The Stata Journal (2018)
18, Number 1, pp. 29–50

validscale: A command to validate measurement scales

Bastien Perrot
University of Nantes, University of Tours, INSERM
SPHERE U1246 "methodS in Patient-centered outcomes and HEalth ResEarch"
Nantes, France
bastien.perrot@univ-nantes.fr

Emmanuelle Bataille
University of Nantes, University of Tours, INSERM
SPHERE U1246 "methodS in Patient-centered outcomes and HEalth ResEarch"
Nantes, France
emmanuelle.anthoine@chu-nantes.fr

Jean-Benoit Hardouin
University of Nantes, University of Tours, INSERM
SPHERE U1246 "methodS in Patient-centered outcomes and HEalth ResEarch"
Nantes, France
jean-benoit.hardouin@univ-nantes.fr

Abstract. Subjective measurement scales are used to measure nonobservable respondent characteristics in several fields such as clinical research, educational sciences, or psychology. To be useful, the scores resulting from the questionnaire must be validated; that is, they must provide the psychometric properties validity, reliability, and sensitivity. In this article, we present the `validscale` command, which carries out the required statistical analyses to validate a subjective measurement scale. We have also developed a dialog box, and `validscale` will soon be implemented online with Numerics by Stata.

Keywords: st0512, validscale, subjective measurements scales, score, psychometrics, validity, reliability

1 Introduction

Several domains require measurement scales to measure concepts, but no technical instrument in these domains allows one to obtain a measure. This is the case, for example, in education sciences, when we want to measure the ability of students; in social and human sciences, when we want to measure characteristics of individuals like personality traits or behaviors; and in health, when we want to measure perceived health, quality of life, pain, or mental disorders.

Generally, these concepts are measured using a questionnaire composed of items. Questionnaires can be unidimensional (they measure only one concept) or multidimen-

 st0512

sional (they measure several concepts), so they can lead to one or more measures able to measure the concept or concepts of interest. Historically, the model of measurement is defined in the framework of classical test theory (CTT). If other theories of measurement coexist today like item response theory (IRT) or Rasch measurement theory (RMT), CTT continues to be largely used in several domains (psychology, health, etc.) to validate scales. The success of CTT can be explained by the simplicity of obtaining the measures, because, in this framework, the measure of each concept can be obtained using scores computed as a combination (for example, sum or mean) of responses to the items.

To construct a valid and reliable questionnaire, one must provide its psychometric properties: Validity is the degree to which an instrument measures the concept or concepts of interest accurately. Reliability is the degree to which an instrument measures the concept or concepts of interest consistently. Validity and reliability are assessed by checking their respective facets. Validity is composed of content validity, construct validity, and criterion validity. Reliability is composed of internal consistency, test-retest reliability, and scalability.

Content validity should be evaluated by experts. In this step, the experts define the number of concepts measured by the scale, the definition of each of these concepts, and the assumed relationships between the concepts and the questionnaire items. This facet of validity is based on qualitative methods, but all the other facets can be assessed using statistical analyses to statistically confirm the experts' opinions. However, there is currently no statistical software package to perform all of these analyses in a user-friendly way. The objective of the `validscale` command is to perform the necessary analyses to validate a measurement scale in the framework of CTT.

2 Psychometric properties assessed by validscale

The concepts described below are based on the terminology used by Fayers and Machin (2007).

Construct validity:

Convergent and divergent validity test whether the items of the questionnaire measure the constructs they are designed to measure. Convergent validity tests whether an item is sufficiently correlated to the score computed with items of the same dimension. Divergent validity tests whether an item is poorly correlated to the scores computed in the other dimensions (for multi-dimensional scales).

Structural validity tests the dimensional structure of the questionnaire (the number of dimensions).

Criterion validity:

Concurrent validity is assessed by comparing the score or scores of the questionnaire with previously validated instruments or a gold standard measuring approximately the same concept or concepts.

Known-groups validity tests whether the scores differ according to known groups in a predictable way.

Reliability:

Internal consistency refers to how unidimensional the dimension is and whether it is composed of enough items.

Reproducibility refers to how well the items and scores are stable when the individual's state is stable.

3 The validscale command

3.1 Description

The `validscale` command computes elements to provide structural validity, convergent and divergent validity, concurrent validity, reproducibility, known-groups validity, internal consistency, and scalability. The command is intended to be used with questionnaires that comprise dichotomous (that is, two response categories) or polytomous items (that is, more than two response categories, for example, the Likert-type scale). The user defines the items used to compute the scores. The second parameter required (`partition()`) is the repartition of the items in the different dimensions of the questionnaire.

3.2 Syntax

`validscale` *varlist*`,` `partition(`*numlist*`)` [`scorename(`*string*`)` `scores(`*varlist*`)` `categories(`*numlist*`)` `impute(`*method*`)` `noround` `compscore(`*method*`)` `descitems` `graphs` `cfa` `cfamethod(`*method*`)` `cfasb` `cfastand` `cfanocovdim` `cfacov(`*string*`)` `cfarmsea(`#`)` `cfacfi(`#`)` `cfaor` `convdiv` `tconvdiv(`#`)` `convdivboxplots` `alpha(`#`)` `delta(`#`)` `h(`#`)` `hjmin(`#`)` `repet(`*varlist*`)` `kappa` `ickappa(`#`)` `scores2(`#`)` `kgv(`*varlist*`)` `kgvboxplots` `kgvgroupboxplots` `conc(`*varlist*`)` `tconc(`#`)`]

varlist contains the variables (items) used to compute the scores. The first items of *varlist* compose the first dimension, the following items define the second dimension, and so on.

validscale requires the commands delta (Hardouin 2007a), detect2 (included with the software for this article), imputeitems (Hardouin 2007b), mi_twoway (Hamel 2014), kapci (Reichenheim 2004), loevh (Hardouin 2004), and lstrfun (Blanchette 2010).

3.3 Options

partition(*numlist*) defines the number of items in each dimension. The number of elements in this list indicates the number of dimensions. partition() is required.

scorename(*string*) defines the names of the dimensions. By default, the dimensions are named Dim1, Dim2, ... unless scores(*varlist*) is used.

scores(*varlist*) selects scores already computed in the dataset. *varlist* must contain as many variables as there are dimensions in the questionnaire. scores(*varlist*) and scorename(*string*) cannot be used together. This option is useful when the scores result from combinations of items that are more complex than the combinations available in the compscore() option (unweighted sum, unweighted mean, or standardization between 0 and 100). In that case, the scores must be computed prior to using validscale.

categories(*numlist*) specifies the minimum and maximum possible values for item response categories. If all the items have the same response categories, the user may specify these two values in *numlist*. If the item response categories differ from a dimension to another, the user must define the minimum and maximum values of items responses for each dimension. So the number of elements in *numlist* must be equal to the number of dimensions times 2. Eventually, the user may specify the minimum and maximum response categories for each item. In this case, the number of elements in *numlist* must be equal to 2 times the number of items. By default, the observed minimum and maximum values are assumed to be the minimum and maximum for each item.

impute(*method*) imputes missing-item responses with person-mean substitution (pms) or the two-way imputation method applied in each dimension (mi). Both methods are applied in each dimension. When pms is specified, missing data are imputed only if the number of missing values in the dimension is less than half the number of items in the dimension.

