Journal of Statistical Software

March 2019, Volume 88, Issue 8. doi: 10.18637/jss.v088.i08

plssem: A Stata Package for Structural Equation Modeling with Partial Least Squares

Sergio Venturini
Università Bocconi

Mehmet Mehmetoglu
Norwegian University of Science and Technology

Abstract

We provide a package called **plssem** that fits partial least squares structural equation models, which is often considered an alternative to the commonly known covariance-based structural equation modeling. **plssem** is developed in line with the algorithm provided by Wold (1975) and Lohmöller (1989). To demonstrate its features, we present an empirical application on the relationship between perception of self-attractiveness and two specific types of motivations for working out using a real-life data set. In the paper we also show that, in line with other software performing structural equation modeling, **plssem** can be used for putting in relation single-item observed variables too and not only for latent variable modeling.

Keywords: factor analysis, latent variables, partial least squares, PLS, PLS-PM, PLS-SEM, path models, Stata, structural equation modeling, SEM.

1. Introduction

The traditional statistical techniques (e.g., linear regression, logistic regression, multilevel regression, etc.) are used to estimate models representing the relationship between one or more than one independent variable and a single dependent variable. The independent and dependent variables in these models are all measured using single items such as income, height, weight, length of education and so on. Following this reasoning, we can refer to these traditional statistical approaches as single-equation techniques containing single-item variables both on the left-hand side (dependent) and right-hand side (independent) of the equation. Typically, these methods are employed in the social sciences to explain and predict quantities of interest.

Structural equation modeling (SEM) too can be used for explanation and prediction purposes in the social sciences. The difference, and accordingly the advantage of SEM over single-

equation techniques, is that SEM allows for estimating the relationship between a number of independent variables and more than one dependent variable at the same time. Furthermore, while the traditional techniques such as regression analysis lets one only use single-item variables, SEM allows for use of multi-item independent and dependent variables.

As such, in a broader sense, we can refer to SEM as a simultaneous multiple-equation technique estimating models including single or/and multi-item variables on both sides of the equations. This broader definition reflects also the reason why in the course of the past four decades SEM has become probably the most popular statistical estimation technique in the social sciences. The approach to incorporating the multi-item variables in SEM has basically led to the development of two different methods: covariance-based structural equation modeling (COV-SEM) introduced by Jöreskog (1969), and variance-based structural equation modeling (VAR-SEM) proposed by Wold (1975). While in COV-SEM the paths between common factors are examined, in VAR-SEM the paths between weighted composites (replacing the common factors) are estimated. This implies that in COV-SEM, multi-item variables are incorporated into the model using the factor analytic technique whereas in VAR-SEM weighted composites are generated from multi-item variables.

In a nutshell, we can view COV-SEM as the factor-based and VAR-SEM as the component-based structural equation modeling methods (Chin 1995). COV-SEM and VAR-SEM are commonly referred to in the literature respectively as maximum likelihood SEM (ML-SEM; see for example Bollen 1989; Kline 2016), which is typically associated with software packages such as **LISREL** (Jöreskog and Sörbom 2015), **EQS** (Bentler 2008), **AMOS** (Arbuckle 2014) or **Mplus** (Muthén and Muthén 2017), and partial least squares (PLS-SEM or PLS-PM; see for example Esposito Vinzi, Trinchera, and Amato 2010), which are instead associated with the software packages **SmartPLS** (Ringle, Wende, and Becker 2015) or **XLSTAT** (Addinsoft 2007).

Although there is an ongoing debate as to the strengths and weaknesses of COV-SEM and PLS-SEM in the literature (see for example Rönkkö and Evermann 2013; Henseler, Dijkstra, Sarstedt, Ringle, Diamantopoulos, Straub, Ketchen, Hair, Hult, and Calantone 2014), there still appears to be a general consensus that these two approaches should be considered complementary rather than alternatives to each other. In line with this observation, Hair, Hult, Ringle, and Sarstedt (2017, p. 23) suggest PLS-SEM be used when:

- the goal is predicting key target constructs;
- formatively measured constructs are part of the structural model;
- the structural model is complex including many indicators/constructs;
- the sample size is small;
- the plan is to use latent variable scores in further analyses.

