

# Create a Shift Table with gt Package in R

Hengwei Liu, Hengrui USA, Princeton, New Jersey

## ABSTRACT

Shift tables for CTCAE grades are often required in oncology studies.

We will use the gt package in R to create a shift table from the baseline CTCAE grade to the maximum grade at post-baseline.

## INTRODUCTION

We will use two made-up datasets ADSL and ADLB. In the datasets there are two treatments, grp1 and grp2. Both treatments are displayed in the table as column headers that span five columns. The five columns are Grade 0 to Grade 4.

We will use the gt package in R to produce the shift table. See the reference [1] and [2] for more details about the gt package.

The datasets, R program and output in this paper are available in reference [3].

## THE DETAILS

We will need four packages in the R program:

- use the haven package to read the SAS® datasets
- use the summarise function from the package dplyr to count the distinct values of a variable
- use the pivot\_wider function from the package tidyr to transpose a dataset
- use the gt package to create the table in HTML format.

This is the R program shift.R. Note that in order to display the grade 0 to 4 at both baseline and post-baseline for each lab parameter, we need to create a frame dataset with all the possible grades and merge it with the input lab data.

```
library(haven)
library(dplyr)
library(tidyr)
library(gt)

the_date <- as.character(Sys.Date())

# read the data
adsl <- read_sas("C:\\shift\\adsl.sas7bdat")
adlb <- read_sas("C:\\shift\\adlb.sas7bdat")
```

```

# get the big N in column headers from adsl
bign <- table(group=adsl$TRT01A)

# get the post-baseline data in ADLB
post <- adlb[(adlb$ADT>adlb$TRTSDT),]

# sort post by USUBJID, PARAM and ATOXGRN
post <- post[order(post$USUBJID,post$PARAM, post$ATOXGRN),]
post$lastdot <- !duplicated(post[c("USUBJID","PARAM")],fromLast=TRUE)

# keep the most severe record for each USUBJID, PARAM
post <- post[(post$lastdot==TRUE),]

# do the counting by TRT01A, PARAM, BTOXGRN, ATOXGRN

count0 <-
  post %>%
  group_by(TRT01A, PARAM, BTOXGRN, ATOXGRN) %>%
  summarise(unique_subj = n_distinct(USUBJID))

# generate a frame data called comb
# it has all the combinations of ATOXGRN, BTOXGRN, PARAM and TRT01A

mat <- matrix(NA, nrow = 25, ncol = 4)
xy <- data.frame(mat)
xy1 <- data.frame(mat)

for (i in seq(1,25)){
  xy[[i,1]] <- data.frame(X1=i%%5)
  xy[[i,2]] <- data.frame(X2=ceiling(i/5)-1)
}

for (i in seq(1:length(unique(count0$PARAM)))){
  xy$X3 <- unique(count0$PARAM)[i]
  assign(paste0('xy', i, sep=''), xy)
}

```

```

comb <- do.call("rbind", mget(sprintf("xy%d",
1:length(unique(count0$PARAM))))))

for (i in seq(1:length(unique(count0$TRT01A)))){
  comb$X4 <- unique(count0$TRT01A)[i]
  assign(paste0('comb', i, sep=''), comb)
}

comb <- do.call("rbind", mget(sprintf("comb%d",
1:length(unique(count0$TRT01A))))))

comb$PARAM <- comb$X3
comb$TRT01A <- comb$X4
comb$BTOXGRN <- as.numeric(unlist(comb$X1))
comb$ATOXGRN <- as.numeric(unlist(comb$X2))

# merge comb with count0
m_count0 <- merge(count0, comb, by=c("TRT01A", "PARAM", "BTOXGRN",
"ATOXGRN"), all=TRUE)
m_count0$denom <- ifelse(m_count0$TRT01A=='grp1', bign[1], bign[2])
m_count0$value <- ifelse(is.na(m_count0$unique_subj), "0",
paste(m_count0$unique_subj, "(",
format(round(100*m_count0$unique_subj/m_count0$denom, 1), nsmall = 1), ")"))

# do the transpose
count1 <- m_count0[(m_count0$TRT01A=="grp1"),]
count2 <- m_count0[(m_count0$TRT01A=="grp2"),]

a1 <- count1 %>%
  pivot_wider(id_cols=c(PARAM, BTOXGRN), names_from = ATOXGRN, values_from =
value,
              names_prefix = "grade")

a2 <- count2 %>%
  pivot_wider(id_cols=c(PARAM, BTOXGRN), names_from = ATOXGRN, values_from =
value,
              names_prefix = "grad")