By default, imputed values are rounded to the nearest whole number, but with the noround option, imputed values are not rounded. If impute() is absent, then noround is ignored.

noround specifies that imputed values are not rounded. By default, imputed values are rounded to the nearest whole number. If impute() is absent, then noround is ignored.

`compscore(`*method*`)` defines the method used to compute the scores. *method* may be `mean` (default), `sum`, or `stand` (set scores from 0 to 100). `compscore()` is ignored if `scores()` is used.

`descitems` displays a descriptive analysis of the items. The option displays missing data rate per item and distribution of item responses. It also computes for each item the Cronbach's alphas obtained by omitting each item in each dimension. Moreover, the option computes Loevinger's H_j coefficients and the number of nonsignificant H_{jk}. See Hardouin, Bonnaud-Antignac, and Sébille (2011) for details about Loevinger's coefficients.

`graphs` displays graphs for items' and dimensions' descriptive analyses. It provides histograms of scores, a biplot of the scores, and a graph showing the correlations between the items.

`cfa` performs a confirmatory factor analysis (CFA) using the `sem` command. It displays estimations of parameters and several goodness-of-fit indices.

`cfamethod(`*method*`)` specifies the method to estimate the parameters. `method` may be `ml` (maximum likelihood), `mlmv` (`ml` with missing values), or `adf` (asymptotic distribution free).

`cfasb` produces Satorra–Bentler-adjusted goodness-of-fit indices by using the option `vce(sbentler)` from `sem`.

`cfastand` displays standardized coefficients for the CFA.

`cfanocovdim` asserts that the latent variables are not correlated.

`cfacov(`*string*`)` adds covariances between measurement errors. `cfacov(`*item1*`*`*item2*`)` estimates the covariance between the errors of *item1* and *item2*. To specify more than one covariance, use `cfacov(`*item1*`*`*item2 item3*`*`*item4*`)`.

`cfarmsea(#)` automatically adds the covariances between measurement errors found with the `estat mindices` command until the root mean square error of approximation (RMSEA) of the model is less than #. More precisely, the "basic" model (without covariances between measurement errors) is fit; then we add the covariance corresponding to the greatest modification index. The model is then fit again with this extra parameter, and so on. The option adds only the covariances between measurement errors within a dimension and can be combined with `cfacov()`. The specified value # may not be reached if all possible within-dimension measurement errors have already been added.

`cfacfi(#)` automatically adds the covariances between measurement errors found with the `estat mindices` command until the comparative fit index (CFI) of the model is greater than #. More precisely, the "basic" model (without covariances between measurement errors) is fit; then we add the covariance corresponding to the greatest modification index. The model is then fit again with this extra parameter, and so on. The option adds only the covariances between measurement errors within a dimension and can be combined with `cfacov()`. The specified value # may not be reached if all possible within-dimension measurement errors have already been added.

`cfaor` is useful when both `cfarmsea()` and `cfacfi()` are used. By default, covariances between measurement errors are added, and the model is fit until both RMSEA and CFI criteria are met. If `cfaor` is used, the estimations stop when one of the two criteria is met.

`convdiv` assesses convergent and divergent validities. The option displays the matrix of correlations between items and rest scores (that is, the scores computed after omitting the item being examined). If `scores(`*varlist*`)` is used, then the correlation coefficients are computed between items and scores of *varlist*.

`tconvdiv(#)` defines a threshold for highlighting some values. # is a real number between 0 and 1. The default is `tconvdiv(0.4)`. Correlations between items and their own score are displayed in red if it is less than #. Moreover, if an item has a smaller correlation coefficient with the score of its dimension than those computed with other scores, the coefficient is displayed in red.

`convdivboxplots` displays boxplots for assessing convergent and divergent validities. The boxes represent the correlation coefficients between the items of a given dimension and all scores. Thus, the box of correlations coefficients between items of a given dimension and the corresponding score must be higher than other boxes. There are as many boxplots as dimensions.

`alpha(#)` defines a threshold for Cronbach's alpha. # is a real number between 0 and 1. The default is `alpha(0.7)`. Alpha coefficients less than # are displayed in red.

`delta(#)` defines a threshold for Ferguson's delta coefficient (see `delta`). Delta coefficients are computed only if `compscore(sum)` is used and `scores()` is not used. # is a real number between 0 and 1. The default is `delta(0.9)`. Delta coefficients less than # are displayed in red.

`h(#)` defines a threshold for Loevinger's H coefficients (see `loevh`). # is a real number between 0 and 1. The default is `h(0.3)`. Loevinger's coefficients less than # are displayed in red.

`hjmin(#)` defines a threshold for Loevinger's H_j coefficients. The displayed value is the minimal H_j coefficient for an item in the dimension (see `loevh`). # is a real number between 0 and 1. The default is `hjmin(0.3)`. If the minimal Loevinger's H_j coefficient is less than #, then it is displayed in red, and the corresponding item is displayed.

`repet(`*varlist*`)` assesses reproducibility of scores by defining in *varlist* the variables corresponding to responses at time 2 (in the same order than for time 1). Scores are computed according to the `partition()` option. Intraclass correlation coefficients (ICC) for the scores and their 95% confidence interval are computed with Stata's `icc` command.

`kappa` computes the kappa statistic for items with Stata's `kap` command.

`ickappa(#)` computes confidence intervals for kappa statistics using `kapci`. # is the number of replications for the bootstrap used to estimate confidence intervals if items are polytomous. If they are dichotomous, an analytical method is used. See `kapci` for more details about calculation of confidence intervals for kappa's coefficients. If the `kappa` option is absent, then the `ickappa()` option is ignored.

`scores2(`*varlist*`)` allows selecting scores at time 2 from the dataset.

`kgv(`*varlist*`)` assesses known-groups validity according to the grouping variables defined in *varlist*. The option performs an analysis of variance (ANOVA) that compares the scores between groups of individuals constructed with variables in *varlist*. A *p*-value based on a Kruskal–Wallis test is also given.

`kgvboxplots` draws boxplots of the scores split into groups of individuals.

`kgvgroupboxplots` groups all boxplots into one graph. If `kgvboxplots` is absent, then `kgvgroupboxplots` is ignored.

`conc(`*varlist*`)` assesses concurrent validity with variables precised in *varlist*. These variables are scores from one or several other scales in the dataset.

`tconc(#)` defines a threshold for the correlation coefficients between the computed scores and those from other scales defined in *varlist*. Correlation coefficients greater than # in absolute value (the default is `tconc(0.4)`) are displayed in bold.

