%ScoreQoL: SAS® Macro-Program that Automates the Scoring of Quality of Life Questionnaires

Mitchikou Pearl B. Tseng, PPD Inc., Taguig City, Philippines

ABSTRACT

The use of Quality of Life (QoL) questionnaires has been gaining attention in different clinical researches as patients' quality of life is becoming a standard endpoint in clinical trials. Results of analyzing these questionnaires bear significant importance since outcomes for survival and response to treatment are becoming more difficult to reach. Although the use of QoL questionnaires has slowly becoming an acceptable approach, programming the scores of each scale/domain is becoming tedious and repetitive since the scoring principle and imputation of missing values are consistent all throughout the questionnaire.

The most common scoring principle is taking the average of the responses in a specified list of items/questions that represent the scale/domain being measured. The average score is then linearly transformed using the range and lowest possible response. Afterwards, issues on missing values are addressed. Most QoL questionnaires just vary on the number of scales/domains being measured, number of items/questions that constitute a certain scale/domain, the range of responses and the cut-off (%) of missing values to be imputed. However, there is still no SAS® statistical package or macro program available which computes the scores of each scale and consequently, addresses the diverse characteristics of QoL questionnaires.

%ScoreQoL is a macro program that automates the scoring of scales/domains of different QoL questionnaires. It has macro parameters for the cut-off, scale name, scale code, item list, range, input and output data set, and reversed scoring. To aid in quality checks, a variable that provides information of each scale's number of missing items versus the expected count is included.

INTRODUCTION

Since maintaining and improving the patients' quality of life is now an important treatment goal in clinical trials in different therapeutic areas, the use of Quality of Life (QoL) questionnaires is gaining prominence in clinical researches. There have been numerous multidimensional self-administered QoL questionnaires that were developed and used in a lot of clinical trials to assess the treatment's impact on the disease and its effect on daily activities and behavior, perceived health measures, and disability/functional status of the patients.

Two of the most common QoL questionnaires are EORTC Quality of Life Questionnaire - Core 36 (EORTC QLQ – C30) and World Health Organization Quality of Life (WHOQOL-BREF). Each QOL questionnaire has its own specification which includes the number of domains of interest, items/question that constitute each domain, dimension of the rating scale and direction of scaling (positive or reversed) of each domain. However, programming the scores of each domain is becoming tedious and repetitive for the computation of the scores of each scale/domain since the scoring principle and imputation of missing values are consistent all throughout the questionnaire.

%ScoreQoL is a simple macro program that relieves the programmer from the tedious task of repeating the codes in computing the each score. In this paper, mock responses from EORTC QLQ – C30 will be used as an input data to visualize how %ScoreQoL works.

EORTC QLQ – C30 is 30-item measure which yields a computation of 15 scores that represent each scale per patient. Depending on the study, there is an option to compute the scores of scales from supplementary questionnaires depending on the disease being studied. Some common supplementary EORTC QLQ modules that are currently available are Breast cancer module (QLQ-BR23), Head & neck cancer module (QLQ-H&N35), Lung cancer module (QLQ-LC13), Ovarian cancer module (QLQ-OV28) and Multiple myeloma module (QLQ-MY20).

SCORING QOL QUESTIONNAIRES AND ADDRESSING MISSING ITEMS

The process shown below is the most common method for scoring each of the scale in different QOL questionnaires.

1. Compute for the average of all the items in a scale.

$$AVGScore = AVGS = \frac{\sum_{j=1}^{n} I_j}{n} \text{ , where } n \text{ is the expected number of items}$$

2. Standardize the average score (AVGS) using linear transformation to achieve a range between 0 and 100.

$$Fin\_Score_{Pos} = 100 \, x \left[\frac{AVGS - lowest\ possible\ response}{range} \right]$$