For more details on the pros and cons of the PLS-SEM approach versus COV-SEM we refer the reader to Hair *et al.* (2017).

In this paper we present the **plssem** package for Stata (StataCorp 2017). The aim of the package is to provide an open-source implementation of the PLS-SEM methodology for Stata. To the best of our knowledge, the only package currently available for fitting PLS-SEM models in Stata is the user-contributed **pls** package developed by Rönkkö (2016). However, as

indicated in the package's documentation, **pls** is provided for educational purposes since it only calculates composite variables and it does not produce any other output as well as it does not allow for any further postestimation analysis of a PLS-SEM fitted model. Essentially, we started from the **pls** code as a basis for the development of our **plssem** package, but then we fully redesigned and enhanced it with numerous additional output and tools for postestimation.

At the time of writing, PLS-SEM is not supported by any of the most popular commercial statistical software packages like SAS (SAS Institute Inc. 2013) or SPSS (IBM Corporation 2017), which only support PLS regression[1]. However, many open-source and commercial software packages have been developed independently over the years for fitting PLS-SEM models. Currently, the most widespread open-source implementations of the PLS-SEM methodology are all for the R software (R Core Team 2019), in particular **matrixpls** (Rönkkö 2017), **plspm** (Sanchez, Trinchera, and Russolillo 2017) and **semPLS** (Monecke and Leisch 2012). For what regards the commercial packages, the most popular ones are **SmartPLS** and **XLSTAT-PLSPM**[2]. We now provide a brief overview for each of them. A further now dated comparison of PLS path modeling software is also available in Temme, Kreis, and Hildebrandt (2010).

matrixpls in R: The **matrixpls** package implements a collection of PLS techniques as well as the more recent generalized structured component analysis (GSCA) introduced by Hwang and Takane (2004) (for a detailed presentation see Hwang and Takane 2014) and consistent partial least squares (PLSc) techniques as discussed in Dijkstra and Henseler (2015a,b). The variance of the PLS results is estimated using the bootstrap approach (Davison and Hinkley 1997) through the `matrixpls.boot()` function, which provides the integration with the **boot** package (Canty and Ripley 2017). **matrixpls** is the most recent addition in the set of R packages for PLS-SEM and, in contrast to all the other software packages for the same purpose which work with raw data, it calculates the indicator weights and model estimates from data covariance matrices. The main function, `matrixpls()`, requires that the model specification is performed by providing the list of user-defined adjacency matrices specifying the association between the different variables. Additional functions for postestimation (predictions, residual analysis, model quality indices) are also provided. No method is provided in the package to deal with observed heterogeneity such as multigroup analysis (MGA).

plspm in R: This is an R package developed by Sanchez *et al.* (2017) and dedicated to PLS-SEM analysis. The package comes with a number of functions to perform a series of different types of analysis including bootstrapping. The main function has the same name as the package, `plspm()`, which is designed for running a full PLS-SEM analysis. The package includes also some accessory functions for plotting and displaying results. Additionally, the function `plspm.groups()` allows to compare two groups (multigroup analysis) and it offers two options for doing the comparison, a bootstrap t test and a non-parametric permutation test. Finally, the package also includes a set of functions for the

[1]PLS regression should not be confused with PLS-SEM: the former is a multivariate regression method that maximizes the covariance between dependent and independent variables, which is today most widely used in chemometrics and related areas (Wehrens 2011; Mevik, Wehrens, and Liland 2018); the latter is a path modeling approach which can be considered an alternative to more traditional covariance-based structural equation modeling.

