

Generating TFLs in R - Challenges and Successes compared to SAS

CT05

Amol Waykar (d-wise)

Kevin Kramer (Experis)

Kalyani Komarasetti (Experis)

Andrew Miskell (Eli Lilly and Company)

US PhUSE Connect

March 8th-11th, 2020

Orlando, Florida



Rationale and Background of TFL Creation

- ◆ TFLs have traditionally been created using SAS
- ◆ R is freeware which does statistics and graphics well
- ◆ Need to determine how TFLs will be different in content, aesthetics, and support framework

Version and Consistency

SAS	R
Company owned	Freeware
Professional support team and structure	Support team and structure has day jobs
Almost 100% backward compatible updates	Updates not always backward compatible
Version updates in months or years	Version updates constantly
Slow to add/change functionality	Quick to add/change functionality
Robust validation techniques	Various validation techniques

Sorting

- ◆ Character variables: missing values sorted first in SAS and R
- ◆ Numeric variables: missing values sorted first in SAS. Missing values sorted last in R

SAS	<pre>proc sort data=anl_sort; by numvar; run;</pre>
R	<pre>anl_sort <- anl_sort[order(anl_sort\$numvar),]</pre>

SAS		R
numvar		
.		1
1		2
2		3
3		4
4		5
5		NA

Rounding

At the midpoint (5):

- SAS rounds away from 0
- R rounds to the even number

SAS	Integer	<code>sas_roundvar=round(unroundedvar,1);</code>
	Tenths	<code>roundvar=round(unroundedvar,.1);</code>
R	Integer	<code>dataframe\$r_roundvar <- round(dataframe\$unroundedvar,digits=0)</code>
	Tenths	<code>dataframe\$r_roundvar <- round(dataframe\$unroundedvar,digits=1)</code>

Raw Value (unroundedvar)	Rounding Precision	SAS Rounded Value (PROC SORT)	R Rounded Value (round function)
-2.5	integer	-3	-2
-1.5		-2	-2
-.5		-1	0
.5		1	0
1.5		2	2
2.5		3	2
8.75	tenths	8.8	8.8
8.85		8.9	8.8
8.95		9.0	9.0
9.05		9.1	9.0
9.15		9.2	9.2
9.25		9.3	9.2

p-values (Chi-Squared)

- ◆ Chi-Squared test produces identical results between SAS and R

SAS	R
<pre>DATA test_case; INPUT treatment \$ count indicator; DATALINES; Trt_A 0 1 Trt_B 0 1 Trt_A 200 0 Trt_B 250 0 ; PROC FREQ DATA=test_case; WEIGHT count; TABLES treatment*indicator/ chisq; ods output ChiSq=chisq; run;</pre>	<pre>test_case <- matrix(c(1,1,200,250), ncol = 2) pval_chisq <- suppressWarnings(chisq.test(test_case, correct = FALSE)) pval_chisq</pre>

- ◆ SAS and R p-value=0.8746

p-values (Fisher's Exact)

◆ Fisher's Exact Test Formula:

$$p = ((a + b)! (c + d)! (a + c)! (b + d)!) / a! b! c! d! N!$$

- ◆ When inputs to the 2 X 2 contingency table both contain 0 values the output p-value from SAS and R produce different results.
- ◆ In this scenario: SAS returns a null value while R returns 1.
- ◆ Per formula, R is correct.
Per convention, SAS is accepted.
- ◆ The languages compute the same p-value in every other scenario.

SAS	R
<pre>PROC FREQ DATA=test_case; WEIGHT count; TABLES treatment*indicator; Exact fisher / maxtime = 10; ods output FishersExact = Fishers; RUN;</pre>	<pre>pval <- fisher.test(test_case)</pre>

Output Types and Destinations

◆ Plots and Vector Graphs

SAS: various procs (GPLOT, SGxxx, TEMPLATE, etc) – produce a PDF vector graph

R: packages (ggplot2, gtable, grid, etc) – produce an EMF vector graph embedded in a DOCX file

Output Types and Destinations

◆ Customized Tables and Listings

SAS: PROC REPORT

R: officer and flextable packages - less straightforward

◆ Challenges in R:

- ◆ Issue: Sizing the page and counting rows so titles and footnotes appear properly
Solution: Built inhouse function to do this in a standard way across TFLs.
- ◆ Issue: Report creation slowed down dramatically as number of pages increases when using flextable and officer in DOCX.
Solution: Built inhouse function to produce each page as a separate DOCX file and then combine into a single DOCX file at the end.