Range is the difference between the minimum and maximum response possible for each of the items. For example, a 4-point rating scale questionnaire has a range of 3 since 1 is the lowest possible response and 4 is the highest possible response. Note that, the final score specified above is for positively scaled domains. The final score for the domains with reversed scoring can be computed through subtracting the Fin_Score_{Pos} from 100 or can be computed using the formula below:

$$Fin\_Score_{Neg} = 1 - Fin\_Score_{Pos} = 100 \, x \left[1 - \frac{AVGS - lowest\ possible\ response}{range} \right]$$

HANDLING MISSING DATA

At present, there is still no consensus on the best method for dealing with missing QoL data in clinical trials. The commonly used method in different QoL instruments is the substitution of the average of the answered items to the missing values, given that a certain percentage of the number items from the multiple item scale are non-missing. The usual cut-off being used is 0.5. If the cut-off is not reached, the final score is set to missing.

1. Substitute the average of non-missing items to the missing items.

$$AVGScore = \frac{\left(\sum_{j=1}^{m} I_j\right) + (n-m)\frac{\sum_{j=1}^{m} I_j}{m}}{n},$$

$$where\ n\ is\ the\ expected\ number\ of\ items\ and\ m\ is\ the\ number\ of\ non-missing\ items$$

2. Proceed with the linear transformation of the final score as specified above.

HOW %SCOREQOL MACRO WORKS?

PREREQUESITES

The macro %ScoreQoL automates the computation of scores for each scale by using two SAS® data sets. The first data set is the source data which contains all the observations with the variables that represent the questionnaire items. Thus, the structure of the first data set is horizontal as seen on Display 1 below. Normally, each row represents the responses of each patient per visit. A score per scale will be computed for each row of the source data set.

For source data which is vertical or the question responses are listed per row, PROC TRANSPOSE can be used to transform the data.

Display 1. Source data set in Horizontal Form

The second data set contains the metadata of the questionnaire. Each row of the second data set provides the specifications for each scale. As seen on Display 2, the metadata data set should contain four required variables:

- EXP : character variable for the total number of questionnaire items required per scale

- PARAMCD : user specified character variable for the scale code or name
- ITEMLIST : character variable for the list of variables that represent the items in each scale separated by comma and must be inside '%STR( )'
- REVERSED : character variable for which a value of '1' should be assigned to a scale with reversed scoring and '0' if not

Display 2. Data set providing the specifications of the scales per row

	exp	paramcd	reversed	itemlist
1	5	PF	1	%STR(qs001, qs002, qs003, qs004, qs005)
2	2	RF	1	%STR(qs006, qs007)
3	4	EF	1	%STR(qs021, qs022, qs023, qs024)
4	2	CF	1	%STR(qs020, qs025)
5	2	SF	1	%STR(qs026, qs027)
6	2	QL	0	%STR(qs029, qs030)
7	3	FA	0	%STR(qs010, qs012, qs018)
8	2	NV	0	%STR(qs014, qs015)
9	2	PA	0	%STR(qs009, qs019)
10	1	DY	0	%STR(qs008)
11	1	SL	0	%STR(qs011)
12	1	AP	0	%STR(qs013)
13	1	CO	0	%STR(qs016)
14	1	DI	0	%STR(qs017)
15	1	FI	0	%STR(qs028)

Note that the value specified for the EXP variable should be the same as the count of the variable listed under the variable ITEMLIST. Also, each row of the metadata data set should be unique with respect to the value specified under variable PARAMCD. Creating additional rows in the metadata data set is optional to accommodate other scales from a supplementary questionnaire. A sample SAS® program that shows how to create the metadata data set is shown below.

```
/****************************************************************/
/* Create the metadata of the questionnaire                     */
/****************************************************************/

DATA metadata;
        /*EORTC QLQ - C30*/
        exp='5';
        paramcd='PF';
        reversed='1';
        itemlist='%STR(qs001, qs002, qs003, qs004, qs005)';
        OUTPUT;

        exp='2';
        paramcd='RF';
        reversed='1';
        itemlist='%STR(qs006, qs007)';
        OUTPUT;

        exp='4';
        paramcd='EF';
        reversed='1';
        itemlist='%STR(qs021, qs022, qs023, qs024)';
        OUTPUT;
        …
RUN;
```

