

SPSS Code for Longstring Screening

by John A. Johnson and John D. Mayer

4th Edition 2020-11-27

A note on formatting: The word “comment” appears repeatedly beginning in the left-hand first column of this document, and those lines end (either on the line itself or shortly thereafter) with a period. This convention is maintained so that the comments and code can be placed into SPSS directly if desired.

Note that, in the “Code Itself” section, the instructions, which are yellow-highlighted, cannot be copied in unless they are commented first.

Contents

comment HISTORY AND CITATION.	3
comment LONGSTRING RESPONSDING EXPLAINED	3
comment THE SCALE USED HERE AN EXAMPLE, DESCRIBED.....	4
comment BRIEF DESCRITPION OF THE SEPI-16.....	4
comment THE CODE ITSELF AND INSTRUCTIONS FOR ITS USE.....	5
comment APPENDIX A. AN EXPLANATION OF THE CODE ITSELF	8
comment APPENDIX B. A POSSIBLE ISSUE IN SPSS--CAVEAT EMPTOR.....	11

comment -----.

comment HISTORY AND CITATION.

comment -----.

comment Sometime around 2018, John A. Johnson, Professor of Psychology at Pennsylvania State University, comment provided us (the UNH Personality Lab) with the following code.

comment The original version was the source code used to detect longstring responding, comment as reported in

comment Johnson, John A. (2005). Ascertaining the validity of individual protocols from comment Web-based personality inventories.

comment Journal of Research in Personality, Vol 39(1), Feb, 2005 pp. 103-129.

comment J.D. Mayer then rewrote the code, modifying the variable names and adding further comments and an example.

comment THIS DOCUMENT CAN BE CITED AS:

comment Johnson, J. A. & Mayer, J. D. (2019). SPSS Code for Longstring Screening. Electronic document (3rd Ed). Downloaded from [URL here]

comment This document is protected by a Creative Commons 4.0 License (CC BY 4.0).

comment You are free to redistribute, remix and transform the material herein, so long as you give appropriate credit to the original authors comment and document and indicate if changes were made. (See <https://creativecommons.org/licenses/by/4.0/>) for details.

comment -----.

comment LONGSTRING RESPONDING EXPLAINED.

comment -----.

comment Longstring responding arises when a respondent answers with the same alternative, for example

comment answering "A, A, A, A, A, A,..." or "3, 3, 3, 3, 3, 3, 3" on item-after-item on a survey. The following code detects that comment behavior.

comment -----.

comment THE SCALE USED HERE AN EXAMPLE, DESCRIBED.

comment -----.

comment -----.

comment this version of the code is written to examine longstring responding
Comment in a 16-item scale called the self-estimated personal intelligence-16 (sepi-16)
comment (see, for example, Mayer, J. D., Lortie, B., Panter, A. T., & Caruso, D. (2018).
comment Employees high in personal intelligence differ in workplace perceptions and behavior from their colleagues.
comment Journal of Personality Assessment, 00, 000-000 [First published online].
comment <https://doi.org/10.1080/00223891.2018.1455690>
comment *Note:* THE CODE CAN BE READILY ADAPTED TO ANY MEASURE THAT USES A CONSISTENT
comment RESPONSE SCALE ACROSS MULTIPLE ITEMS

comment -----.

comment -----.

comment BRIEF DESCRITPION OF THE SEPI-16

comment -----.

comment This example concerns the 16-item Self-Estimate of Personal Intelligence Scale (the SEPI).
comment The scale consists of 16 items. Each item is responded to on a 5 point likert scale from 1 to 5.
comment To provide an idea of the test, an item might read: "I understand other people's personalities very well (click one:)" (or)
comment "I understand myself"
comment ...and then the participant would choose:
comment "1. Not at all".
comment "2".
comment "3. Somewhat".
comment "4".
comment "5. Extremely Well".

comment-----.

comment THE CODE ITSELF AND INSTRUCTIONS FOR ITS USE

comment-----.

STRONG RECOMMENDATION: DO NOT COPY THIS CODE DIRECTLY INTO SPSS. RATHER, FIRST COPY THE MATERIAL BELOW INTO A NEW .TXT FILE TO REMOVE ANY ODD, HIDDEN CHARACTERS. THEN COPY IT FROM THE TEXT FILE INTO SPSS.

```
comment          longstring responding for SEPI16.
```

comment Assuming you have cases with missing values, I recommend
comment that you recode them as the value next beyond the possible responses to the scale.
comment i.e., "6" in the case of a 5-point scale, or "8" in the case
comment of a 7-point scale.
comment Here, the missing values are set to 6.

```
RECODE S1 TO S16 (SYSMIS = 6).
RECODE S1 TO S16 (-99 = 6).
```

comment Now, double-check with Frequencies that responses for all items fall within the expected range.
comment That is, if you have a response scale of from 1 to 5, all variables should fall within
comment the range of 1 to 5.
comment In addition, check that the N+1 alternative, i.e., "6" in the case of a 5-point scale, or "8" in the case
comment of a 7-point scale, is reserved for recodes of missing values to that number.

```
FREQUENCIES VARIABLES EQ S1 TO S16.
```

comment Now set up a vector to hold participant responses to each of the test items.