4 Output

4.1 Data used in the examples

The data used in the output are a sample of responses to the French version of the Impact Of Cancer version 2 questionnaire (Crespi et al. 2008). This questionnaire measures the impact of cancer on survivors' lives. It consists especially of four positive-impacts subscales named altruism and empathy (`AE`), health awareness (`HA`), meaning of cancer (`MOC`), and positive self-evaluation (`PSE`) and four negative-impacts subscales named appearance concerns (`AC`), body change concerns (`BCC`), life interferences (`LI`), and worry (`W`). There is also one subscale for employment and relationship concerns (not used here).

The questionnaire is composed of 37 items (`ioc1`–`ioc37`) scored on a 5-point scale from 1 ("strongly disagree") to 5 ("strongly agree"). They are grouped into eight dimensions, and the scores result from the sum of the responses of the corresponding items. The questionnaire was answered by 371 patients.

4.2 Examples

Minimal output

```
. use data_ioc
. validscale ioc1-ioc37, partition(4 4 7 3 3 4 7 5)
> scorename(HA PSE W BCC AC AE LI MOC) compscore(sum)
Items used to compute the scores
HA : ioc1 ioc2 ioc3 ioc4
PSE : ioc5 ioc6 ioc7 ioc8
W : ioc9 ioc10 ioc11 ioc12 ioc13 ioc14 ioc15
BCC : ioc16 ioc17 ioc18
AC : ioc19 ioc20 ioc21
AE : ioc22 ioc23 ioc24 ioc25
LI : ioc26 ioc27 ioc28 ioc29 ioc30 ioc31 ioc32
MOC : ioc33 ioc34 ioc35 ioc36 ioc37
Number of observations: 371
--------------------------------------------------------------------------------
Reliability
--------------------------------------------------------------------------------

            n     alpha     delta         H    Hj_min
HA        369      0.67      0.94      0.35      0.25 (item ioc1)
PSE       368      0.69      0.96      0.39      0.30
W         369      0.90      0.99      0.62      0.59
BCC       369      0.79      0.97      0.61      0.58
AC        369      0.81      0.97      0.62      0.60
AE        368      0.71      0.94      0.43      0.34
LI        367      0.81      0.97      0.42      0.29 (item ioc26)
MOC       363      0.83      0.97      0.53      0.38
```

This is the minimal output produced by `validscale`. In this example, we see that all of Cronbach's alphas are acceptable according to the threshold specified with `alpha(#)` (0.7 by default), except for scales `HA` and `PSE` (0.67 and 0.69, respectively).

Loevinger's H coefficients for the 8 scales are ≥ 0.3, which indicates good scalability. However, the $H^{\texttt{HA}}_{\texttt{ioc1}}$ coefficient is < 0.3 (0.25), which means that item `ioc1` might not be consistent with scale `HA`. The $H^{\texttt{LI}}_{\texttt{ioc26}}$ coefficient is also < 0.3 for item `ioc26`. For other subscales, no $H^{\text{subscale}}_{\text{item}}$ is < 0.3.

By default, Cronbach's alphas < 0.7 and Loevinger's H coefficients < 0.3 are displayed in red. These thresholds can be changed with the options `alpha(#)` and `h(#)`, respectively.

The descitems option

```
. validscale ioc1-ioc37, partition(4 4 7 3 3 4 7 5)
> scorename(HA PSE W BCC AC AE LI MOC) compscore(sum) descitems
Items used to compute the scores
```

(*output omitted*)

Description of items

	Missing	N	Response categories 1	2	3	4	5	Alpha - item	Hj	# of NS Hjk
ioc1	3.77%	357	10.08%	12.61%	24.65%	33.05%	19.61%	0.71	0.25	0
ioc2	1.08%	367	3.00%	8.72%	10.90%	39.78%	37.60%	0.52	0.42	0
ioc3	2.16%	363	2.48%	5.79%	11.02%	44.63%	36.09%	0.53	0.43	0
ioc4	2.43%	362	3.31%	8.56%	18.51%	43.09%	26.52%	0.62	0.33	0
ioc5	2.96%	360	9.44%	15.28%	22.78%	28.06%	24.44%	0.70	0.30	0
ioc6	2.96%	360	10.28%	15.28%	24.17%	33.61%	16.67%	0.54	0.47	0
ioc7	2.43%	362	4.97%	8.01%	22.10%	42.27%	22.65%	0.67	0.34	0
ioc8	2.16%	363	14.60%	19.83%	33.06%	20.66%	11.85%	0.58	0.44	0
ioc9	2.43%	362	15.47%	22.65%	14.64%	28.18%	19.06%	0.89	0.63	0
ioc10	3.23%	359	33.43%	27.58%	20.89%	12.26%	5.85%	0.90	0.59	0
ioc11	1.89%	364	5.49%	9.62%	13.74%	42.03%	29.12%	0.89	0.61	0
ioc12	3.23%	359	8.64%	18.94%	19.22%	37.05%	16.16%	0.89	0.63	0
ioc13	3.23%	359	13.65%	24.79%	18.11%	30.36%	13.09%	0.88	0.66	0
ioc14	1.62%	365	12.05%	26.30%	14.25%	28.49%	18.90%	0.89	0.60	0
ioc15	1.08%	367	6.81%	19.62%	18.26%	39.78%	15.53%	0.89	0.64	0
ioc16	1.35%	366	10.11%	21.86%	12.57%	30.33%	25.14%	0.67	0.62	0
ioc17	1.62%	365	9.86%	22.19%	13.97%	32.33%	21.64%	0.69	0.61	0
ioc18	1.08%	367	20.98%	34.88%	12.53%	21.80%	9.81%	0.78	0.58	0
ioc19	3.23%	359	14.76%	27.58%	19.50%	24.79%	13.37%	0.75	0.61	0
ioc20	2.70%	361	27.15%	29.36%	19.67%	14.68%	9.14%	0.69	0.66	0
ioc21	1.35%	366	20.22%	19.13%	16.39%	26.50%	17.76%	0.77	0.60	0
ioc22	1.08%	367	6.27%	12.26%	20.71%	45.50%	15.26%	0.71	0.34	0
ioc23	0.81%	368	1.63%	2.45%	5.71%	50.54%	39.67%	0.68	0.40	0
ioc24	1.89%	364	2.75%	8.52%	25.00%	42.03%	21.70%	0.55	0.52	0
ioc25	2.43%	362	5.25%	16.57%	33.98%	29.28%	14.92%	0.63	0.44	0
ioc26	2.16%	363	28.37%	38.84%	16.53%	9.64%	6.61%	0.82	0.29	0
ioc27	2.70%	361	28.53%	37.40%	11.08%	15.79%	7.20%	0.78	0.46	0
ioc28	1.89%	364	40.11%	35.71%	10.99%	7.97%	5.22%	0.78	0.48	0
ioc29	1.62%	365	33.42%	34.52%	11.23%	13.42%	7.40%	0.78	0.43	0
ioc30	1.35%	366	27.60%	31.15%	12.57%	19.13%	9.56%	0.79	0.43	0
ioc31	1.35%	366	36.34%	35.79%	10.93%	12.02%	4.92%	0.78	0.47	0
ioc32	2.96%	360	14.17%	17.22%	17.78%	33.89%	16.94%	0.81	0.35	0
ioc33	2.70%	361	8.59%	24.38%	23.27%	31.30%	12.47%	0.84	0.38	0
ioc34	2.96%	360	6.94%	26.39%	30.83%	24.72%	11.11%	0.78	0.54	0
ioc35	4.04%	356	13.48%	26.40%	28.65%	22.47%	8.99%	0.78	0.56	0
ioc36	3.50%	358	20.95%	32.68%	24.86%	15.64%	5.87%	0.79	0.54	0
ioc37	3.23%	359	18.66%	31.20%	22.84%	19.22%	8.36%	0.76	0.60	0