COMPONENTS

%ScoreQoL macro program has three components. The first and main component is an inner macro, %AVAL, which individually computes the score of a certain scale being measured. The second component uses CALL EXECUTE to call the inner macro, %AVAL, multiple times, depending on the number of rows provided in the metadata data set. Consequently, there are *n* data sets that are created, where *n* is the number of scales in the questionnaire. Lastly, a DATA step which combines all the data sets using the SET statement is the third component that completes the macro program.

%ScoreQoL macro has the following characteristics:

- There are macro parameters for the cut-off value and range which can be changed depending on the requirements of the questionnaire.

- There are macro parameters for scale/domain codes, and input and output data set names.

- Since there are some scales which are constituted by negatively worded questions, a parameter in the inner macro is also included to identify items with reversed scoring.

- A variable NOTE which can be optionally kept for validation process and counter checking is also included. This variable provides one of the following information about the list of specified items in the scale: (a) Count of missing items exceeds cutoff, (b) No missing items and (c) Count of missing items: N

Display 3. Sample Output which includes the variable SCORE and NOTE

	paramcd	Unique Subject Identifier	Visit Name	aval	note
45831	EF	SUBJECT-1002-10001	SCREENING	0	No missing item
45832	EF	SUBJECT-1002-10002	VISIT 01	58.333333333	No missing item
45833	EF	SUBJECT-1002-10002	SCREENING	58.333333333	No missing item
50155	EF	SUBJECT-3910-01050	SCREENING	58.333333333	No missing item
50156	EF	SUBJECT-3910-01050	VISIT 02	50	Missing 2
50157	EF	SUBJECT-3910-01050	VISIT 03	.	Missing count exceeds cutoff
91661	PF	SUBJECT-1002-10001	SCREENING	6.6666666667	No missing item
91662	PF	SUBJECT-1002-10002	VISIT 01	60	No missing item
91663	PF	SUBJECT-1002-10002	SCREENING	60	No missing item
95985	PF	SUBJECT-3910-01050	SCREENING	66.666666667	No missing item
95986	PF	SUBJECT-3910-01050	VISIT 02	55.555555556	Missing 2
95987	PF	SUBJECT-3910-01050	VISIT 03	.	Missing count exceeds cutoff
109993	RF	SUBJECT-1002-10001	SCREENING	0	No missing item
109994	RF	SUBJECT-1002-10002	VISIT 01	50	No missing item
109995	RF	SUBJECT-1002-10002	SCREENING	50	No missing item
114317	RF	SUBJECT-3910-01050	SCREENING	66.666666667	No missing item
114318	RF	SUBJECT-3910-01050	VISIT 02	.	Missing count exceeds cutoff
114319	RF	SUBJECT-3910-01050	VISIT 03	66.666666667	Missing 1

MACRO CALL

A sample output of the macro is shown in Display 3 can be seen above. The entire code for this %ScoreQoL macro is given in the Appendix to this paper, and the sample call of the macro is shown on the next page.

The input parameters to the macro program are as follows:

- source_data : The name of the SAS® data set which contains the questionnaire responses

- metadata : The name of the SAS® data set which contains the specification of the questionnaires

- out_data : The name of the output data set which contains all the scores for each scale

- cutoff : The cut-off value for addressing missing data for which valid values range from 0 to 1

- range : The difference between the highest and lowest possible response

```
/*****************************************************************/
/* Sample MACRO CALL
/*****************************************************************/
%Score_QoL(   source_data   = SOURCE,
              metadata      = METADATA,
              out_data      = FINAL,
              cutoff        = 0.5,
              range         = 3
              );
```

CONCLUSION

Programming the computation of the scores of each scale is less tedious and repetitive using %Score_QoL since providing the specifications of the questionnaire in data set form and the source data set is sufficient for the macro to work. Moreover, %ScoreQoL macro can be used in scoring any QoL questionnaire which follows the same scoring principle and same handling of missing data.