VECTOR ANS_SEPI = S1 TO S16.

COMMENT Instruction: All responses to items are expected to be on the same response scale.
 COMMENT The next line refers to a 5-point response scale as used on the SEPI. If you are using
 COMMENT a 7-point scale, then instead of 1 to 6, the L, LLONG, etc., should go from 1 to 8.
 COMMENT Missing values should be set to the maximum response on the scale + 1—just for
 COMMENT this check (you can set the missing values back to your standard value after finishing COMMENT with this code.

DO REPEAT L= LLONG1 TO LLONG6, TMPLONG1 TO TMPLONG6.
 COMPUTE L = 1.
 END REPEAT.

COMMENT Instruction: Change the 1 to 6 below to whatever your numerical range is (plus one if COMMENT or missing values).

VECTOR LLONG=LLONG1 TO LLONG6.
 VECTOR TMPLONG=TMPLONG1 TO TMPLONG6.

COMMENT Instruction: Change “=16” in the next line to however many items are on your test, COMMENT i.e., it would be “= 44” instead of = 16
 for the BigFive.

COMPUTE Kitems = 16.
 LOOP NEXT=2 TO Kitems.
 COMPUTE INDX=NEXT-1.

COMPUTE ALTN=ANS_SEPI(NEXT).

COMMENT Instructions: If your missing value code was 99, for example,
 COMMENT change it to the “plus one” value of your scale here;
 COMMENT that is, if you had a five-point response scale as with the SEPI,
 COMMENT it should be “EQ 6”; if you used a 7-point response scale, make it “EQ 8”.
 COMMENT THIS IS NOW HANDLED ABOVE.

COMMENT Instructions: Note, you will need to substitute the acronym for your scale for the SEPI
 COMMENT wherever it appears below, unless you are using the SEPI.

```

IF (ANS_SEPI(NEXT)) EQ (ANS_SEPI(INDX)) TMPLONG(ALTN) = TMPLONG(ALTN) + 1.
IF (TMPLONG(ALTN) GT LLONG(ALTN)) LLONG(ALTN) = TMPLONG(ALTN).
IF (ANS_SEPI(NEXT) NE ANS_SEPI(INDX)) TMPLONG(ALTN) = 1.
END LOOP.

```

COMMENT Instructions: The code below is to examine maximum longstring responding for a COMMENT test with 5 alternatives and missing data.
COMMENT If you have more than that, you will need to add "llong7, llong8..." and so forth until COMMENT you reach the end of your possibilities.

```
compute sepilongstr = max(llong1, llong2, llong3, llong4, llong5, llong6).
```

COMMENT and we check our results.

```
frequencies variables = llong1 to llong6, sepilongstr.
```

```
list variables = sepilongstr, llong1, llong2, llong3, llong4, llong5, llong6, s1 to s16.
```

comment Finally, remember to revert your dummy-coded missing values back to actual missing
comment values before doing further analyses--and check that the conversion is correct.

```
recode s1 to s16 (6 = -99).
```

```
missing values s1 to s16 (-99).
```

```
frequencies variables = s1 to s16.
```

comment-----.

comment APPENDIX A. AN EXPLANATION OF THE CODE ITSELF

comment-----.

comment We first set up the vector, ANS_SEPI, that contains the test responses in order.

comment The SPSS "vector" command allows us to designate answers with subscripts.

comment Note that to print, compute, or otherwise work with vectors in SPSS, once defined, they always have a subscript.

VECTOR ANS_SEPI = SEPI1 TO SEPI16.

comment Next, we begin working with the variables llong and tmlong 1 through 5.

comment llong designates the longest string; tmlong designates what is temporarily the longest string.

comment In this next step, we initialize all the variables at 1.

comment The variable L takes on sequentially the values of llong 1 through 5,

comment The values are 1 through 5 because there are five alternatives (from 1 "Not at All" to 5 "Extremely Well")

comment to each item of this SEPI form. Tmlong also is 1 through 5 for the same reason.

comment By setting L to 1, and because L takes on all the values,

comment all longstring and temporary longstring variables are set equal to 1 at first.

COMMENT NOTE IN THIS VERSION WITH MISSING DATA, MISSING IS PLACED IN INDEX AREA 6--LLONG6 AND TMPLONG6.

comment I suggest that even if you don't have any missing data, you use this convention in case some comment regarding

comment the "[6]" position arises up later.

DO REPEAT L= LLONG1 TO LLONG6, TMPLONG1 TO TMPLONG6.

COMPUTE L = 1.

END REPEAT.

comment This section defines the, LLONG and TMPLONG variables as vectors

comment which contain the variables we just defined as all equal to 1 to start.

VECTOR LLONG=LLONG1 TO LLONG6.