(*output omitted*)

The column `Alpha-item` shows the Cronbach's alpha coefficients of the scales when the considering item is removed. For example, removing `ioc1` from the first subscale (`HA`) would increase the alpha coefficient for this subscale from 0.67 (see previous table) to 0.71, whereas removing `ioc2` would decrease the alpha coefficient from 0.67 to 0.52.

The column `Hj` corresponds to the Loevinger's coefficients of item-scale consistency. We note that the first `Hj` coefficient of the column (0.25) corresponds to the value of $H^{\texttt{HA}}_{\texttt{ioc1}}$ displayed in the previous table.

The last column indicates the number of H_{jk} coefficients (Loevinger's coefficients between the considering item and each of the other items of the dimension) that are not significantly different from zero. If one H_{jk} coefficient would appear not to be significantly different from zero, it would indicate a problem in item-scale consistency.

The graph option

```
. validscale ioc1-ioc37, partition(4 4 7 3 3 4 7 5)
> scorename(HA PSE W BCC AC AE LI MOC) graph
```

(*output omitted*)

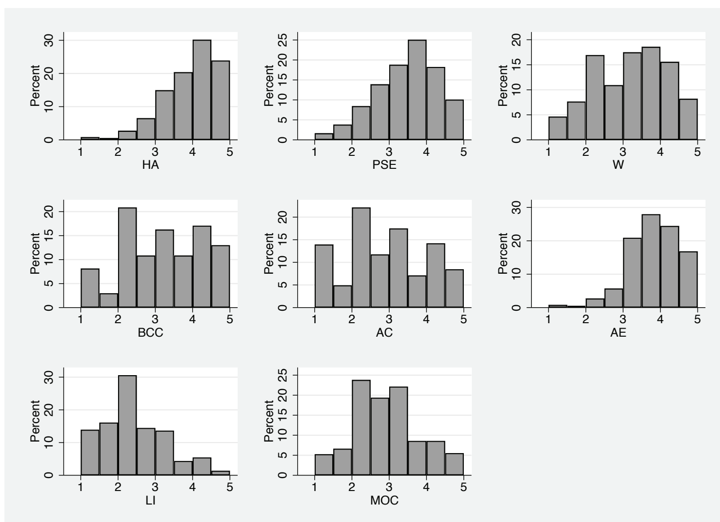

Figure 1. Histograms of scores

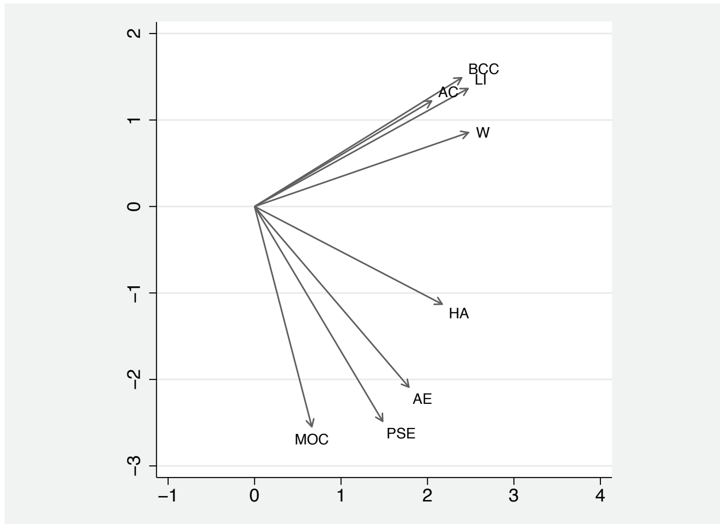

Figure 2. Correlations between scores

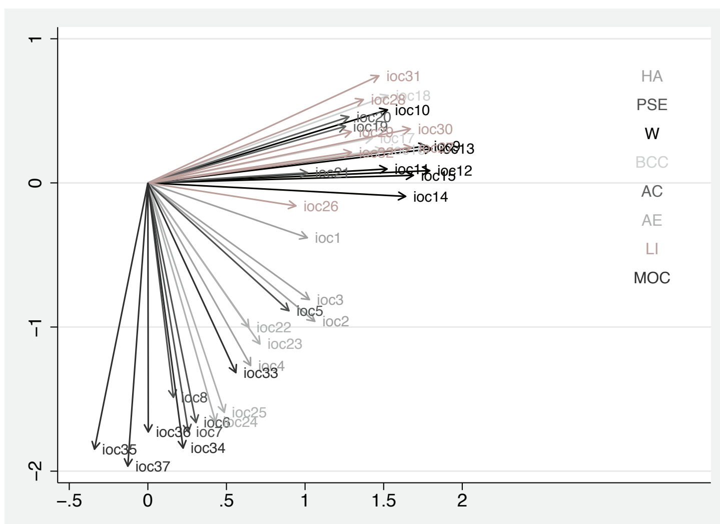

Figure 3. Correlations between items

The **graph** option produces histograms of scores (figure 1), a biplot[1] of the scores (figure 2), and a biplot of the items (figure 3).

The histogram allows one to examine the distribution of the scores, particularly to check normality and to identify potential floor or ceiling effects (that is, high proportion

1. Only variables are represented in the biplots.

of subjects scoring minimum score or maximum score, respectively). Figures 2 and 3 are produced with the `biplot` command. In figure 2, the cosine of the angle between arrows approximates the correlation between the scores. This allows one to graphically assess the correlation between the scores. For example, appearance concerns (`AC`), body change concerns (`BCC`), and life interferences (`LI`) seem very correlated with each other but uncorrelated with the meaning of cancer (`MOC`).

In figure 3, the cosine of the angle between arrows approximates the correlation between the items. Moreover, in figure 3, items within the same dimension are represented in the same color. Thus, we expect items of the same color to be close to each other. In this example, we note that item `ioc5`, although theoretically grouped with items `ioc6`, `ioc7`, and `ioc8`, seems more correlated with `ioc2` and `ioc3`.