REFERENCES

- Fayers PM, Aaronson NK, Bjordal K, Groenvold M, Curran D, Bottomley A, on behalf of the EORTC Quality of Life Group. (2001). *The EORTC QLQ-C30 Scoring Manual (3rd Edition)*. 6-7. Published by: European Organisation for Research and Treatment of Cancer, Brussels. Available at http://www.eortc.be/qol/files/SCManualQLQ-C30.pdf.

- Welch B, Rhee S. 2012. "Multiple Techniques for Scoring Quality of Life Questionnaires". *Proceedings of the PharmaSUG China 2012 Conference*. Available at https://www.pharmasug.org/proceedings/2012/HO/PharmaSUG-2012-HO01.pdf.

- World Health Organization (WHO). 2012. *Programme on Mental Health: WHOQOL User Manual*. 52-54. World Health Organization, Geneva. Available at http://apps.who.int/iris/bitstream/10665/77932/1/WHO_HIS_HSI_Rev.2012.03_eng.pdf.

ACKNOWLEDGMENTS

CONTACT INFORMATION

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

Name: Mitchikou Pearl Tseng
Enterprise: PPD Pharmaceutical Development Philippines Corp.
Address: 22/F Net Park Building, 5th Avenue, Crescent Park West
City, State ZIP: Taguig City, Philippines 1634
Work Phone: +632 505 7598
E-mail: mitchikoupearl.tseng@ppdi.com

APPENDIX: SOURCE CODE OF MACRO

```
/****************************************************************/
/* MACRO %SCORE_QOL
/****************************************************************/
%MACRO Score_QoL(source_data=, metadata=, out_data=, cutoff=, range=);

*****FIRST COMPONENT*****;
      %MACRO aval(itemlist=, exp=, paramcd=, reversed=);
      DATA aval_&paramcd.;
            LENGTH paramcd $8 Note $40;
            SET &source_data.;

*If the percentage of missing values exceeds the cutoff set the score to missing;
            perc=N(&itemlist.)/ &exp.;
            IF perc < &cutoff. THEN DO;
                  aval=.;
                  paramcd="&paramcd.";
                  note="Missing count exceeds cutoff";
                  OUTPUT;
                  END;

            ELSE IF perc >= &cutoff. THEN DO;
                  IF N(&itemlist.) = &exp. THEN DO;
                        aval=100 *((MEAN(&itemlist.) - 1) / &range.);
                              IF &reversed.=1 THEN aval=100-aval;
                        paramcd="&paramcd.";
                        note="No missing item";
                        OUTPUT;
                        END;
                  *Mean Imputation;
                  ELSE IF N(&itemlist.) < &exp. THEN DO;
                        inc_avg=(SUM(&itemlist.)/N(&itemlist.))*NMISS(&itemlist.);
                        all_avg= SUM(SUM(&itemlist.), inc_avg)/&exp.;
                        aval=((all_avg- 1)/ &range.) * 100;
                              IF &reversed.=1 THEN aval=100-aval;
                        paramcd="&paramcd.";
                        note="Missing "||STRIP(PUT(NMISS(&itemlist.),best.));
                        OUTPUT;
                  END;
            END;
      RUN;
      %MEND aval;

*****SECOND COMPONENT*****;
      DATA _null_;
        SET &metadata.( obs=max ) ;
            CALL EXECUTE(  '%nrstr(%aval(itemlist= '
                        !! strip(strip(itemlist))
                        !! ', exp= '
                        !! strip( exp )
                        !! ', paramcd= '
                        !! strip( paramcd )
                        !! ', reversed=  '
                        !! strip( reversed )
                        !! ') )' ) ;
       RUN;

*****THIRD COMPONENT*****;
      DATA &out_data. (Drop = perc inc_avg all_avg);
            SET aval_:;
      RUN;
%MEND Score_QoL;
```