

Create TLGs and log files by sassy

Perphy ZHAO

.sanofi

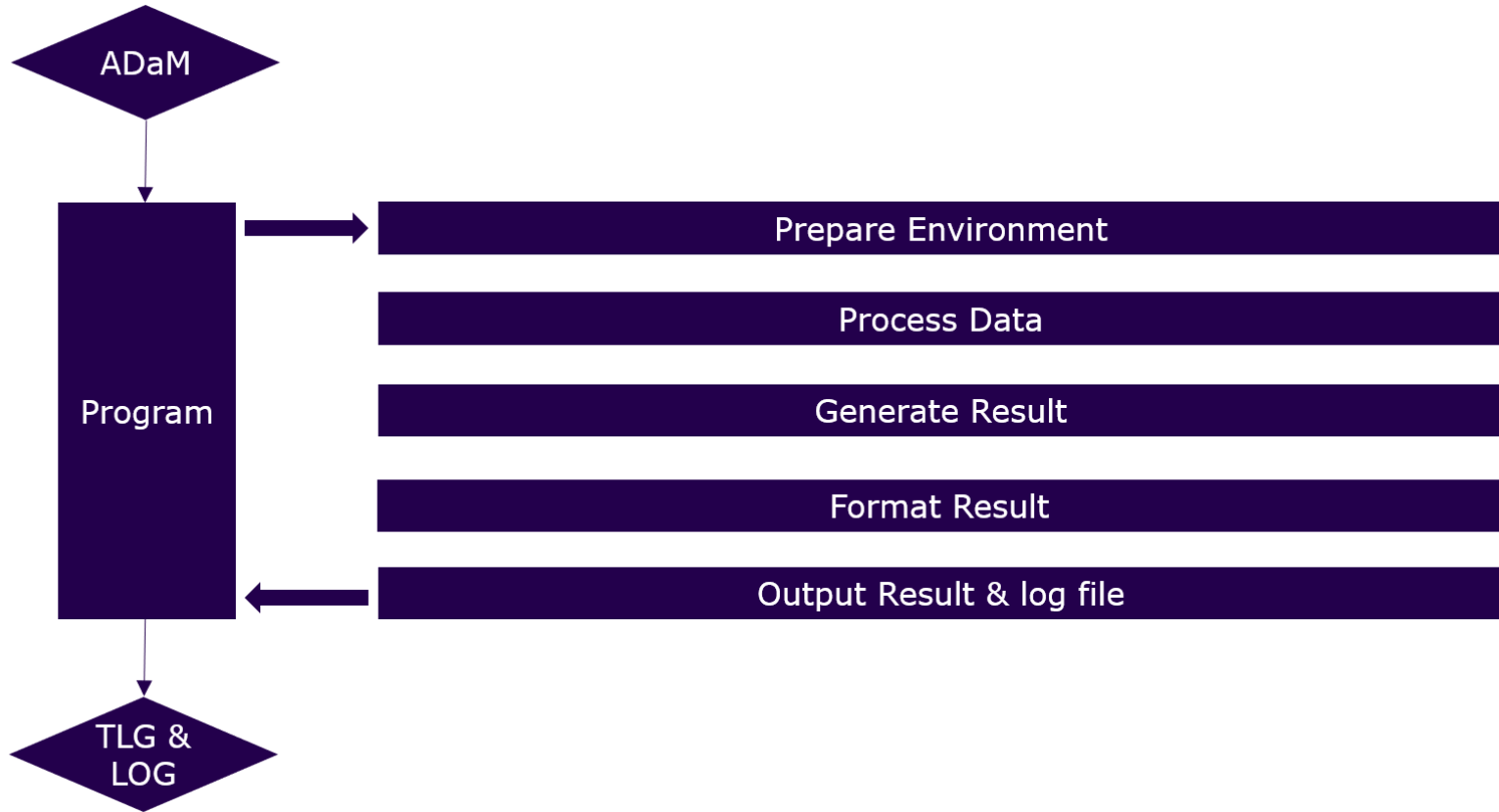
Agenda

- Introduction of sassy
- TLGs Programming & Log Output
- Functions Deep Dive
- Summary
- Q&A

Introduction of sassy

sanofi

Process Flow



Introduction of sassy

sassy package makes R easier, especially pure SAS programmers, to create TLGs and log files. This package is a meta-package brings several SAS concept to R, and the programming grammar is highly similar with SAS.

- `libr`: defines libnames, generate data dictionaries, and simulate data steps.
- `fmtr`: provides functions format data and creating format catalogs.
- `procs`: functions simulate SAS procedures and includes simulations of `FREQ`, `MEANS`, `TRANSPOSE`, `SORT` and `PRINT` procedures.
- `reporter`: report with easy layout capabilities and the ability to write reports in `RTF`, `DOCX`, `TXT` and `HTML` file formats.
- `logr`: produces a traceable log files.
- `common`: utility functions across the sassy family packages, and useful in their own right.

TLGs Programming & Log Output

sanofi

Table Programming

```
# Prepare Environment -----
```

```
library(tidyverse)
library(haven)
library(sassy)
```

```
ADD <- "STUDY_TRIPLETE/CSR/ADS/DATA" # datasets path
OUTPUT <- "STUDY_TRIPLETE/CSR/REPORT/OUTPUT" # output path
LOG <- "STUDY_TRIPLETE/CSR/REPORT/LOG" # log path
```

```
libname(ADS, ADD, "sas7bdat") # create a libname ADS
lib_load(ADS) # load data into workspace
```

```
# library 'ADS': 31 items
- attributes: sas7bdat Loaded
- path:
- items:
```

	Name	Extension	Rows	Cols	Size	LastModified
1	adae	sas7bdat	1727	161	2.4 Mb	2022-01-17 03:24:37
2	adcm	sas7bdat	17911	137	18.8 Mb	2022-01-17 03:25:17
3	addv	sas7bdat	770	100	816.9 Kb	2022-01-17 03:39:23
4	adeg	sas7bdat	10470	133	10.7 Mb	2022-01-17 03:28:15
5	ades	sas7bdat	6733	90	4.8 Mb	2022-01-17 03:27:49
6	adex	sas7bdat	9035	97	7.3 Mb	2022-01-17 03:27:09
7	adis	sas7bdat	2118	112	2 Mb	2022-01-17 03:38:41

```
# library 'ADS': 31 items
- attributes: sas7bdat not loaded
- path: /CSR/ADS/DATA
- items:
```

	Name	Extension	Rows	Cols	Size	LastModified
1	adae	sas7bdat	1727	161	2.4 Mb	2022-01-17 03:24:37
2	adcm	sas7bdat	17911	137	18.8 Mb	2022-01-17 03:25:17
3	addv	sas7bdat	770	100	816.9 Kb	2022-01-17 03:39:23
4	adeg	sas7bdat	10470	133	10.7 Mb	2022-01-17 03:28:15
5	ades	sas7bdat	6733	90	4.8 Mb	2022-01-17 03:27:49
6	adex	sas7bdat	9035	97	7.3 Mb	2022-01-17 03:27:09
7	adis	sas7bdat	2118	112	2 Mb	2022-01-17 03:38:41

The screenshot shows the RStudio interface. The Environment pane is active, displaying the ADS dataset as a 'Large lib (31 elements, 461.1 MB)'. The dataset is expanded to show a list of files with their respective observation counts and variable counts. The files listed are ADS.adae, ADS.adcm, ADS.addv, ADS.adeg, ADS.ades, ADS.adex, and ADS.adis.