The cfa option

```
. validscale ioc1-ioc37, partition(4 4 7 3 3 4 7 5)
> scorename(HA PSE W BCC AC AE LI MOC) cfa cfacov(ioc1*ioc3)
  (output omitted)
```

```
Confirmatory factor analysis

Warning: some items have less than 7 response categories. If multivariate
normality assumption does not hold, maximum likelihood estimation might not
be appropriate. Consider using cfasb in order to apply Satorra-Bentler
adjustment or using cfamethod(adf).

Covariances between errors added: e.ioc1*e.ioc3

Number of used individuals:  292
```

Item	Dimension	Factor loading	Standard error	Intercept	Standard error	Error variance	Variance of dimension
ioc1	HA	1.00	.	3.36	0.07	1.33	0.16
ioc2	HA	2.05	0.46	3.95	0.06	0.45	
ioc3	HA	1.53	0.31	4.01	0.06	0.55	
ioc4	HA	1.47	0.34	3.77	0.06	0.68	
ioc5	PSE	1.00	.	3.42	0.07	1.32	0.32
ioc6	PSE	1.56	0.24	3.27	0.07	0.69	
ioc7	PSE	1.15	0.20	3.70	0.06	0.66	
ioc8	PSE	1.37	0.22	2.91	0.07	0.80	
ioc9	W	1.00	.	3.06	0.08	0.62	1.19
ioc10	W	0.77	0.06	2.26	0.07	0.75	
ioc11	W	0.74	0.06	3.77	0.07	0.65	
ioc12	W	0.88	0.06	3.27	0.07	0.52	
ioc13	W	0.98	0.06	3.01	0.07	0.45	
ioc14	W	0.89	0.06	3.12	0.08	0.75	
ioc15	W	0.82	0.06	3.33	0.07	0.49	
ioc16	BCC	1.00	.	3.30	0.08	0.73	1.08
ioc17	BCC	0.92	0.07	3.30	0.07	0.68	
ioc18	BCC	0.90	0.08	2.62	0.08	0.76	
ioc19	AC	1.00	.	2.90	0.07	0.56	1.07
ioc20	AC	1.01	0.08	2.45	0.07	0.51	
ioc21	AC	0.92	0.08	2.91	0.08	1.07	

```
ioc22        AE    1.00          .          3.48       0.06    0.89          0.25
ioc23        AE    0.85       0.14          4.22       0.05    0.44
ioc24        AE    1.62       0.23          3.65       0.06    0.32
ioc25        AE    1.53       0.22          3.26       0.06    0.51

ioc26        LI    1.00          .          2.21       0.07    1.04          0.27
ioc27        LI    1.70       0.23          2.26       0.07    0.63
ioc28        LI    1.55       0.21          2.02       0.07    0.63
ioc29        LI    1.58       0.22          2.21       0.07    0.82
ioc30        LI    1.83       0.26          2.47       0.08    0.83
ioc31        LI    1.65       0.23          2.08       0.07    0.56
ioc32        LI    1.33       0.21          3.17       0.08    1.21

ioc33       MOC    1.00          .          3.06       0.07    1.09          0.29
ioc34       MOC    1.39       0.19          3.02       0.06    0.61
ioc35       MOC    1.66       0.22          2.84       0.07    0.52
ioc36       MOC    1.58       0.21          2.46       0.07    0.57
ioc37       MOC    2.01       0.26          2.61       0.07    0.29


Covariances between dimensions:
        HA    PSE      W    BCC     AC     AE     LI    MOC
 HA   0.16      .      .      .      .      .      .      .
PSE   0.11   0.32      .      .      .      .      .      .
  W   0.11   0.04   1.19      .      .      .      .      .
BCC   0.22   0.07   0.14   1.08      .      .      .      .
 AC   0.16  -0.02   0.65  -0.06   1.07      .      .      .
 AE   0.08   0.04   0.46   0.65  -0.04   0.25      .      .
 LI   0.10   0.17   0.09   0.06   0.06  -0.07   0.27      .
MOC   0.06   0.01   0.38   0.39   0.29   0.02   0.10   0.29

Goodness of fit:
   chi2            df     chi2/df      RMSEA [90% CI]        SRMR      NFI
1103.86            600          1.8    0.054 [0.049 ; 0.059]   0.074    0.796
(p-value = 0.000)
   RNI           CFI          IFI        MCI
  0.894         0.894        0.895    0.421
```

The `cfa` option uses the official Stata command `sem` to perform a CFA. The option `cfacov(ioc1*ioc3)` allows one to consider a covariance between the errors of `ioc1` and `ioc3`.

Goodness-of-fit indices computed by Stata or from Gadelrab (2010) are given below the tables of estimates.

In this example, the RMSEA is < 0.06, which indicates acceptable fit. However, the CFI is only 0.89. To improve the CFI, we could specify direct effects between some items by using `cfacov()` or `cfacfi()` (see Options for details).

A warning is displayed because items have only five response categories. In that case, we could use `cfasb` to apply Satorra–Bentler adjustment after maximum likelihood estimation.

The convdiv option

```
. validscale ioc1-ioc37, partition(4 4 7 3 3 4 7 5)
> scorename(HA PSE W BCC AC AE LI MOC) convdiv convdivboxplot
Items used to compute the scores
```

(*output omitted*)

Correlation matrix

	HA	PSE	W	BCC	AC	AE	LI	MOC
ioc1	0.266	0.171	0.319	0.278	0.262	0.208	0.243	0.093
ioc2	0.535	0.343	0.382	0.206	0.102	0.363	0.202	0.122
ioc3	0.536	0.346	0.306	0.265	0.220	0.258	0.196	0.106
ioc4	0.403	0.274	0.229	0.064	0.002	0.332	0.132	0.266
ioc5	0.359	0.362	0.308	0.172	0.107	0.242	0.174	0.126
ioc6	0.296	0.609	0.083	0.013	0.084	0.391	0.077	0.286
ioc7	0.316	0.418	0.102	-0.007	0.052	0.423	-0.011	0.382
ioc8	0.157	0.546	0.024	0.002	0.077	0.321	0.046	0.253
ioc9	0.364	0.181	0.743	0.359	0.280	0.184	0.530	-0.024
ioc10	0.216	0.072	0.643	0.364	0.258	0.057	0.466	-0.114
ioc11	0.394	0.045	0.666	0.335	0.156	0.166	0.408	-0.010
ioc12	0.411	0.163	0.734	0.433	0.327	0.230	0.476	-0.020
ioc13	0.318	0.165	0.792	0.389	0.323	0.180	0.524	-0.027
ioc14	0.390	0.191	0.698	0.378	0.272	0.186	0.425	0.062
ioc15	0.427	0.145	0.741	0.424	0.285	0.185	0.431	0.011
ioc16	0.268	0.023	0.421	0.673	0.352	0.174	0.546	0.046
ioc17	0.237	0.057	0.344	0.650	0.477	0.138	0.474	-0.033
ioc18	0.233	0.076	0.451	0.565	0.451	0.049	0.509	-0.068
ioc19	0.176	0.042	0.330	0.508	0.647	0.146	0.376	-0.056
ioc20	0.160	0.072	0.327	0.454	0.701	0.143	0.414	-0.109
ioc21	0.184	0.147	0.258	0.310	0.620	0.184	0.338	-0.025
ioc22	0.271	0.310	0.225	0.137	0.170	0.389	0.216	0.142
ioc23	0.373	0.321	0.222	0.123	0.083	0.435	0.157	0.158
ioc24	0.287	0.391	0.119	0.041	0.083	0.643	0.016	0.271
ioc25	0.287	0.394	0.057	0.127	0.185	0.527	0.053	0.292
ioc26	0.149	0.098	0.235	0.324	0.214	0.171	0.386	0.107
ioc27	0.277	0.162	0.552	0.413	0.316	0.167	0.635	-0.041
ioc28	0.190	-0.000	0.398	0.352	0.323	0.035	0.648	-0.080
ioc29	0.171	0.062	0.344	0.391	0.241	0.048	0.583	0.038
ioc30	0.220	0.147	0.583	0.414	0.309	0.146	0.573	-0.024
ioc31	0.182	0.014	0.396	0.472	0.348	0.048	0.630	-0.053
ioc32	0.201	0.095	0.333	0.558	0.365	0.130	0.416	0.063
ioc33	0.253	0.266	0.184	0.117	0.044	0.280	0.210	0.433
ioc34	0.231	0.322	0.053	0.010	0.004	0.199	0.026	0.652
ioc35	0.082	0.250	-0.144	-0.124	-0.181	0.175	-0.118	0.673
ioc36	0.117	0.277	-0.053	-0.012	-0.044	0.225	-0.002	0.637
ioc37	0.111	0.312	-0.115	-0.007	-0.047	0.235	-0.046	0.748

```
Convergent validity: 33/37 items (89.2%) have a correlation coefficient with
                     the score of their own dimension greater than 0.400
```

```
Divergent validity:  33/37 items (89.2%) have a correlation coefficient with
                     the score of their own dimension greater than those
                     computed with other scores.
```