[2]Other commercial packages are also available, such as **ADANCO** (Henseler 2017) and **WarpPLS** (Kock 2018), but these are less popular than **SmartPLS** and **XLSTAT-PLSPM**.

detection of latent classes by using the REBUS-PLS approach for uncovering unobserved heterogeneity in PLS-SEM models (Trinchera 2007; Esposito Vinzi, Trinchera, Squillacciotti, and Tenenhaus 2008). As for **matrixpls**, model specification occurs through user-defined adjacency matrices. A book-length description of the package is provided in Sanchez (2013).

semPLS in R: This is a further package developed by Monecke and Leisch (2012) for structural equation modeling with partial least squares in R. The `plsm()` function is used to create a valid model specification (a so called '`plsm`' object), while `sempls()` fits the model. Models can be specified by providing the user-defined adjacency matrices. Bootstrapping is available too by leveraging the **boot** package (Canty and Ripley 2017) and the calculation of quality indices (R^2, Q^2, Dillon-Goldstein's ρ, etc.) is performed via specific methods. However, no method is provided in the package for dealing with observed (e.g., MGA) or unobserved heterogeneity (e.g., REBUS-PLS). A distinctive feature of **semPLS** is that it is possible to export '`plsm`' objects for use with the popular **sem** package (Fox, Nie, and Byrnes 2017). Similarly, it is possible to import model specification created with **SmartPLS** with the function `read.splsm()`.

SmartPLS: Now in its third official release (`http://www.smartpls.com`), it is a stand-alone commercial software supported by a community of scholars centered at the University of Hamburg (Germany), School of Business (Hair *et al.* 2017), which currently represents by far the most popular and comprehensive software implementation of the PLS-SEM methodology. Model specification is performed by drawing the structural model for the latent variables and by assigning the indicators to the latent variables through an easy to use GUI. **SmartPLS** provides state of the art PLS techniques for fitting PLS-SEM models including bootstrapping and nonlinear relationships. Both observed and unobserved heterogeneity can be accounted for using several approaches such as MGA and finite mixture (FIMIX) segmentation (Hahn, Johnson, Herrmann, and Huber 2002; Sarstedt, Becker, Ringle, and Schwaiger 2011). Finally, mediation and moderation (interaction effects) analysis are also available, as well as hierarchical component models (second-order models) for fitting more complex structural models.

XLSTAT-PLSPM: **XLSTAT** is a complete statistical add-in for Microsoft Excel developed by Addinsoft (`http://www.xlstat.com`). It is structured in modules that provide specialized suites of commands to analyze data in different fields (biomedical sciences, ecology, marketing, psychology, quality control and sensory analysis). **XLSTAT-PLSPM** is the module that provides the estimation of PLS path models. The package includes all the recent methodological features of the PLS-SEM approach. In particular, it provides bootstrapping but also MGA and REBUS-PLS for dealing with observed and unobserved heterogeneity respectively.

The **plssem** package for Stata presented in this manuscript includes the following features:

- Model specification using an equation-like style.
- Standard and bootstrap based estimation of PLS-SEM models.
- Mediation analysis through estimation and inference (including bootstrap) for up to five indirect effects.

- Moderation analysis through the inclusion of interactions among latent variables in the structural model specification; this provides an implementation of the so called *product indicator approach* (Sanchez 2013, Section 7.3).
- Possibility to fit models that include equations with binary dependent variables. To the best of our knowledge, none of the existing PLS-SEM software facilitates binary dependent variable estimation using maximum likelihood.
- Multigroup analysis of outer loadings and path coefficients for dealing with observed heterogeneity; in particular, it allows the comparison of an arbitrary number of groups using either normal-based, bootstrap or permutation tests.
- Potential to estimate *higher-order construct models* (sometimes also, maybe inappropriately, called *hierarchical models*; see for example Lohmöller 1989, Section 3.5).
- A range of graphical and postestimation commands for representing and inspecting the results of a fitted PLS-SEM model.

The **plssem** package is available through the Statistical Software Components (SSC) archive[3], often called the Boston College Archive, and it allows to fit various PLS-SEM models. To install the package one needs to execute the command

```
. ssc install plssem
```

which will download and copy all the ado, help and data files for the commands discussed here[4].

The rest of the paper is organized as follows: In Section 2 we present the main technical aspects of the PLS-SEM approach as well as the most common indicators discussed in the literature for assessing the quality of a fitted model. Section 3 provides an introduction to the **plssem** package. In particular, after discussing the general syntax, we provide a full description of the available options. Moreover, we present the different postestimation commands one can run after fitting a model and the objects that are saved during the estimation. These objects can clearly be used in subsequent analyses. In Section 4 we show some empirical applications of the PLS-SEM approach with the **plssem** package using two different data sets. Finally, Section 5 provides some closing thoughts and our plans for the future releases of the package. In the rest of the paper we adopt the same mathematical notation provided by Monecke and Leisch (2012, pp. 9–13).