Dataset	Observations	Variables
ADS.adae	1727 obs.	161 variables
ADS.adcm	17911 obs.	137 variables
ADS.addv	770 obs.	100 variables
ADS.adeg	10470 obs.	133 variables
ADS.ades	6733 obs.	90 variables
ADS.adex	9035 obs.	97 variables
ADS.adis	2118 obs.	112 variables

Table Programming

```
# Prepare data
adsl <- datastep(ADS.adsl,
  keep = c('USUBJID', 'SEX', 'AGE', 'RACE', 'SAFFL',
    find.names(ADS.adsl, pattern = 'TRTGRP*'),
    'AGEGRP', 'AGEGRPN'),
  where = expression(SAFFL == 'Y'),
  {
    if (AGE > 10)
    {
      AGEGRP = '> 10'
      AGEGRPN = 1L
    }
    else
    {
      AGEGRP = '<= 10'
      AGEGRPN = 2L
    }
  }
  # any operation can be done in {} like sas
))
```

keep option is like retain statement in SAS, can retain the variables' order

	USUBJID Unique Subject Identifier	SEX Sex	AGE Age	RACE Race	SAFFL Safety Population Flag	TRTGRP1 Treatment Classifications	TRTGRP1N Treatment Classifications (N)	AGEGRP	AGEGRPN
1	00001	M	11	WHITE	Y	Treatment	2	> 10	1
2	00002	M	13	WHITE	Y	Treatment	2	> 10	1
3	00003	M	12	WHITE	Y	Placebo	1	> 10	1
4	00004	M	9	WHITE	Y	Placebo	1	<= 10	2
5	00005	F	8	WHITE	Y	Placebo	1	<= 10	2
6	00006	F	15	WHITE	Y	Treatment	2	> 10	1
7	00007	M	8	WHITE	Y	Treatment	2	<= 10	2
8	00008	F	13	WHITE	Y	Treatment	2	> 10	1
9	00009	M	14	OTHER	Y	Placebo	1	> 10	1

Table Programming

```
# Age Summary
```

```
age_sum <- proc_means(adsl, var = AGE,  
  stats = v(n, mean, std, median, q1, q3, min, max),  
  by = TRTGRP1,  
  options = v(notype, nofreq))
```

```
# Format the stats
```

```
fc <- fcat(MEAN = '%.1f', STD = ' (%.2f)',  
  Q1 = '%.1f', Q3 = '%.1f',  
  MIN = '%d', MAX = '%d')
```

```
age_fmt <- datastep(age_sum,  
  format = fc,  
  drop = find.names(age_sum, start = 4),  
  {  
    `Mean (SD)` <- fapply2(MEAN, STD)  
    Median <- MEDIAN  
    `Q1 : Q3` <- fapply2(Q1, Q3, sep = " : ")  
    `Min : Max` <- fapply2(MIN, MAX, sep = " : ")  
  })
```

```
age_tran <- proc_transpose(age_fmt,  
  var = names(age_fmt),  
  copy = VAR, id = BY,  
  name = LABEL)
```

Packages Help Viewer Presentation

TRTGRP1=Placebo									
Variable	N	Mean	Std Dev	Median	Lower Quartile	Upper Quartile	Minimum	Maximum	
AGE	31	12.7741935	1.7834363	12.0000000	12.0000000	15.0000000	9.0000000	16.0000000	

TRTGRP1=Treatment									
Variable	N	Mean	Std Dev	Median	Lower Quartile	Upper Quartile	Minimum	Maximum	
AGE	69	12.7826087	2.0641874	13.0000000	11.0000000	14.0000000	8.0000000	17.0000000	

age_sum x

Filter

	BY	VAR	N	MEAN	STD	MEDIAN	Q1	Q3	MIN	MAX
	TRTGRP1	Variable		Mean	Std Dev	Median	Lower Quartile	Upper Quartile	Minimum	Maximum
1	Placebo	AGE	31	12.77419	1.783436	12	12	15	9	16
2	Treatment	AGE	69	12.78261	2.064187	13	11	14	8	17

age_fmt x

Filter

	BY	VAR	N	Mean (SD)	Median	Q1 : Q3	Min : Max
	TRTGRP1	Variable					
1	Placebo	AGE	31	12.8 (1.78)	12	12.0 : 15.0	9 : 16
2	Treatment	AGE	69	12.8 (2.06)	13	11.0 : 14.0	8 : 17

age_tran x

Filter

	VAR	LABEL	Placebo	Treatment
1	AGE	N	31	69
2	AGE	Mean (SD)	12.8 (1.78)	12.8 (2.06)
3	AGE	Median	12	13
4	AGE	Q1 : Q3	12.0 : 15.0	11.0 : 14.0
5	AGE	Min : Max	9 : 16	8 : 17

Table Programming

```
# Sex frequency
sex_freq <- proc_freq(ads1, tables = SEX,
                     by = TRTGRP1,
                     options = 'nonobs')

# Format the stats
sex_fmt <- value(condition(x == 'F', 'Female', order = 1L),
                 condition(x == 'M', 'Male', order = 2L))

fc <- fcat(SEX = sex_fmt,
           CNT = '%2d',
           PCT = ' (%5.2f%)')

sex_fmt <- datastep(sex_freq,
                   format = fc,
                   drop = v(CNT,PCT),
                   rename = list(CAT = 'LABEL'),
                   {
                     CNTPCT = fapply2(CNT, PCT)
                     CAT <- fapply(CAT, fc$SEX)
                   })

sex_tran <- proc_transpose(sex_fmt,
                          var = CNTPCT,
                          copy = VAR, id = BY,
                          by = LABEL)
```

Viewer Presentation

Table of TRTGRP1=Placebo, Sex

SEX	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	12	38.71	12	38.71
M	19	61.29	31	100.00

Table of TRTGRP1=Treatment, Sex

SEX	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	19	27.54	19	27.54
M	50	72.46	69	100.00

sex_freq ×

BY	VAR	CAT	CNT	PCT
		SEX	Frequency	Percent
1	Placebo	SEX	F	12 38.70968
2	Placebo	SEX	M	19 61.29032
3	Treatment	SEX	F	19 27.53623
4	Treatment	SEX	M	50 72.46377

sex_fmt ×

BY	VAR	LABEL	CNTPCT
		SEX	
1	Placebo	SEX	Female 12 (38.71%)
2	Placebo	SEX	Male 19 (61.29%)
3	Treatment	SEX	Female 19 (27.54%)
4	Treatment	SEX	Male 50 (72.46%)

sex_tran ×

VAR	LABEL	NAME	Placebo	Trtement
1	SEX	Female	CNTPCT 12 (38.71%)	19 (27.54%)
2	SEX	Male	CNTPCT 19 (61.29%)	50 (72.46%)