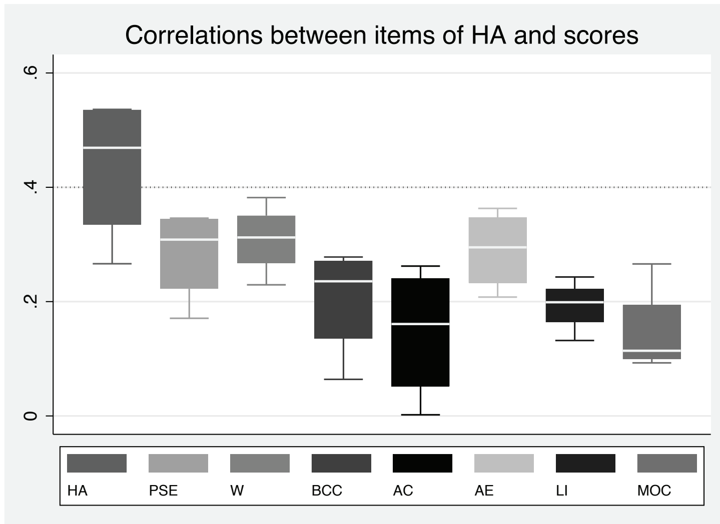

Figure 4. Correlations between items of `HA` and scores

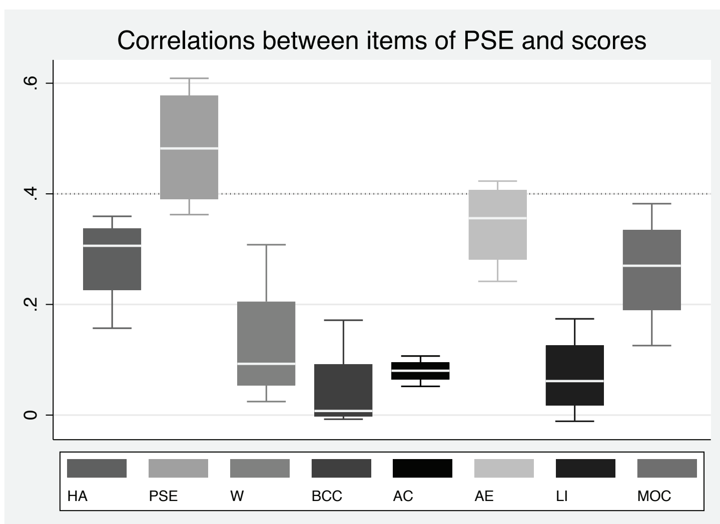

Figure 5. Correlations between items of `PSE` and scores

`convdiv` assesses convergent and divergent (discriminant) validities through the examination of a correlation matrix. The elements of this matrix are the correlation coefficients between items and rest scores.

Elements on the diagonal (correlations between an item and the rest score of its dimension) are displayed in bold font in Stata. On the diagonal, values less than `tconvdiv(#)` (0.4 by default) are displayed in red, indicating lack of convergent validity. For each row, off-diagonal values greater than values on the diagonal are displayed in red, indicating lack of divergent validity. In this example, the value of the correlation coefficient between `ioc1` and `HA` would be displayed in red because $0.266 < 0.4$. The value of the correlation coefficient between `ioc1` and `W` and between `ioc1` and `BCC` would also be displayed in red because $0.319 > 0.266$ and $0.278 > 0.266$, respectively.

The `convdivboxplot` option produces boxplots representing the values of the correlation matrix. In this example, because there are eight subscales, eight graphs composed of eight boxes are generated. Figures 4 and 5 correspond to two of these graphs (the six remaining graphs are not shown). In figure 4, we expect the first boxplot to be the "highest" in the graph because it represents the correlations between `HA` and its own items. We also expect these correlations to be ≥ 4. In figure 5, we expect the second boxplot to be higher than others because it represents the correlations between `PSE` and its own items.

The repet() option

```
. set seed 1234
. foreach v of varlist ioc1-ioc37 {
  2.         generate `v´_2 = round(rnormal(`v´,0.5))
  3. }
  (output omitted)
. validscale ioc1-ioc37, partition(4 4 7 3 3 4 7 5)
> scorename(HA PSE W BCC AC AE LI MOC) repet(ioc1_2-ioc37_2) kappa ickappa(500)
  (output omitted)
```