2. The PLS-SEM methodology

As PLS-SEM resembles ML-SEM in many ways, it can be explained and illustrated using a slightly adjusted version of the LISREL terminology (Jöreskog, Olsson, and Wallentin 2016) and graphical notation used originally for ML-SEM. As depicted in Figure 1, a typical PLS-SEM model will consist of two parts: the measurement (or outer) and the structural (or inner) models.

[3]The SSC archive is hosted by http://www.RePEc.org/.

[4]Alternatively, the latest version of the package can be retrieved also from https://github.com/sergioventurini/plssem.

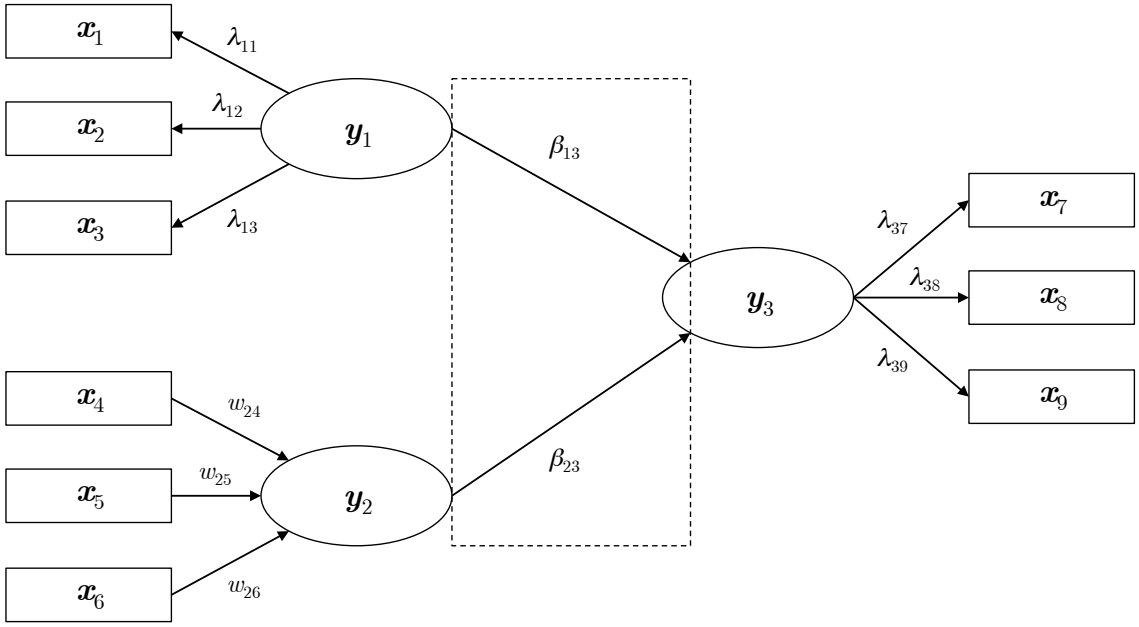

Figure 1: Graphical representation of a PLS-SEM model. Latent variables are displayed in ellipses and indicators (i.e., manifest variables) are displayed in boxes. Arrows pointing to indicators represent constructs measured in a reflective way ($\boldsymbol{y}_1$ and $\boldsymbol{y}_3$), while those going from indicators to latent variables ($\boldsymbol{y}_2$) correspond to constructs measured in a formative way. The dashed box highlights the structural part of the model.