Table Programming

```
# Combine all block  
dem_demo_s_t <- datasteper(sex_tran,  
  set = list(race_tran, age_tran, age_tran1),  
  drop = v(NAME),  
  {})
```

	VAR	LABEL	NAME	Placebo	Treatment
1	SEX	Female	CNTPCT	12 (38.71%)	19 (27.54%)
2	SEX	Male	CNTPCT	19 (61.29%)	50 (72.46%)

	VAR	LABEL	NAME	Placebo	Treatment
1	RACE	White	CNTPCT	30 (96.77%)	65 (94.20%)
2	RACE	Black or African American	CNTPCT	0 (0.00%)	2 (2.90%)
3	RACE	Other	CNTPCT	1 (3.23%)	2 (2.90%)

	VAR	LABEL	Placebo	Treatment
1	AGE	N	31	69
2	AGE	Mean (SD)	12.8 (1.78)	12.8 (2.06)
3	AGE	Median	12	13
4	AGE	Q1 : Q3	12.0 : 15.0	11.0 : 14.0
5	AGE	Min : Max	9 : 16	8 : 17

	VAR	LABEL	NAME	Placebo	Treatment
1	AGEGRP	<= 10	CNTPCT	2 (6.45%)	7 (10.14%)
2	AGEGRP	> 10	CNTPCT	29 (93.55%)	62 (89.86%)

	VAR	LABEL	Placebo	Treatment
1	SEX	Female	12 (38.71%)	19 (27.54%)
2	SEX	Male	19 (61.29%)	50 (72.46%)
3	RACE	White	30 (96.77%)	65 (94.20%)
4	RACE	Black or African American	0 (0.00%)	2 (2.90%)
5	RACE	Other	1 (3.23%)	2 (2.90%)
6	AGE	N	31	69
7	AGE	Mean (SD)	12.8 (1.78)	12.8 (2.06)
8	AGE	Median	12	13
9	AGE	Q1 : Q3	12.0 : 15.0	11.0 : 14.0
10	AGE	Min : Max	9 : 16	8 : 17
11	AGEGRP	<= 10	2 (6.45%)	7 (10.14%)
12	AGEGRP	> 10	29 (93.55%)	62 (89.86%)

Table Programming

```
# Report
var_fmt <- c('AGE' = 'Age(years)', 'AGEGRP' = 'Age Group',
            'SEX' = 'Sex', 'RACE' = 'Race')

# Create Table
tbl <- create_table(dem_demo_s_t, first_row_blank = FALSE) %>%
  column_defaults(c('Placebo', 'Treatment'), align = 'center', width = 3.65) %>%
  stub(vars = c('VAR', 'LABEL'), 'Demographic Category', width = 4.5) %>%
  define(VAR, blank_after = TRUE, dedupe = TRUE, label = '',
        format = var_fmt, label_row = TRUE) %>%
  define(LABEL, indent = .25, label = '') %>%
  define(Placebo, label = 'Placebo', n = bigN['Placebo'],
        align = 'center') %>%
  define(Treatment, label = 'Treatment', n = bigN['Treatment'],
        align = 'center') %>%
  titles('Demographic data, data at baseline and medication details',
        'Demographics',
        'Patient characteristics at baseline - Safety population',
        bold = TRUE,
        align = 'left',
        borders = 'bottom') %>%
  footnotes('PGM=../CSR/REPORT/PGM/dem_demo_s_t.R
            OUT=OUTPUT/dem_demo_mod_s_t_x.rtf',
            borders = 'top',
            blank_row = 'none')

rpt <- create_report(file.path(QCO, "dem_demo_s_t"),
                    output_type = "RTF",
                    font = "Arial") %>%
  page_header('Sponsor: Sanofi', 'Study: XXX') %>%
  set_margins(left = 1.1, right = 0.79, top = 1.3, bottom = 0.67) %>%
  add_content(tbl) %>%
  page_footer('Date Produced: {Sys.Date()}', right = 'Page [pg] of [tpg]')

res <- write_report(rpt)
```

VAR	LABEL	Placebo	Treatment
1	SEX Female	12 (38.71%)	19 (27.54%)
2	SEX Male	19 (61.29%)	50 (72.46%)
3	RACE White	30 (96.77%)	65 (94.20%)
4	RACE Black or African American	0 (0.00%)	2 (2.90%)
5	RACE Other	1 (3.23%)	2 (2.90%)
6	AGE N	31	69
7	AGE Mean (SD)	12.8 (1.78)	12.8 (2.06)
8	AGE Median	12	13
9	AGE Q1 : Q3	12.0 : 15.0	11.0 : 14.0
10	AGE Min : Max	9 : 16	8 : 17

Sponsor: Sanofi
 Demographic data, data at baseline and medication details
 Demographics
 Patient characteristics at baseline - Safety population
 Study: XXX

Demographic Category	Placebo (N=31)	Treatment (N=69)
Sex		
Female	12 (38.71%)	19 (27.54%)
Male	19 (61.29%)	50 (72.46%)
Race		
White	30 (96.77%)	65 (94.20%)
Black or African American	0 (0.00%)	2 (2.90%)
Other	1 (3.23%)	2 (2.90%)
Age(years)		
N	31	69
Mean (SD)	12.8 (1.78)	12.8 (2.06)
Median	12	13
Q1 : Q3	12.0 : 15.0	11.0 : 14.0
Min : Max	9 : 16	8 : 17
Age Group		
<= 10	2 (6.45%)	7 (10.14%)
> 10	29 (93.55%)	62 (89.86%)

PGM=../CSR/REPORT/PGM/dem_demo_s_t.R OUT=OUTPUT/dem_demo_mod_s_t_x.rtf

Listing Programming

```
# Prepare list data set
# Create list
tbl <- create_table(final1, first_row_blank = FALSE) %>%
  column_defaults(c('TRTDEMO', 'VISIT_', 'HGB', 'HCT', 'PLAT')) %>%
  spanning_header(from = 3, to = 5,
    label = "Parameter(Unit)\n
    value/Change from baseline{supsc('a')}}") %>%
  define(TRTDEMO, blank_after = TRUE, dedupe = TRUE,
    align = 'left', label_align = 'left', width = 2.6,
    label = "Arm/Patient\n(Age,Sex{supsc('a')},
    Race{supsc('b')},Weight(kg))"
  ) %>%
  define(VISIT_, align = 'left', width = 1.6, label_align = 'left',
    label = "Measurement date{supsc('c')}/Dose administration
    on the same day") %>%
  define(HGB, align = 'centre', width = 1.6,
    label = 'Hemoglobin\n(G/L)') %>%
  define(HCT, align = 'centre', width = 1.6,
    label = 'Hematocrit') %>%
  define(PLAT, align = 'centre', width = 1.6,
    label = 'Plalets\n(GIGA/L)')
    total width is about 9 for the horizontal

rpt <- create_report(file.path(QCO, "lab_pcsa_rbc_s_1"),
  output_type = "RTF",
  font = "Times") %>%
  page_header('Sponsor: Sanofi', 'Study: XXX') %>%
  set_margins(left = 1.1, right = 0.79, top = 1.3, bottom = 0.67) %>%
  add_content(tbl) %>%
  page_footer('Date Produced: {Sys.Date()}', right = 'Page [pg] of [tpg]')

res <- write_report(rpt)
```