Reproducibility

Dimension	n	Item	Kappa	95% CI for Kappa (bootstrapped)	ICC	95% CI for ICC
HA	368	ioc1	0.59	[0.52 ; 0.66]	0.96	[0.96 ; 0.97]
		ioc2	0.53	[0.46 ; 0.59]		
		ioc3	0.52	[0.45 ; 0.58]		
		ioc4	0.57	[0.50 ; 0.64]		
PSE	367	ioc5	0.62	[0.46 ; 0.59]	0.97	[0.97 ; 0.98]
		ioc6	0.60	[0.54 ; 0.66]		
		ioc7	0.57	[0.51 ; 0.64]		
		ioc8	0.62	[0.57 ; 0.69]		
W	366	ioc9	0.62	[0.45 ; 0.58]	0.99	[0.98 ; 0.99]
		ioc10	0.56	[0.51 ; 0.63]		
		ioc11	0.53	[0.46 ; 0.59]		
		ioc12	0.59	[0.53 ; 0.65]		
		ioc13	0.61	[0.55 ; 0.67]		
		ioc14	0.56	[0.51 ; 0.62]		
		ioc15	0.56	[0.49 ; 0.61]		
BCC	369	ioc16	0.57	[0.50 ; 0.64]	0.98	[0.97 ; 0.98]
		ioc17	0.56	[0.49 ; 0.61]		
		ioc18	0.60	[0.54 ; 0.66]		
AC	366	ioc19	0.57	[0.55 ; 0.67]	0.98	[0.97 ; 0.98]
		ioc20	0.54	[0.48 ; 0.61]		
		ioc21	0.55	[0.49 ; 0.61]		
AE	368	ioc22	0.54	[0.54 ; 0.66]	0.96	[0.96 ; 0.97]
		ioc23	0.52	[0.46 ; 0.59]		
		ioc24	0.59	[0.52 ; 0.65]		
		ioc25	0.58	[0.51 ; 0.64]		
LI	366	ioc26	0.63	[0.51 ; 0.64]	0.98	[0.98 ; 0.99]
		ioc27	0.58	[0.52 ; 0.64]		
		ioc28	0.56	[0.50 ; 0.63]		
		ioc29	0.60	[0.55 ; 0.67]		
		ioc30	0.61	[0.54 ; 0.66]		
		ioc31	0.56	[0.48 ; 0.61]		
		ioc32	0.57	[0.51 ; 0.63]		
MOC	361	ioc33	0.62	[0.57 ; 0.69]	0.98	[0.97 ; 0.98]
		ioc34	0.57	[0.52 ; 0.64]		
		ioc35	0.57	[0.51 ; 0.63]		
		ioc36	0.63	[0.56 ; 0.69]		
		ioc37	0.59	[0.53 ; 0.65]		

The `repet()` option computes ICC and its corresponding confidence intervals with the Stata command `icc` to assess the reproducibility of scores. *varlist* is the list of items measured at time 2.

The `kappa` option computes kappa statistics with the Stata command `kap` to assess the reproducibility of items. Computation of confidence intervals for kappa coefficients are possible with the `ickappa()` option based on the community-contributed command `kapci`.

If the `scores()` option is used, one would probably want to also use the `scores2()` option to indicate the variables corresponding to the scores computed at time 2. In that case, ICCs are based on these scores rather than on a combination of items, as defined by `compscore()`.

Because there was only one time of measurement for this questionnaire, responses to items `ioc1-ioc37` at the second time of measurement have been simulated for the sake of this example.

Kappas > 0.6 would indicate good reproducibility of items, and ICCs > 0.75 would indicate good reproducibility of scores.

The kgv() option

```
. validscale ioc1-ioc37, partition(4 4 7 3 3 4 7 5)
> scorename(HA PSE W BCC AC AE LI MOC) kgv(chemo) kgvboxplot kgvgroup
  (output omitted)
------------------------------------------------------------------------------
Known-groups validity
------------------------------------------------------------------------------

        chemo
                    mean         sd    p-value
HA      0 (n=106)    3.71       0.76      0.101  (KW: 0.060)
        1 (n=245)    3.85       0.76

PSE     0 (n=105)    3.20       0.85      0.042  (KW: 0.029)
        1 (n=245)    3.40       0.85

W       0 (n=105)    3.10       0.90      0.535  (KW: 0.471)
        1 (n=244)    3.17       1.01

BCC     0 (n=105)    2.87       1.13      0.009  (KW: 0.011)
        1 (n=247)    3.20       1.06

AC      0 (n=105)    2.58       1.10      0.011  (KW: 0.014)
        1 (n=245)    2.91       1.13

AE      0 (n=104)    3.62       0.65      0.187  (KW: 0.095)
        1 (n=247)    3.73       0.74

LI      0 (n=104)    2.29       0.80      0.157  (KW: 0.215)
        1 (n=246)    2.42       0.85

MOC     0 (n=103)    2.76       0.83      0.213  (KW: 0.190)
        1 (n=242)    2.90       0.93
```

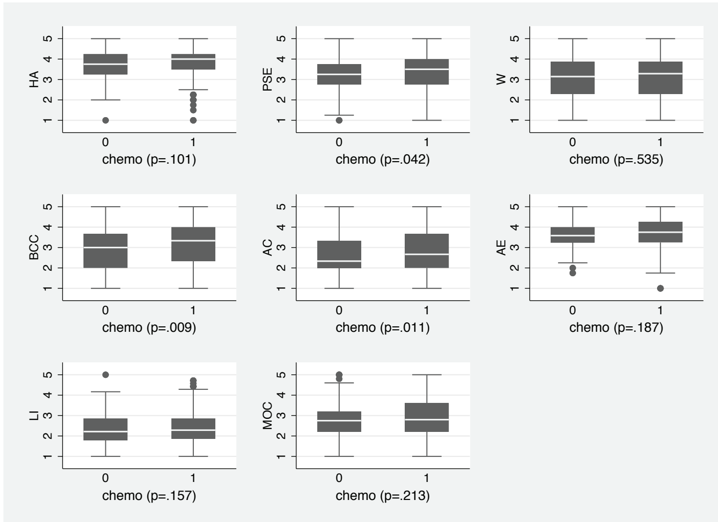

Figure 6. Known-groups validity: chemotherapy/no chemotherapy

The `kgv()` option assesses the known-groups validity of the questionnaire. In this example, an ANOVA is performed to compare the scores between women who had chemotherapy and those who had not.

A Kruskal–Wallis test is also performed in case ANOVA assumptions were not met.

We can see that respondents who received chemotherapy scored higher than others on the `BCC` (body change concerns) dimension (3.20 versus 2.87, $p = 0.009$).

`kgvboxplots` produces boxplots of scores by the group variable; `kgvgroupboxplots` groups the boxplots into a single graph (figure 6).

The conc() option

```
. validscale ioc1-ioc37, partition(4 4 7 3 3 4 7 5)
> scorename(HA PSE W BCC AC AE LI MOC) conc(sf12mcs sf12pcs)
  (output omitted)
─────────────────────────────────────────────────────────────────────────────
Concurrent validity
─────────────────────────────────────────────────────────────────────────────

        sf12mcs    sf12pcs
HA     -0.17      -0.14
PSE    -0.04      -0.10
W      -0.44      -0.21
BCC    -0.48      -0.44
AC     -0.26      -0.15
AE     -0.16      -0.07
LI     -0.49      -0.42
MOC     0.12      -0.00
```

In this example, the conc() option assesses the concurrent validity by examining the correlations between the scores of the Impact Of Cancer version 2 and the SF-12 (Ware, Kosinski, and Keller 1996). Correlations coefficients ≤ -0.40 or ≥ 0.40 are displayed in bold font in Stata. This threshold can be changed with tconc(#).

For instance, the worry (W) score is (negatively) correlated with the mental component summary score (sf12mcs) of the SF-12 ($\rho = -0.44$) but only moderately correlated with the physical component summary score (sf12pcs) of the SF-12 ($\rho = -0.21$).