VECTOR TMPLONG=TMPLONG1 TO TMPLONG6.

comment Next, we set up the index variables called NEXT and INDX, which will become subscripts of the vectors.
 comment It is important to use the exact names NEXT and INDX (unless you are an experienced programmer and fully
 comment understand these lines of code).
 comment INDX starts out as the subscript of the initial (first) variable and then increments by +1 each step.
 comment NEXT is the subscript for the variable that follows the initial (first) variable in the vector, with the earlier variable designated by the
 comment subscript INDX (see INDX immediately above).

comment In the following LOOP, Kitems is the number of test items.

COMPUTE Kitems = 16.

comment The first time through the loop, the value of the
 comment second variable (subscripted NEXT) is compared to the value of the first variable (subscripted INDX).
 comment When NEXT increments the next time through the loop, the value of the third variable is compared to the value
 comment of the second variable.
 comment And so forth, until the value of the Kitem-th variable (the last variable) is compared to the next-to-the-last variable.

comment NEXT takes on the values, sequentially, of 2 to Kitems.
 comment At each step of the loop, INDX is equal to NEXT-1.
 comment At each step of the loop, ALTN is equal to 1,2,3,4 5, the participant's chosen alternative.
 comment In other words, INDX might be SEPI-16 var 1 and NEXT would be var 2.

LOOP NEXT=2 TO Kitems.
 COMPUTE INDX=NEXT-1.

comment altn--> the chosen alternative 1, 2, 3...Nalternatives.

COMPUTE ALTN=ANS_SEPI(NEXT).

comment back to missing data again--reassigning 99--the original missing-value code--to 6.

IF (ALTN EQ 99) ALTN EQ 6.

comment indx and next are the subscripts of the vectors containing the participant responses:
 comment the original position (i.e., response to the third item), and the one right after (i.e., fourth item response).

comment Now if the variable ANS_SEPI(NEXT) is equal to the variable just before it, TMPLONG is incremented by 1.

```
IF (ANS_SEPI(NEXT)) EQ (ANS_SEPI(INDX)) TMPLONG(ALTN) = TMPLONG(ALTN) + 1.
```

comment Step 2. If TMPLONG is greater than LLong, LLong is set equal to TMPLong.

comment In other words, LLong "remembers" how long the longest TMPLong is (or was), proceeding through the loop.

comment LLong is the "longest long"

comment tmlong is the "temporary longest".

comment ALTN designates if the string is of 1s, 2s, 3s, ...to the maximum choice value (5, in this case).

comment That is, we are keeping track of the longest strings of all five variables at once, switching

comment among them as does the participant's answers.

```
IF (TMPLONG(ALTN) GT LLONG(ALTN)) LLONG(ALTN) = TMPLONG(ALTN).
```

```
IF (ANS_SEPI(NEXT) NE ANS_SEPI(INDX)) TMPLONG(ALTN) = 1.
```

```
END LOOP.
```

comment now we use the internal SPSS function "max" to grab the longest string.

```
compute sepilongstr = max(llong1, llong2, llong3, llong4, llong5, llong6).
```

comment and we check our results.

```
frequencies variables = llong1 to llong6, sepilongstr.
```

```
list variables = sepilongstr, llong1, llong2, llong3, llong4, llong5, llong6, sepi1 to sepi16.
```

comment -----.

comment APPENDIX B. A POSSIBLE ISSUE IN SPSS--CAVEAT EMPTOR.

comment -----.

comment The code contains the following statement near the very end
 comment IF (TMPLONG(ALTN) GT LLONG(ALTN)) LLONG(ALTN) = TMPLONG(ALTN).

comment In one out of five or ten implementations of this code, SPSS generates an error message on the order of
 comment "ERROR: LOOP INDEX OUT OF RANGE"
 comment SPSS is generally extremely "well-behaved." That said, Mayer contacted SPSS about this in 2017 and
 comment SPSS tech support acknowledged it was a bug on their end and also said that the results provided in the
 comment output (the software kept calculating) were nonetheless correct, which did appear to be the case.
 comment Since 2017, we have successfully implemented this code in many data analyses without issue.
 comment But the problem continues to be unresolved as we encountered it again in Dec. 2019.

comment We do not like unexplained error messages in our output, so we developed the following work around:
 comment If you are unlucky and encounter that error message, rewrite the above IF statement to simplify it.
 comment Take out the ALTN--the index variable for (in the SEPI scale example described above) the 5 choice alternatives
 comment from "Not at all" to "Extremely Well" and break down
 comment the code into five statements as follows:

comment IF (TMPLONG(1) GT LLONG(1)) LLONG(1) = TMPLONG(1).
 comment IF (TMPLONG(2) GT LLONG(2)) LLONG(2) = TMPLONG(2).
 comment IF (TMPLONG(3) GT LLONG(3)) LLONG(3) = TMPLONG(3).
 comment IF (TMPLONG(4) GT LLONG(4)) LLONG(4) = TMPLONG(4).
 comment IF (TMPLONG(5) GT LLONG(5)) LLONG(5) = TMPLONG(5).

comment and insert those as a replacement for the original IF statement in the code.

comment That should fix the problem...good luck!