	TRTDEMO	VISIT_	HGB	HCT	PLAT
1	Placebo/0001(10,M,W,32)	2017-08-15/D-364/N	129/0	0.39/0	184/0
2	Placebo/0001(10,M,W,32)	2018-08-14/D1/Y	130/1	0.39/-0.004	178/-6
3	Placebo/0001(10,M,W,32)	2018-09-11/D29/Y	136/7	0.45/0.06	252/68
4	Placebo/0001(10,M,W,32)	2018-11-05/D84/Y	128/-1	0.45/0.061	223/39
5	Placebo/0001(10,M,W,32)	2019-01-30/D170/Y	133/4	0.47/0.084+	219/35

Sponsor: Sanofi
 Clinical Laboratory data
 Listings
 Red blood cells, platelets and coagulation - Listing of patients with abnormalities (PCSA) during the TEAE period - Safety population
 Study: XXX

Arm/Patient (Age,Sex,Race,Weight(kg))	Measurement date/Dose administration on the same day	Parameter(Unit) value/Change from baseline*		
		Hemoglobin (G/L)	Hematocrit	Plalets (GIGA/L)
Placebo/0001(10,M,W,32)	2017-08-15/D-364/N	129/0	0.39/0	184/0
	2018-08-14/D1/Y	130/1	0.39/-0.004	178/-6
	2018-09-11/D29/Y	136/7	0.45/0.06	252/68
	2018-11-05/D84/Y	128/-1	0.45/0.061	223/39
	2019-01-30/D170/Y	133/4	0.47/0.084+	219/35
	2019-04-24/D254/Y	128/-1	0.44/0.05	203/19
	2019-08-14/D366/N	132/3	0.45/0.061	227/43
Treatment0002(9,M,W,28)	2019-11-11/D455/N	129/0L	0.44/0.048	210/26
	2017-09-28/D-364/N	151/0	0.45/0	308/0
	2018-09-27/D1/Y	132/-19	0.39/-0.063	295/-13
	2018-10-25/D29/Y	139/-12	0.44/-0.012	322/14
	2019-01-02/D98/Y	132/-19	0.43/-0.026	326/18
	2019-03-14/D169/Y	128/-23-	0.42/-0.033	300/-8
	2019-06-06/D253/Y	130/-21-	0.42/-0.029	295/-13
	2019-09-23/D362/N	134/-17	0.43/-0.025	365/57
	2020-01-27/D488/N	133/-18	0.44/-0.014	362/54

*M=Male, F=Female
 *W=Caucasian/White, B=Black, A=Asian/Oriental, I=American Indian or Alaska Native, N=Native Hawaiian or Other Pacific Islander, O=Other
 *U=Unscheduled
 *Relative day to the first dose of XXXX
 *L: below lower limit of normal (LLN); H: above upper limit of normal (ULN). - or +: values reaching the lower or upper PCSA limit
 PGM=...CSR.REPORT/PGM/lab_pcsa_s_1R OUT=OUTPUT/lab_pcsa_rbc_ful_s_1_rtf



Figure Programming

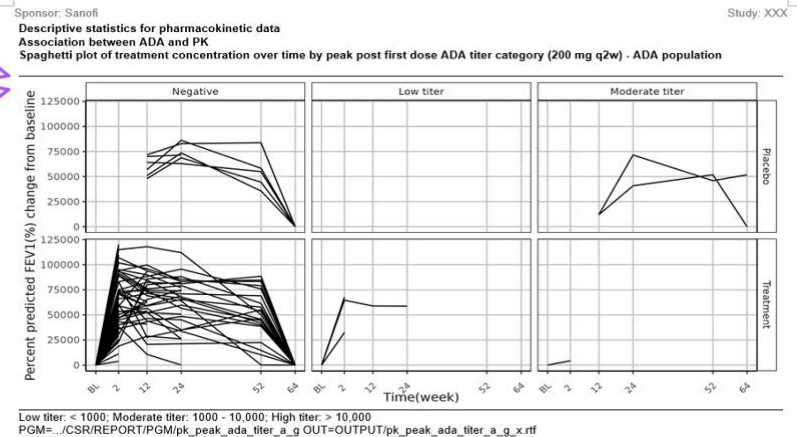
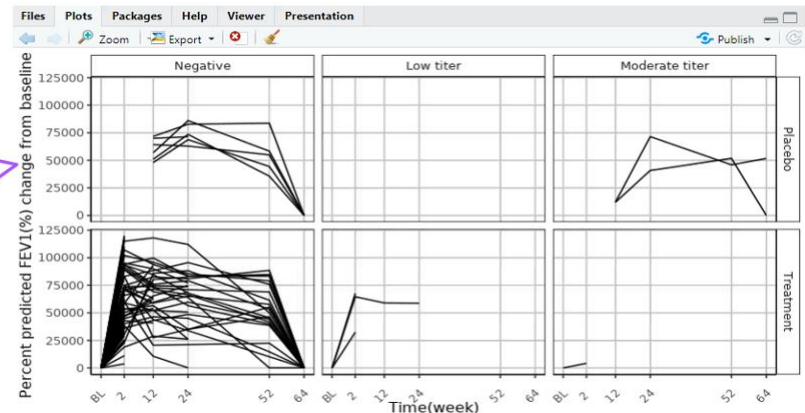
```
# Prepare data for plot
# can use tidyverse, sassy to prepare, or any other package

# Plot
# can use ggplot, survfit to plot, or any other jpg format
p_plot <- ggplot(final, aes(x = week, y = AVAL,
                           group = as.factor(USUBJID)))

# Create plot object and report
page <- create_plot(p_plot, 4.3, 9) %>%
  titles('Descriptive statistics for pharmacokinetic data',
        'Association between ADA and PK',
        'Spaghetti plot of treatment concentration over time by peak post
        first dose ADA titer category (200 mg q2w) - ADA population',
        bold = TRUE,
        align = 'left',
        borders = 'bottom') %>%
  footnotes('Low titer: < 1000; Moderate titer: 1000 - 10,000;
            High titer: > 10,000',
            'PGM=../CSR/REPORT/PGM/pk_peak_ada_titer_a_g
            OUT=OUTPUT/pk_peak_ada_titer_a_g.x.rtf',
            borders = 'top',
            blank_row = 'none')

rpt <- create_report(file.path(QCO, 'pk_peak_ada_titer_a_g'),
                    output_type = 'RTF',
                    font = 'Arial') %>%
  set_margins(left = 1.1, right = 0.79, top = 1.3, bottom = 0.67) %>%
  page_header('Sponsor: Sanofi', 'Study: XXX') %>%
  add_content(page) %>%
  page_footer(paste0('Date Produced: ', fapply(Sys.time(), '%d%b%y %H:%M')),
             right = "Page [pg] of [tpg]")

write_report(rpt)
```



Log Output

```
# Set options for log
options("logr.autolog" = TRUE,
       "logr.on" = TRUE,
       "logr.notes" = TRUE,
       "procs.print" = TRUE,
       "tidylog.display" = list(log_print))

# Open log
lgpth <- log_open(file.path(QCL, "dem_demo_s_t.log"),
                  logdir = FALSE,
                  autolog = NULL,
                  compact = FALSE,
                  traceback = TRUE)

# Input your code

# Close log
log_close()
```

log for the demographic age summary block

Note: there will be an outlook file in your log folder if your program has error(s)