4.3 Complex syntax of validscale

An example of a complex syntax with most of the available options is given below.

```
validscale ioc1-ioc37, partition(4 4 7 3 3 4 7 5)                  ///
  scorename(HA PSE W BCC AC AE LI MOC) categories(1 5) impute(pms)   ///
  noround compscore(sum) descitems graphs cfa cfamethod(ml) cfastand ///
  cfacov(ioc1*ioc3 ioc2*ioc4) convdiv tconvdiv(0.4) convdivboxplots  ///
  alpha(0.7) delta(0.9) h(0.3) hjmin(0.3) repet(ioc1_2-ioc37_2)      ///
  kappa ickappa(500) kgv(chemo radio) kgvboxplots kgvgroupboxplots   ///
  conc(sf12mcs sf12pcs) tconc(0.4)
```

5 Implementation of validscale

5.1 Dialog box

A dialog box is available if you want to use validscale more intuitively. The db validscale command displays the dialog box shown in figure 7 (only the first tab is shown here).

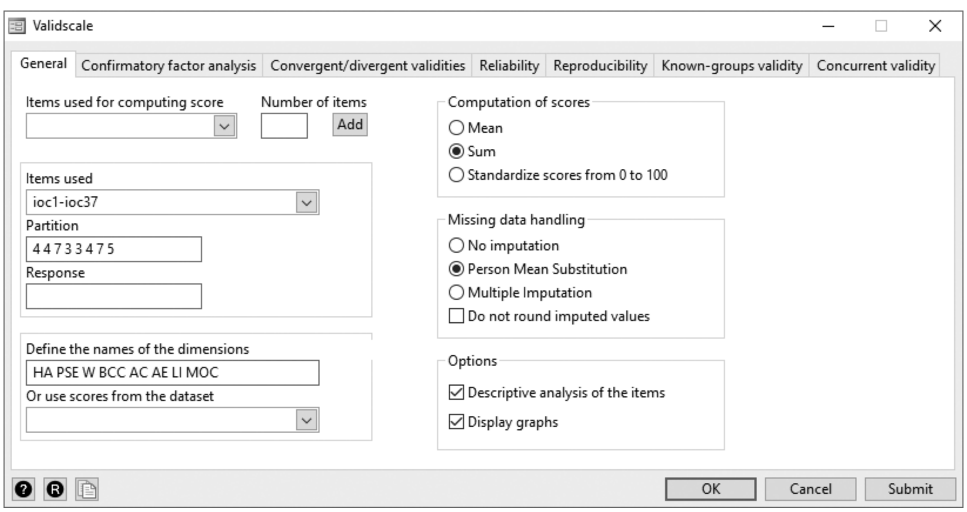

Figure 7. Dialog box for validscale

5.2 Online implementation with Numerics by Stata

`validscale` will soon be implemented as a command in an online application called the PRO-online project, which is dedicated to the analysis of patient-reported outcomes. The PRO-online project aims at proposing to students or researchers a way to perform analyses in the field of classical test or item response theories on their own data in a user-friendly way without complex handling of the data. Analyses are performed with Numerics by Stata.

6 Conclusion

`validscale` is a command performing analyses to assess the psychometric properties of subjective measurement scales in the framework of CTT in a user-friendly way. It provides information on structural validity, convergent and discriminant validities, reproducibility, known-groups validity, internal consistency, and scalability.

Other theories of measurement coexist in psychometry, particularly IRT and RMT. In these two theories, a latent construct is defined from the relationships between items. Stata provides several commands for IRT (by using the **irt** command). RMT can be used in Stata by using the community-contributed command **raschtest** (Hardouin 2007c) for dichotomous items or `pcmodel` (Hamel et al. 2016) for polytomous items. Finally, `validscale` can be used in a user-friendly way with a dialog box and will soon be implemented in an online application with Numerics by Stata.

7 References

Blanchette, D. 2010. lstrfun: Stata module to modify long local macros. Statistical Software Components S457169, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s457169.html.

Crespi, C. M., P. A. Ganz, L. Petersen, A. Castillo, and B. Caan. 2008. Refinement and psychometric evaluation of the impact of cancer scale. *Journal of the National Cancer Institute* 100: 1530–1541.

Fayers, P., and D. Machin. 2007. *Quality of Life: The Assessment, Analysis and Interpretation of Patient-Reported Outcomes*. 2nd ed. Chichester, UK: Wiley.

Gadelrab, H. 2010. *Evaluating the Fit of Structural Equation Models: Sensitivity to Specification Error and Descriptive Goodness-of-Fit Indices*. Saarbrücken, Germany: Lambert Academic Publishing.

Hamel, J.-F. 2014. mi_twoway: Stata module for computing scores on questionnaires containing missing item responses. Statistical Software Components S457901, Boston College Department of Economics. https://ideas.repec.org/c/boc/bocode/s457901.html.

Hamel, J.-F., V. Sébille, G. Challet-Bouju, and J.-B. Hardouin. 2016. Partial credit model: Estimations and tests of fit with pcmodel. *Stata Journal* 16: 464–481.

Hardouin, J.-B. 2004. loevh: Stata module to compute Guttman errors and Loevinger *H* coefficients. Statistical Software Components S439401, Department of Economics, Boston College. http://ideas.repec.org/c/boc/bocode/s439401.html.

———. 2007a. delta: Stata module to compute the Delta index of scale discrimination. Statistical Software Components S456851, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s456851.html.

———. 2007b. imputeitems: Stata module to impute missing data of binary items. Statistical Software Components S456807, Department of Economics, Boston College. https://econpapers.repec.org/software/bocbocode/s456807.htm.

———. 2007c. Rasch analysis: Estimation and tests with raschtest. *Stata Journal* 7: 22–44.

Hardouin, J.-B., A. Bonnaud-Antignac, and V. Sébille. 2011. Nonparametric item response theory using Stata. *Stata Journal* 11: 30–51.

Reichenheim, M. E. 2004. Confidence intervals for the kappa statistic. *Stata Journal* 4: 421–428.

Ware, J. E., Jr., M. Kosinski, and S. D. Keller. 1996. A 12-item short-form health survey: Construction of scales and preliminary tests of reliability and validity. *Medical care* 34: 220–233.

About the authors

Bastien Perrot is a PhD student in biostatistics at the SPHERE U1246 "methodS in Patient-centered outcomes and HEalth ResEarch" unit at the University of Nantes, France. Part of his research deals with developing and validating measurements scales with applications in health.

Emmanuelle Bataille is a biostatistician at the SPHERE U1246 "methodS in Patient-centered outcomes and HEalth ResEarch" unit at the University of Nantes, France.

Jean-Benoit Hardouin is a teacher–researcher in biostatistics at the SPHERE U1246 "methodS in Patient-centered outcomes and HEalth ResEarch" unit at the University of Nantes, France. His research deals with psychometrics and especially with applying item response theory in clinical trials.