesocpt.R",  right = Sys.time(), center = "Confidential")
%>%
  footnotes("Note: Adverse events were coded using MedDRA Version 24.1")
```



PROTOCOL: DIDA 00001-123 Page 1 of 12
DRUG/INDICATION: DIDA00001/COMPOUND-ASSOCIATED STUDY DATABASE VERSION: 10MAY2023
TLF Version: Final Database LOCK (21APR2021) TASK: Primary Analysis

Table 3.2.1.1
Summary of Treatment-Emergent Adverse Events by MedDRA System Organ Class and Preferred Term
(Safety Population)



Note: Adverse events were coded using MedDRA Version 24.1

Program: aesocpt.R Confidential 2024-03-19 11:14:23.672447

Note that the above code includes the code to produce the table header, page footer, and footnotes at the bottom of the page.

STEP 7: WRITE REPORT AND CLOSE LOG

The final step involves writing out the report generated in the preceding steps. This is the crucial stage where the report is reridered and saved to a file. In this instance, we opt for a DOCX output format. However, it's worth noting that the reporter package offers the flexibility to generate reports in RTF, PDF, HTML, and TXT formats as well. Below is the code to accomplish this task:

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

log_close()
```

OUTPUT

Below are the DOCX pages generated throught the outlined steps. I'm presenting only the first two pages of the results for your reference.

Table 3.2.1.1
Summary of Treatment-Emergent Adverse Events by MedDRA System Organ Class and Preferred Term
(Safety Population)

System Organ Class Preferred Term	Treatment Group 15 mg BID (N=77)	 5 mg BID (N=87)	 Placebo (N=45)	 Total (N=209)
Number (%) of Participants with any TEAE	60 (77.9%)	67 (77.0%)	34 (75.6%)	161 (77.0%)
Blood and lymphatic system disorders	15 (19.5%)	26 (29.9%)	12 (26.7%)	53 (25.4%)
Anaemia	13 (16.9%)	21 (24.1%)	10 (22.2%)	44 (21.1%)
Bandaemia	0 (0.0%)	1 (1.1%)	0 (0.0%)	1 (< 1.0%)
Blood loss anaemia	0 (0.0%)	2 (2.3%)	0 (0.0%)	2 (< 1.0%)
Eosinophilia	1 (1.3%)	0 (0.0%)	0 (0.0%)	1 (< 1.0%)
Leukocytosis	1 (1.3%)	2 (2.3%)	2 (4.4%)	5 (2.4%)
Lymphocytosis	1 (1.3%)	0 (0.0%)	0 (0.0%)	1 (< 1.0%)
Thrombocytopenia	2 (2.6%)	4 (4.6%)	1 (2.2%)	7 (3.3%)
Cardiac disorders	11 (14.3%)	15 (17.2%)	10 (22.2%)	36 (17.2%)
Atrial fibrillation	1 (1.3%)	5 (5.7%)	3 (6.7%)	9 (4.3%)
Atrioventricular block complete	1 (1.3%)	0 (0.0%)	0 (0.0%)	1 (< 1.0%)
Atrioventricular block second degree	0 (0.0%)	1 (1.1%)	1 (2.2%)	2 (< 1.0%)
Bradycardia	2 (2.6%)	3 (3.4%)	1 (2.2%)	6 (2.9%)
Cardio-respiratory arrest	0 (0.0%)	0 (0.0%)	1 (2.2%)	1 (< 1.0%)
Cardiopulmonary failure	1 (1.3%)	1 (1.1%)	1 (2.2%)	3 (1.4%)
Myocardial infarction	0 (0.0%)	1 (1.1%)	0 (0.0%)	1 (< 1.0%)
Nodal rhythm	1 (1.3%)	0 (0.0%)	0 (0.0%)	1 (< 1.0%)
Pulseless electrical activity	1 (1.3%)	0 (0.0%)	1 (2.2%)	2 (< 1.0%)
Sinus bradycardia	1 (1.3%)	1 (1.1%)	0 (0.0%)	2 (< 1.0%)
Sinus node dysfunction	1 (1.3%)	0 (0.0%)	0 (0.0%)	1 (< 1.0%)
Sinus tachycardia	0 (0.0%)	1 (1.1%)	1 (2.2%)	2 (< 1.0%)
Supraventricular tachycardia	1 (1.3%)	0 (0.0%)	1 (2.2%)	2 (< 1.0%)

Note: Adverse events were coded using MedDRA Version 24.1

Figure 2: Adverse event table output page 1.

Table 3.2.1.1
Summary of Treatment-Emergent Adverse Events by MedDRA System Organ Class and Preferred Term
(Safety Population)

System Organ Class Preferred Term	Treatment Group 15 mg BID (N=77)	 5 mg BID (N=87)	 Placebo (N=45)	 Total (N=209)
Tachycardia	2 (2.6%)	1 (1.1%)	0 (0.0%)	3 (1.4%)
Torsade de pointes	0 (0.0%)	1 (1.1%)	0 (0.0%)	1 (< 1.0%)
Ventricular arrhythmia	0 (0.0%)	2 (2.3%)	0 (0.0%)	2 (< 1.0%)
Ventricular tachyarrhythmia	0 (0.0%)	1 (1.1%)	0 (0.0%)	1 (< 1.0%)
Ventricular tachycardia	1 (1.3%)	0 (0.0%)	0 (0.0%)	1 (< 1.0%)
Ear and labyrinth disorders	1 (1.3%)	0 (0.0%)	0 (0.0%)	1 (< 1.0%)
Vertigo positional	1 (1.3%)	0 (0.0%)	0 (0.0%)	1 (< 1.0%)
Endocrine disorders	1 (1.3%)	1 (1.1%)	1 (2.2%)	3 (1.4%)
Adrenal insufficiency	1 (1.3%)	1 (1.1%)	1 (2.2%)	3 (1.4%)
Gastrointestinal disorders	16 (20.8%)	19 (21.8%)	9 (20.0%)	44 (21.1%)
Abdominal distension	0 (0.0%)	0 (0.0%)	1 (2.2%)	1 (< 1.0%)
Abdominal pain	1 (1.3%)	1 (1.1%)	0 (0.0%)	2 (< 1.0%)
Abdominal pain upper	0 (0.0%)	0 (0.0%)	1 (2.2%)	1 (< 1.0%)
Acquired macroglossia	1 (1.3%)	0 (0.0%)	0 (0.0%)	1 (< 1.0%)
Chronic gastritis	0 (0.0%)	1 (1.1%)	0 (0.0%)	1 (< 1.0%)
Colitis ischaemic	0 (0.0%)	1 (1.1%)	0 (0.0%)	1 (< 1.0%)
Constipation	5 (6.5%)	6 (6.9%)	2 (4.4%)	13 (6.2%)
Diarrhoea	2 (2.6%)	1 (1.1%)	2 (4.4%)	5 (2.4%)
Duodenal ulcer perforation	1 (1.3%)	0 (0.0%)	0 (0.0%)	1 (< 1.0%)
Duodenogastric reflux	0 (0.0%)	1 (1.1%)	0 (0.0%)	1 (< 1.0%)
Dysphagia	1 (1.3%)	1 (1.1%)	1 (2.2%)	3 (1.4%)
Gastric haemorrhage	0 (0.0%)	0 (0.0%)	1 (2.2%)	1 (< 1.0%)

Note: Adverse events were coded using MedDRA Version 24.1

Figure 2: Adverse event table output page 2.

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 adverse event 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