- Name
- pk_ada_pre_a_t
- pk_ae_isr_a_t
- pk_ae_isr_a_t
- pk_eff_event_summary_sub_ada_a_t
- pk_fev1pp_ada_a_t
- pk_fev1pp_ada_a_t
- pk_pksumm_ada_a_t

```
dem_demo_s_t_log [X]
-----
1
2 Log Path: /QC/LOG/dem_demo_s_t_log
3 Program Path: /MISC/test/pharma R2024.R
4 Working Directory: /CSR_R
5 User Name:
6 R Version: 4.2.1 (2022-06-23)
7 Machine: w-r-ide-prod-01.pharma.aventis.com x86_64
8 Operating System: Linux 3.10.0-1160.el7.x86_64 #1 SMP Tue Aug 18 14:50:17 EDT 2020
9 Base Packages: stats graphics gRDevices utils datasets methods base Other Packages: tidylog_1.0.2
10 Log Start Time: 2024-02-17 12:10:29
11
12
13 # A user-defined format: 3 conditions
14 Name Type Expression Label Order
15 1 x U x <= 10 <= 10 1
16 2 x U x > 10 > 10 2
17
18 NOTE: Log Print Time: 2024-02-17 12:10:34
19 NOTE: Elapsed Time: 5.46428513526917 secs
20
21
22 # A user-defined format: 3 conditions
23 Name Type Expression Label Order
24 1 x U x == "F" Female 1
25 2 x U x == "M" Male 2
26
27 NOTE: Log Print Time: 2024-02-17 12:10:36
28 NOTE: Elapsed Time: 1.67262363433838 secs
29
30 # A user-defined format: 3 conditions
31 Name Type Expression Label Order
32 1 x U x == "WHITE" White 1
33 2 x U x == "BLACK OR AFRICAN AMERICAN" Black or African American 2
34 3 x U x == "OTHER" Other 3
35
36 NOTE: Log Print Time: 2024-02-17 12:10:37
37 NOTE: Elapsed Time: 0.880235910415649 secs
38
39 Compile format catalog
40
41 NOTE: Log Print Time: 2024-02-17 12:10:38
42 NOTE: Elapsed Time: 1.1965868473053 secs
43
44 # A format catalog: 11 formats
45 - $AGEGRP: type U, 2 conditions
46 - $SEX: type U, 2 conditions
47 - $RACE: type U, 3 conditions
48 - $MEAN: type S, "%.1f"
49 - $STD: type S, "%.2f"
50 - $Q1: type S, "%.1f"
51 - $Q3: type S, "%.1f"
52 - $MIN: type S, "%d"
53 - $MAX: type S, "%d"
54 - $CNT: type S, "%2d"
55 - $PCT: type S, "%6.1f%%"
56
57 NOTE: Log Print Time: 2024-02-17 12:10:39
58 NOTE: Elapsed Time: 1.062155900831604 secs
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2
```

Functions Deep Dive

sanofi

Functions in libr

- `datastep`: Steps through data row-by-row, similar with SAS data step.
- `delete`: Removes records from a datastep.
- `dictionary`: Creates a data dictionary, like SAS contents procedure.
- `dsarray`: Creates a data step array.
- `dsattr`: Assigns datastep variable attributes.
- `libname`: Creates a data library, like libname statment in SAS.
- `lib_add`: Adds data to a data library.
- `lib_copy`: Copies a data library.
- `lib_delete`: Deletes a data library.
- `lib_export`: Exports a data library.
- `lib_load`: Loads a library into the workspace.
- `lib_remove`: Removes data from a data library.
- `lib_unload`: Unloads a library from the workspace.
- `lib_path`: Gets the path for a data library.
- `output`: Outputs records from a datastep.

Call functions in libr

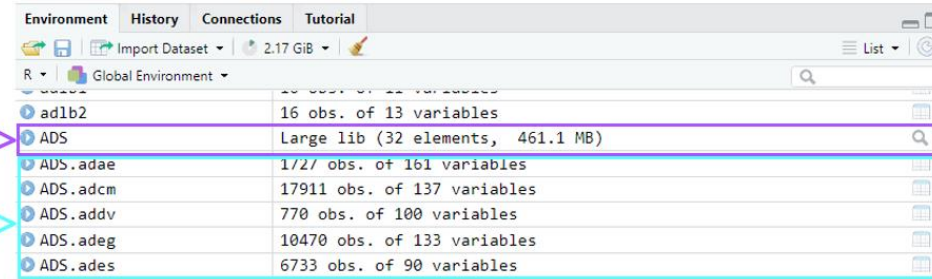
```
# Create a data library ADS  
libname(ADS, MADD, "sas7bdat")
```

```
# Load the library into the workspace  
lib_load(ADS)
```

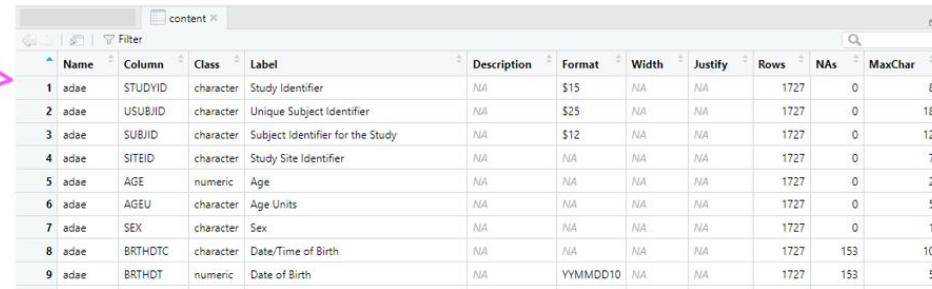
```
# Create a data dictionary, like SAS contents procedure  
content <- dictionary(ADS)
```

```
# Add data final1 to the library  
lib_add(ADS, final1)
```

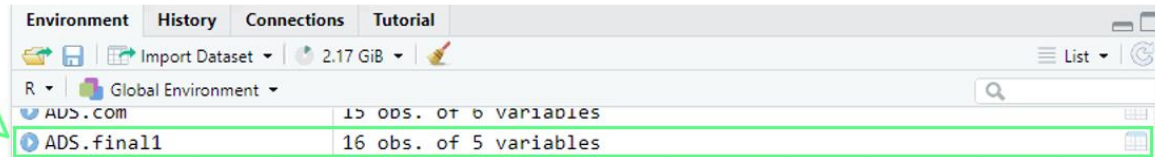
```
# Remove data final1 to the library  
lib_remove(ADS, final1)
```



Library	Observations	Variables
ad1b2	16 obs.	of 13 variables
ADS	Large lib (32 elements,	461.1 MB)
ADS.adae	1727 obs.	of 161 variables
ADS.adcm	17911 obs.	of 137 variables
ADS.addv	770 obs.	of 100 variables
ADS.adeq	10470 obs.	of 133 variables
ADS.ades	6733 obs.	of 90 variables