ANCOVA (with and without Repeated Measures)

- ◆ In R we can calculate below statistics in numerous ways for the box-and-whisker plots
 - Least Squares Mean (LSM) change estimate
 - LSM change Standard Error (SE) and
 - p-values (Probt Values)
- ◆ One of the main challenges –
 - To get the results from SAS and R to match 100%
 - Specially, when different covariance structures are used

ANCOVA (with and without Repeated Measures)

◆ NO REPEATED MEASURE :

- Resulted in 100% match for the LSM and p-values between SAS (with PROC MIXED and ODS) and R (with lm and emmeans functions)

◆ WITH REPEATED MEASURE:

- Not always 100% in every instance with SAS
- Matched to certain decimal places depending on the selected Covariate Matrices/Structure

ANCOVA without Repeated Measures

◆ CASE-1: No Repeated Measure (Calculating LSM_CHG, LSM_CHG_SE and P_VALUE)

SAS:

```
proc mixed data=outloc.box_mmr_test;  
  class trtn(ref="100") visitcn usubjid;  
  by paramcd param;  
  model chg=base visitcn trtn trtn*visitcn;  
  lsmeans trtn/diff ;  
  lsmeans trtn*visitcn/diff slice=visitcn;  
run;
```

R:

```
# USING NO REPEATED MEASURE - LM FUNCTION  
model_chg_m = lm(CHG ~ BASE + VISITCN_F + TRTN_F + TRTN_F*VISITCN_F, data =  
sas_ds1)  
  
lsm_calc_lst_chg_m <- emmeans(model_chg_m, revpairwise ~ TRTN_F*VISITCN_F,  
adjust="none")
```

ANCOVA without Repeated Measures

◆ CASE-1: No Repeated Measure (Calculating LSM_CHG, LSM_CHG_SE and P_VALUE)

R					SAS				
USING NO REPEATED MEASURE					USING NO REPEATED MEASURE				
TRTN_F	VISITCN_F	TRTN_F	VISITCN_F	P_VALUE	trtn	_visitcn	_trtn	visitcn	Probt
200	1	100	1	0.00000000	200	1	100	1	0.00000000
200	2	100	2	0.00000000	200	2	100	2	0.00000000
200	3	100	3	0.00000000	200	3	100	3	0.00000000
300	1	100	1	0.11510889	300	1	100	1	0.11510889
300	2	100	2	0.78961278	300	2	100	2	0.78961278
300	3	100	3	0.27338961	300	3	100	3	0.27338961
400	1	100	1	0.03594974	400	1	100	1	0.03594974
400	2	100	2	0.01002154	400	2	100	2	0.01002154
400	3	100	3	0.08186870	400	3	100	3	0.08186870
TRTN_F VISITCN_F LSM_CHG LSM_CHG_SE					trtn visitcn Estimate StdErr				
100	1	23.095927	1.553357		100	1	23.095927	1.5533569	
100	2	23.607709	1.553357		100	2	23.607709	1.5533569	
100	3	22.003855	1.553357		100	3	22.003855	1.5533569	
200	1	6.049322	1.530213		200	1	6.0493218	1.5302131	
200	2	4.725101	1.530213		200	2	4.7251005	1.5302131	
200	3	6.830730	1.530213		200	3	6.8307301	1.5302131	
300	1	26.938881	1.870359		300	1	26.938881	1.8703592	
300	2	24.257093	1.870359		300	2	24.257093	1.8703592	
300	3	24.672254	1.870359		300	3	24.672254	1.8703592	
400	1	28.150635	1.829541		400	1	28.150635	1.8295411	
400	2	29.825481	1.829541		400	2	29.825481	1.8295411	
400	3	26.191883	1.829541		400	3	26.191883	1.8295411	

ANCOVA with Repeated Measures

◆ CASE-2: With Repeated Measures (Calculating LSM_CHG, LSM_CHG_SE and P_VALUE)

SAS:

```
proc mixed data=outloc.box_mmr_test;  
  class trtn(ref="100") visitcn usubjid;  
  by paramcd param;  
  model chg=base visitcn trtn trtn*visitcn;  
  repeated visitcn/subject=usubjid type=ar(1);  
  lsmeans trtn/diff ;  
  lsmeans trtn*visitcn/diff slice=visitcn;  
run;
```

R:

```
# USING REPEATED MEASURES – lme function for AR1  
model_chg_m = lme(CHG ~ BASE + VISITCN_F + TRTN_F + TRTN_F*VISITCN_F, random = ~1|USUBJID,  
correlation = corAR1(form = ~VISITCN | USUBJID), data = sas_ds1)  
lsm_calc_lst_chg_m <- emmeans(model_chg_m, revpairwise ~ TRTN_F*VISITCN_F, adjust="none")  
  
# USING REPEATED MEASURES - gls FUNCTION for AR1  
model_chg_m = gls(CHG ~ BASE + VISITCN_F + TRTN_F + TRTN_F*VISITCN_F, correlation =  
corAR1(0.8,form = ~VISITCN | USUBJID), data = sas_ds1)  
lsm_calc_lst_chg_m <- emmeans(model_chg_m, revpairwise ~ TRTN_F*VISITCN_F, adjust="none")
```