```

```

a1 <- a1[c("PARAM","BTOXGRN","grade0", "grade1", "grade2", "grade3", "grade4"
)]
a2 <- a2[c("grad0", "grad1", "grad2", "grad3", "grad4" )]

# combine a1 and a2

df_merge <- cbind(a1,a2)
df_merge$prefix <- 'Grade'
df_merge$BTOXGRC <- paste(df_merge$prefix, df_merge$BTOXGRN)
df <- df_merge[c("PARAM","BTOXGRC","grade0", "grade1", "grade2", "grade3",
"grade4","grad0", "grad1", "grad2", "grad3", "grad4" )]

df %>%
  gt(groupname_col="PARAM")

# use gt to do the reporting
tab_html <- df %>%
  gt(groupname_col="PARAM") %>%

  tab_header(
    title = "Table 14.3.4. Shift Table from CTCAE Grade at Baseline to the
Most Severe CTCAE Grade at Post-baseline",
    subtitle = "Safety Population"
  ) %>%

  tab_source_note(
    source_note = paste('Program Source: shift.R', the_date)
  ) %>%

  cols_label(

    BTOXGRC = html("Baseline CTCAE Grade"),
    grade0 = html("Grade 0"),
    grade1 = html("Grade 1"),

```

```

    grade2 = html("Grade 2"),
    grade3 = html("Grade 3"),
    grade4 = html("Grade 4"),
    grad0 = html("Grade 0"),
    grad1 = html("Grade 1"),
    grad2 = html("Grade 2"),
    grad3 = html("Grade 3"),
    grad4 = html("Grade 4")

) %>%

tab_options(
  table.border.top.color = "white",
  heading.border.bottom.color = "black",
  table.border.bottom.color = "white",
  table_body.border.bottom.color = "black",
  table_body.hlines.color = "white",
  row_group.border.bottom.color = "white",
  row_group.border.top.color = "white",
  column_labels.border.top.color = "black",
  column_labels.border.bottom.color = "black",
) %>%

tab_spanner(
  label = html(paste("Group 1 Most Severe Post-baseline <br> (N=", bign[1],
")")),
  columns = c(grade0, grad1, grade2, grade3, grade4)
) %>%

tab_spanner(
  label = html(paste("Group 2 Most Severe Post-baseline <br> (N=", bign[2],
")")),
  columns = c(grad0, grad1, grad2, grad3, grad4)
) %>%

cols_align(
  align = "left",
  columns = c(BTOXGRC)
)

```

```
# output the HTML table

tab_html %>%
  gtsave("shift.html", path = "C:\\\\shift" )
```

The table created with this R program is shown in display 1.

Table 14.3.4. Shift Table from CTCAE Grade at Baseline to the Most Severe CTCAE Grade at Post-baseline  
Safety Population

Baseline CTCAE Grade	Group 1 Most Severe Post-baseline (N= 25 )					Group 2 Most Severe Post-baseline (N= 25 )				
	Grade 0	Grade 1	Grade 2	Grade 3	Grade 4	Grade 0	Grade 1	Grade 2	Grade 3	Grade 4
Hemoglobin(g/L)										
Grade 0	19 ( 76.0 )	5 ( 20.0 )	0	0	0	19 ( 76.0 )	6 ( 24.0 )	0	0	0
Grade 1	0	1 ( 4.0 )	0	0	0	0	0	0	0	0
Grade 2	0	0	0	0	0	0	0	0	0	0
Grade 3	0	0	0	0	0	0	0	0	0	0
Grade 4	0	0	0	0	0	0	0	0	0	0
Leukocytes(10 <sup>9</sup> /L)										
Grade 0	16 ( 64.0 )	2 ( 8.0 )	3 ( 12.0 )	1 ( 4.0 )	0	20 ( 80.0 )	2 ( 8.0 )	2 ( 8.0 )	0	0
Grade 1	0	0	1 ( 4.0 )	1 ( 4.0 )	0	0	0	1 ( 4.0 )	0	0
Grade 2	1 ( 4.0 )	0	0	0	0	0	0	0	0	0
Grade 3	0	0	0	0	0	0	0	0	0	0
Grade 4	0	0	0	0	0	0	0	0	0	0

**Display 1. Shift Table in HTML Format, Created with gt Package in R**

## CONCLUSION

The package gt is a great tool to create tables.

## REFERENCES

- [1] Some detailed discussion about gt package, available at:  
[https://aosmith16.github.io/spring-r-topics/slides/week04\\_gt\\_tables.html#1](https://aosmith16.github.io/spring-r-topics/slides/week04_gt_tables.html#1)
- [2] Presentation by Rich Iannone, available at:  
<https://www.youtube.com/watch?v=h1KAjSfSbmk&t=872s>
- [3] the datasets, R program and output in this paper are available at:  
<https://github.com/hengweiliu2020/create-shift-table-with-gt-package>

## CONTACT INFORMATION

Hengwei Liu  
 Hengrui USA  
 400 Alexander Park

Princeton, NJ 08540

[Hengwei\\_liu@yahoo.com](mailto:Hengwei_liu@yahoo.com)

SAS and all other SAS Institute Inc. product 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.