Name	Column	Class	Label	Description	Format	Width	Justify	Rows	NAs	MaxChar
1 adae	STUDYID	character	Study Identifier	NA	\$15	NA	NA	1727	0	8
2 adae	USUBJID	character	Unique Subject Identifier	NA	\$25	NA	NA	1727	0	18
3 adae	SUBJID	character	Subject Identifier for the Study	NA	\$12	NA	NA	1727	0	12
4 adae	SITEID	character	Study Site Identifier	NA	NA	NA	NA	1727	0	7
5 adae	AGE	numeric	Age	NA	NA	NA	NA	1727	0	2
6 adae	AGEU	character	Age Units	NA	NA	NA	NA	1727	0	5
7 adae	SEX	character	Sex	NA	NA	NA	NA	1727	0	1
8 adae	BRTHDTC	character	Date/Time of Birth	NA	NA	NA	NA	1727	153	10
9 adae	BRTHDT	numeric	Date of Birth	NA	YMMDD10	NA	NA	1727	153	5



Library	Observations	Variables
AUS.COM	15 obs.	of 6 variables
ADS.final1	16 obs.	of 5 variables

datastep() function

- data: The data to step through.
- steps: The operations to perform on the data.
- keep: A vector of quoted variables names to keep in the output dataset, like keep statement in SAS.
- drop: A vector of quoted variables names to drop in the output dataset, like keep statement in SAS.
- rename: A named vector of quoted variables to rename, like rename statement in SAS.
- by: A vector of quoted variables to use for by-group processing, which will active the first. and last., automatic variables.
- calculate: Steps to set up calculated variables.
- retain: A list of variable names and initial values to retain, like retain statement in SAS.
- attrib: A named list of attributes, like attr statement in SAS.
- arrays: A named list of dsarry objects, like array statement in SAS.
- sort_check: Checks to see if the input data is sorted according to the by variable parameter.
- format: A named list of formats to assign to the output data frame, like format statement in SAS.
- label: A named list of labels to assign to the output data frame, like label statement in SAS.
- where: An expression to filter the output dataset.
- set: A dataset or list of datasets to append to the input data frame, like set statement in SAS.
- merge: A dataset or list of datasets to merge to the input data frame, like merge statement in SAS.
- merge_by: Be used to identify the variable(s) to merge by if merge the merge parameter us used.
- merge_in: A vector of column names to be used to hold the merge flags, like in option in SAS.
- log: Whether or not to log the datastep.

datastep()-Example

```
# Set two data sets, like set statement in SAS  
adsl_set <- datastep(data = adsl_F,  
  set = adsl_M,  
  {  
    AGE1 = AGE + 2  
  })
```

	USUBJID	SEX	AGE	RACE	SAFFL	TRTGRP1	TRTGRP1N
1	00016	F	13	WHITE	Y	Treatment	2

	USUBJID	SEX	AGE	RACE	SAFFL	TRTGRP1	TRTGRP1N
1	00014	M	13	WHITE	Y	Treatment	2
2	00047	M	13	WHITE	Y	Placebo	1
3	00065	M	13	WHITE	Y	Placebo	1
4	00089	M	13	WHITE	Y	Treatment	2

AGE1 = AGE + 2

	USUBJID	SEX	AGE	RACE	SAFFL	TRTGRP1	TRTGRP1N	AGE1
1	00016	F	13	WHITE	Y	Treatment	2	15
2	00014	M	13	WHITE	Y	Treatment	2	15
3	00047	M	13	WHITE	Y	Placebo	1	15
4	00065	M	13	WHITE	Y	Placebo	1	15
5	00089	M	13	WHITE	Y	Treatment	2	15

datastep()-Example

```
# Inner join
adsl_merge_inner <- datastep(data = adsl_arm,
  merge = adsl_age,
  merge_by = 'USUBJID',
  merge_in = c('ina', 'inb'),
  where = expression(ina & inb),
  {})

# Left join
adsl_merge_left <- datastep(data = adsl_arm,
  merge = adsl_age,
  merge_by = 'USUBJID',
  merge_in = c('ina', 'inb'),
  where = expression(ina),
  {})

# Right join
adsl_merge_right <- datastep(data = adsl_arm,
  merge = adsl_age,
  merge_by = 'USUBJID',
  merge_in = c('ina', 'inb'),
  where = expression(inb),
  {})

# Full join
adsl_merge_full <- datastep(data = adsl_arm,
  merge = adsl_age,
  merge_by = 'USUBJID',
  merge_in = c('ina', 'inb'),
  where = expression(ina | inb),
  {})
```

USUBJID	TRTGRP1
1 00016	Treatment
2 00014	Treatment
3 00047	Placebo
4 00065	Placebo
5 00089	Treatment

USUBJID	AGE
1 00016	13
2 00047	13

USUBJID	TRTGRP1	AGE	ina	inb
1 00016	Treatment	13	1	1
2 00047	Placebo	13	1	1

USUBJID	TRTGRP1	AGE	ina	inb
1 00016	Treatment	13	1	1
2 00047	Placebo	13	1	1
3 00014	Treatment	NA	1	0
4 00065	Placebo	NA	1	0
5 00089	Treatment	NA	1	0

datastep()-Example

```
# Sort data set with proc_sort function in procs packages
adsl_sort <- proc_sort(data = adsl_set,
  by = v(TRTGRP1N, TRTGRP1),
  keep = v(USUBJID, TRTGRP1N, TRTGRP1, AAGE))
```

USUBJID	TRTGRP1N	TRTGRP1	AAGE
1 00047	1	Placebo	16
2 00065	1	Placebo	17
3 00016	2	Treatment	14
4 00014	2	Treatment	15
5 00089	2	Treatment	18

```
# Retain & calculate options
adsl_retain <- datastep(data = adsl_sort,
  by = c('TRTGRP1N', 'TRTGRP1'),
  retain = list(obs_num = 0, obs_num1 = 0),
  calculate = { mean_aage = mean(AAGE) }
  {
    if (first.)
      obs_num = 1
    else
      obs_num = obs_num+1

    if (first.)
      obs_num1 = 0
    else
      obs_num1 = obs_num1+1
  })
```

```
if (first.)
  obs_num = 1
else
  obs_num = obs_num+1
```

```
if (first.)
  obs_num1 = 0
else
  obs_num1 = obs_num1+1
```

Note: calculate option does not active by option

USUBJID	TRTGRP1N	TRTGRP1	AAGE	mean_aage	obs_num1	obs_num
1 00047	1	Placebo	16	16	1	1
2 00065	1	Placebo	17	16	2	2
3 00016	2	Treatment	14	16	1	1
4 00014	2	Treatment	15	16	2	2
5 00089	2	Treatment	18	16	3	3

Deep Dive in procs

- `proc_freq`: Generates frequency statistics.
- `proc_means`: Calculates summary statistics.
- `proc_print`: Prints a dataset.
- `proc_sort`: Sorts a dataset.
- `proc_transpose`: Transposes a dataset.
- `proc_ttest`: Calculates T-test statistics.