ANCOVA with Repeated Measures

◆ CASE-2: With Repeated Measures (Calculating LSM_CHG, LSM_CHG_SE and P_VALUE)

R (USING - lme function)	SAS
USING REPEATED MEASURE	USING REPEATED MEASURE - AR1
TRTN_F VISITCN_F TRTN_F_VISITCN_F_ P_VALUE	trtn _visitcn _trtn visitcn Probt
200 1 100 1 0.00000000	200 1 100 1 0.00000000
200 2 100 2 0.00000000	200 2 100 2 0.00000000
200 3 100 3 0.00000000	200 3 100 3 0.00000000
300 1 100 1 0.11732492	300 1 100 1 0.11564699
300 2 100 2 0.78996190	300 2 100 2 0.78966959
300 3 100 3 0.27522788	300 3 100 3 0.27382843
400 1 100 1 0.03772905	400 1 100 1 0.03639332
400 2 100 2 0.01104292	400 2 100 2 0.01027126
400 3 100 3 0.08405548	400 3 100 3 0.08242219
TRTN_F VISITCN_F LSM_CHG LSM_CHG_SE	trtn visitcn Estimate StdErr
100 1 23.095858 1.553381	100 1 23.095858 1.5533810
100 2 23.607640 1.553381	100 2 23.607640 1.5533810
100 3 22.003786 1.553381	100 3 22.003786 1.5533810
200 1 6.049564 1.530260	200 1 6.0495642 1.5302604
200 2 4.725343 1.530260	200 2 4.7253429 1.5302604
200 3 6.830973 1.530260	200 3 6.8309725 1.5302604
300 1 26.938943 1.870387	300 1 26.938943 1.8703873
300 2 24.257155 1.870387	300 2 24.257155 1.8703873
300 3 24.672317 1.870387	300 3 24.672317 1.8703873
400 1 28.150324 1.829602	400 1 28.150324 1.8296024
400 2 29.825170 1.829602	400 2 29.825170 1.8296024
400 3 26.191572 1.829602	400 3 26.191572 1.8296024

ANCOVA with Repeated Measures

◆ CASE-2: With Repeated Measures (Calculating LSM_CHG, LSM_CHG_SE and P_VALUE)

R (USING - gls function)	SAS
USING REPEATED MEASURE	USING REPEATED MEASURE - AR1
TRTN_F VISITCN_F TRTN_F_VISITCN_F_ P_VALUE	trtn _visitcn _trtn visitcn Probt
200 1 100 1 0.00000000	200 1 100 1 0.00000000
200 2 100 2 0.00000000	200 2 100 2 0.00000000
200 3 100 3 0.00000000	200 3 100 3 0.00000000
300 1 100 1 0.11510259	300 1 100 1 0.11564699
300 2 100 2 0.78957476	300 2 100 2 0.78966959
300 3 100 3 0.27337402	300 3 100 3 0.27382843
400 1 100 1 0.03596261	400 1 100 1 0.03639332
400 2 100 2 0.01002606	400 2 100 2 0.01027126
400 3 100 3 0.08189290	400 3 100 3 0.08242219
TRTN_F VISITCN_F LSM_CHG LSM_CHG_SE	trtn visitcn Estimate StdErr
100 1 23.095858 1.553381	100 1 23.095858 1.5533810
100 2 23.607640 1.553381	100 2 23.607640 1.5533810
100 3 22.003786 1.553381	100 3 22.003786 1.5533810
200 1 6.049564 1.530260	200 1 6.0495642 1.5302604
200 2 4.725343 1.530260	200 2 4.7253429 1.5302604
200 3 6.830973 1.530260	200 3 6.8309725 1.5302604
300 1 26.938943 1.870387	300 1 26.938943 1.8703873
300 2 24.257155 1.870387	300 2 24.257155 1.8703873
300 3 24.672317 1.870387	300 3 24.672317 1.8703873
400 1 28.150324 1.829602	400 1 28.150324 1.8296024
400 2 29.825170 1.829602	400 2 29.825170 1.8296024
400 3 26.191572 1.829602	400 3 26.191572 1.8296024

Conclusion

- ◆ Generating TFLs in R can be done
- ◆ Knowing defaults and differences between SAS and R is important to either re-creating same results or documenting why values are slightly different

Questions



Amol Waykar - amol.waykar@d-wise.com

Kevin Kramer - kevin.kramer@experis.com

Kalyani Komarasetti - kalyani.komarasetti@experis.com

Andrew Miskell - andrew.miskell@lilly.com