proc_freq() in procs

- data: The input dataframe to perform frequency calculations on.
- tables: The variables to perform frequency counts on.
- output: Whether or not to return datasets from function, valid values are 'out', 'none' and 'report', default is 'out', also accepts data shaping keywords 'long', 'stacked', 'wide'.
- by: An optional by group, accepts a vector of one or more variable names.
- weight: An optional weight parameter, which is passed as a variable name to use for the weight.
- options: The options desired for the function, which are passed to the parameter as a vector of quoted strings, can also use the v() function to pass unquoted strings. The following options are available: 'chiq', 'crosstab', 'fisher', 'list', 'missing', 'nlevels', 'ncol', 'nocum', 'nofreq', 'nopercen', 'noprint', 'noobs', 'norow', 'nosparse', 'notable', 'outcum'.

proc_freq()-Example

```
bigN_long<- proc_freq(adsl, tables = TRTGRP1,  
  output = long,  
  options = v(nopercent, nonobs)  
)
```

```
bigN_wide<- proc_freq(adsl, tables = TRTGRP1,  
  output = wide,  
  options = v(nopercent, nonobs))
```

```
bigN_stack<- proc_freq(adsl, tables = TRTGRP1,  
  output = stacked,  
  options = v(nopercent, nonobs))
```

	VAR	STAT	Placebo	Treatment
1	TRTGRP1	CNT	31	69

	VAR	CAT	CNT
		TRTGRP1	Frequency
1	TRTGRP1	Placebo	31
2	TRTGRP1	Treatment	69

	VAR	CAT	STAT	VALUES
1	TRTGRP1	Placebo	CNT	31
2	TRTGRP1	Treatment	CNT	69

proc_means() in procs

- data: The input dataframe to perform summary statistics.
- var: The variable(s) to perform summary statistics for.
- output: Same as proc_freq.
- by: Same as proc_freq.
- class: The class parameter is similar to the by parameter, but the output is different. By groups will create completely separate tables, while class groups will be continued in the same table.
- stats: A vector of summary statistics keywords. Valid keywords are: "css", "clm", "cv", "kurt", "kurtosis", "lclm", "mean", "median", "mode", "min", "max", "n", "nmiss", "nobs", "p1", "p5", "p10", "p20", "p25", "p30", "p40", "p50", "p60", "p70", "p75", "p80", "p90", "p95", "p99", "q1", "q3", "qrange", "range", "skew", "skewness", "std", "stddev", "stderr", "sum", "uclm", "uss", and "vari". For hypothesis testing, the function supports "t", "prt", "probt", and "df". Default statistics are: "n", "mean", "std", "min", and "max".
- options: A vector of optional keywords. Valid values are: "alpha =", "completetypes", "maxdec =", "noprint", "notype", "nofreq", "nonobs", "nway". The "notype", "nofreq", and "nonobs" keywords will turn off columns on the output datasets. The "alpha =" option will set the alpha value for confidence limit statistics. The default is 95% (alpha = 0.05). The "maxdec =" option sets the maximum number of decimal places displayed on report output. The "nway" option returns only the highest type values.

proc_sort() in procs

- data: The input data to sort.
- by: A vector of variables to sort by.
- keep: A vector of variables on the output data to keep. All other variables will be dropped.
- order: The sort order of the variables on the by parameter. Valid values are 'ascending' or 'descending'. These values may also be abbreviated to 'asc', 'desc', 'a', or 'd'.
- options: Any options desired for the sort. Available options are 'dupkey' and 'nodupkey'. The 'nodupkey' option removes duplicate rows from the sorted dataset. The 'dupkey' option removes unique rows from the sorted dataset.
- as.character: If TRUE, will cast any factors in the 'by' parameter to character. Default is FALSE. This parameter is included because it is common to use factors for sorting in R, but you may not want to keep the variable as a factor. This parameter therefore allows you to use the factor for the sort, but then convert back to a character once the sort is complete.

proc_transpose() in procs

- data: The input data to transpose.
- by: An optional by group. Parameter accepts a vector of one or more quoted variable names. If the by group is requested, the data will be subset by that variable and the transpose function will transpose each group and stack them together in a single table.
- var: The variable(s) to transpose. Parameter accepts a vector of variable names. By default, all numeric variables will be transposed.
- id: The variable or variables to use for the transposed column names.
- idlabel: The variable to use for the transposed column labels.
- copy: A vector of variables to retain in the output data without transposition. Values will be truncated or recycled to fit the number of output rows.
- name: Specifies the name of the variable to be used for the var values.
- namelabel: The label to use for the name variable.
- prefix: Contains a prefix to be used in the construction of column names.
- delimiter: Specifies a delimiter to be used in the construction of column names.
- suffix: Contains a suffix to be used in the construction of column names.
- where: An expression to filter the rows after the transform is complete. Use the expression function to define the where clause.
- options: Optional keywords that affect the transpose. Default is NULL. Available option is "noname" which drops the name column from the output dataset.
- log: Whether or not to log the procedure. Default is TRUE. This parameter is used internally

proc_transpose()-Example

```

race_tran <- proc_transpose(race_fmt,
                           var = CNTPCT,
                           copy = VAR, id = BY,
                           by = v(CATN,CAT)) %>%
proc_sort(by=v(CATN,CAT))

race_tran1 <- proc_transpose(race_fmt,
                             var = CNTPCT,
                             # copy = VAR,
                             id = BY,
                             by = v(VAR, CATN, CAT)) %>%
proc_sort(by=v(CATN,CAT))

```

Variables in copy parameter can be passed to parameter

	BY	VAR	CAT RACE	CNTPCT	CATN
1	Placebo	RACE	Black or African American	0 (0.00%)	2
2	Placebo	RACE	Other	1 (3.23%)	3
3	Placebo	RACE	White	30 (96.77%)	1
4	Treatment	RACE	Black or African American	2 (2.90%)	2
5	Treatment	RACE	Other	2 (2.90%)	3
6	Treatment	RACE	White	65 (94.20%)	1



	VAR	CAT	CATN	NAME	Placebo	Treatment
1	RACE	White	1	CNTPCT	30 (96.77%)	65 (94.20%)
2	RACE	Black or African American	2	CNTPCT	0 (0.00%)	2 (2.90%)
3	RACE	Other	3	CNTPCT	1 (3.23%)	2 (2.90%)

	VAR	CATN	CAT	NAME	Placebo	Treatment
1	RACE	1	White	CNTPCT	30 (96.77%)	65 (94.20%)
2	RACE	2	Black or African American	CNTPCT	0 (0.00%)	2 (2.90%)
3	RACE	3	Other	CNTPCT	1 (3.23%)	2 (2.90%)

Deep Dive in logr

- `log_close`: Closes the log.
- `log_code`: A `TRUE` or `FALSE` value to indicate success or failure of the function.
- `log_open`: Initializes the log file.
- `log_path`: Gets the path to the currently opened log, which takes no parameters.
- `log_print`: Prints an object to the currently opened log.
- `log_status`: gets the status of the log. Possible status values are `'on'`, `'off'`, `'open'`, or `'closed'`, and which takes no parameters.

Deep Dive in fmtr

- `condition`: Creates a condition for a user-defined format, which used in conjunction with the value function.
- `descriptions`: Gets or sets descriptions for data frame columns
- `fapply`: Applies formatting to a vector.
- `fapply2`: Applies formatting to two different vectors, and combines them into a single vector.
- `fattr`: Assigns formatting attributes to a vector.
- `fact`: Creates a format catalog, which is a collection of formats.
- `fdata`: Applies formatting attributes to the entire data frame.
- `fmt_cnt_pct`, `fmt_mean_sd`, `fmt_median`, `fmt_n`, `fmt_quantile_range`, `fmt_range`: A family functions to calculate and format a count and percent, mean and standard deviation, median, numeric count, quantile range, numeric range.
- `formats`: Gets or sets formats for a data frame.
- `value`: Creates a user-defined format.
- `width`: Gets or sets column widths for a data frame.

fmt family-Example

```
# Format family function
fmt_family <- proc_sort(data = adsl,
  by = v(TRTGRP1N, TRTGRP1)) %>%
  datastep(by = v(TRTGRP1N, TRTGRP1),
    calculate = { age_mean_sd = fmt_mean_sd(AGE)
                  age_n = fmt_n(AGE)
                  age_median = fmt_median(AGE)
                  age_quantile = fmt_quantile_range(AGE)
                  age_range = fmt_range(AGE)},
    drop = c('SEX', 'AGEGRP', 'AGEGRPN', 'SAFFL',
             'RACE', 'SEX_FMT', 'RACE_FMT'),
            { VAR = 'AGE'
            }) %>%
  proc_sort(by = v(VAR),
    keep = v(VAR, age_n, age_mean_sd, age_median,
             age_quantile, age_range),
    options = 'nodupkey')
```

USUBJID	AGE	TRTGRP1	TRTGRP1N	age_range	age_quantile	age_median	age_n	age_mean_sd	VAR
1 00003	21	Placebo		11 - 25	16.0 - 20.0	18.0	100	17.8 (3.0)	AGE
2 00004	22	Placebo		11 - 25	16.0 - 20.0	18.0	100	17.8 (3.0)	AGE
3 00005	21	Placebo		11 - 25	16.0 - 20.0	18.0	100	17.8 (3.0)	AGE
4 00009	21	Placebo		11 - 25	16.0 - 20.0	18.0	100	17.8 (3.0)	AGE
5 00010	19	Placebo		11 - 25	16.0 - 20.0	18.0	100	17.8 (3.0)	AGE

VAR	age_n	age_mean_sd	age_median	age_quantile	age_range
1 AGE	100	17.8 (3.0)	18.0	16.0 - 20.0	11 - 25

Only keep the unique records according by option

formats()-Example

```
# formats function
```

```
# Subset dataset
```

```
mtcars1 <- mtcars
mtcars1$name <- rownames(mtcars)
mtcars_subset <- datastep(data = mtcars1,
  keep = c('name', 'mpg', 'cyl'),
  steps = {
    if (n. <= 5)
      output()
  })
```

```
mtcars_subset
```

```
# Assign formats
```

```
attr(mtcars_subset$mpg, "format") <- value(condition(x >= 20, "High"),
  condition(x < 20, "Low"))
```

```
attr(mtcars_subset$cyl, "format") <- function(x) format(x, nsmall = 1)
```

```
# Display formatted data
```

```
fdata(mtcars_subset)
```

```
lst <- formats(mtcars_subset)
```

```
lst
```

```
# Alter format list and reassign
```

```
lst$mpg <- value(condition(x >= 22, "High"),
  condition(x < 22, "Low"))
```

```
lst$cyl <- function(x) format(x, nsmall = 2)
```

```
formats(mtcars_subset) <- lst
```

```
# Display formatted data
```

```
fdata(mtcars_subset)
```

```
# Clear formats
```

```
formats(mtcars_subset) <- NULL
```

```
# Confirm formats are cleared
```

```
formats(mtcars_subset)
```

```
> # Clear formats
> formats(mtcars_subset) <- NULL
> # Confirm formats are cleared
> formats(mtcars_subset)
list()
```

```
> mtcars_subset
```

	name	mpg	cyl
1	Mazda RX4	21.0	6
2	Mazda RX4 Wag	21.0	6
3	Datsun 710	22.8	4
4	Hornet 4 Drive	21.4	6
5	Hornet Sportabout	18.7	8

```
> fdata(mtcars_subset)
```

	name	mpg	cyl
1	Mazda RX4	High	6.0
2	Mazda RX4 Wag	High	6.0
3	Datsun 710	High	4.0
4	Hornet 4 Drive	High	6.0
5	Hornet Sportabout	Low	8.0

```
> lst
```

```
$mpg
```

```
# A user-defined format: 2 conditions
```

Name	Type	Expression	Label	Order	
1	x	U	x >= 20	High	NA
2	x	U	x < 20	Low	NA

```
$cyl
```

```
function(x) format(x, nsmall = 1)
```

```
> fdata(mtcars_subset)
```

	name	mpg	cyl
1	Mazda RX4	Low	6.00
2	Mazda RX4 Wag	Low	6.00
3	Datsun 710	High	4.00
4	Hornet 4 Drive	Low	6.00
5	Hornet Sportabout	Low	8.00

Deep Dive in reporter

- `add_content`: Adds an object to the report content list. A report will accept multiple pieces of content.
- `add_style`: Adds a style object to a report specification.
- `cell_style`: A class to define the style for a cell in a table.
- `column_defaults`: A function to set default attributes for columns on a table, like column statement in SAS.
- `create_plot`: Create a plot specification that can be added as content to a report. The function supports plot objects returned by `ggplot` or `ggsurvplot`. It does not support the Base R plot function.
- `create_report`: Creates a report shell to which you may add titles, footnotes, content, etc
- `create_style`: Creates a style object to control background colors and font settings on your report. The style object can be applied to a report using the `add_style()` function. Currently, styles may only be applied to HTML reports.
- `create_table`: The `create_table` function creates a table object to which further specifications can be added. The object can be added to a report using the `add_content` function.
- `define`: A function to define a table column.
- `footnotes`: Adds one or more footnotes to the report.
- `lowercase_parens`: These functions are used to format the "N=" population label on column headers.
- `page_by`: Adds a page by variable to a report, table, or plot, which will generate a page break for each value of the page by variable.

Deep Dive in reporter

- `page_footer`: Adds a page footer to the report. The page footer will appear on each page of the report, at the bottom of the page.
- `page_header`: Adds a page header to the report. The page header will appear at the top of each page of the report.
- `set_margins`: Sets the page margins for the report. The units for this parameter can be inches or centimeters, depending on the units of measure specified on the `create_report` function.
- `spanning_header`: Creates a header that spans multiple columns. Spanning headers are used to group related columns.
- `stub`: Combines columns into a nested report stub.
- `titles`: Adds one or more titles to an object as a title block.
- `write_report`: This function writes a `report_spec` object to the file system, using the specifications provided in the object.

Summary

sanofi

References

- [sassy: vignettes/sassy-dm.Rmd \(rdrr.io\)](#)
- [CRAN - Package sassy \(r-project.org\)](#)
- [Application of R language in clinical data \(lexjansen.com\)](#)

Q&A

sanofi

•
Thank you
•

sanofi