The *measurement model* provides the relationships between latent variables (or constructs[5]) and the indicators they are defined by. The measurement part is represented in Figure 1 by all arrows apart from those included in the dashed box. The example includes two *reflective* (i.e., $\boldsymbol{y}_1$ and $\boldsymbol{y}_3$) and one *formative* (i.e., $\boldsymbol{y}_2$) construct. The association between the reflective constructs and the corresponding indicators (that is the arrows pointing from the constructs to the indicators) is indicated in the picture by $\lambda_{11}, \lambda_{12}, \lambda_{13}, \lambda_{37}, \lambda_{38}$ and λ_{39}, which are also called *outer loadings*. The relationship between the formative construct and the corresponding indicators (i.e., the arrows pointing from indicators to constructs) are denoted with w_{24}, w_{25} and w_{26} and are also referred to as *outer weights*. All indicators are *congeneric* in that none of them loads on more than one construct decided *a priori* (Brown 2015). The measurement model can be described by an adjacency matrix $\boldsymbol{M}$ whose entries m_{kj} take the value one if indicator $\boldsymbol{x}_k$ belongs to the block that defines latent variable $\boldsymbol{y}_j$, and zero otherwise, with $k = 1, \ldots, K$ and $j = 1, \ldots, J$. The adjacency matrix of the measurement model for the example shown in Figure 1 is provided in Table 1. Note that the matrix $\boldsymbol{M}$ does not convey any information about whether a construct is measured in a reflective or formative way.

The *structural model* shows the relationships between latent variables themselves. For the example shown in Figure 1 the structural model is represented by the arrows included in the dashed-line box. Latent variables in the structural model that are used as predictors are called *exogenous*, while those denoted as outcome variables are called *endogenous*. In our

[5]In the SEM literature latent variables or constructs are often related to multi-item variables used in factor-based SEM. However, as explained by Henseler *et al.* (2014, p. 3), one can also use these terms to refer to multi-item variables used in component-based SEM.

	$\boldsymbol{y}_1$	$\boldsymbol{y}_2$	$\boldsymbol{y}_3$
$\boldsymbol{x}_1$	1	0	0
$\boldsymbol{x}_2$	1	0	0
$\boldsymbol{x}_3$	1	0	0
$\boldsymbol{x}_4$	0	1	0
$\boldsymbol{x}_5$	0	1	0
$\boldsymbol{x}_6$	0	1	0
$\boldsymbol{x}_7$	0	0	1
$\boldsymbol{x}_8$	0	0	1
$\boldsymbol{x}_9$	0	0	1

Table 1: Measurement model adjacency matrix $\boldsymbol{M}$ for the example shown in Figure 1. The elements m_{kj} of the matrix are set to one if indicator $\boldsymbol{x}_k$ belongs to the block that defines latent variable $\boldsymbol{y}_j$, and zero otherwise.

	$\boldsymbol{y}_1$	$\boldsymbol{y}_2$	$\boldsymbol{y}_3$
$\boldsymbol{y}_1$	0	0	1
$\boldsymbol{y}_2$	0	0	1
$\boldsymbol{y}_3$	0	0	0

Table 2: Structural model adjacency matrix $\boldsymbol{S}$ for the example shown in Figure 1. The elements s_{ij} of the matrix are set to one if the latent variable $\boldsymbol{y}_i$ is a predecessor of the latent variable $\boldsymbol{y}_j$ in the model, and zero otherwise.

example there are two exogenous ($\boldsymbol{y}_1$ and $\boldsymbol{y}_2$) and one endogenous ($\boldsymbol{y}_3$) latent variable. The relationships among the latent variables are labeled using the corresponding *path coefficients* (β_{13} and β_{23}). The structural model can also be summarized by an adjacency matrix $\boldsymbol{S}$ whose entries s_{ij} take the value one if the latent variable $\boldsymbol{y}_i$ is a predecessor of the latent variable $\boldsymbol{y}_j$ in the model, and zero otherwise, with $i, j = 1, \ldots, J$. The adjacency matrix of the structural model for the example shown in Figure 1 is reported in Table 2. Note that matrix $\boldsymbol{S}$ allows to recover the information about whether a latent variable is exogenous or endogenous. More specifically, if the column corresponding to the latent variable $\boldsymbol{y}_j$ contains only zeros, that indicates that $\boldsymbol{y}_j$ is exogenous. In other words, contrary to the matrix $\boldsymbol{M}$ for the measurement model, $\boldsymbol{S}$ accounts for the directionality of the relationships among the latent variables.

To sum up the description of a PLS-SEM model, the structural part is similar to a regression model, while the measurement part resembles a factor or a principal component analysis. As such, PLS-SEM can be viewed as an advanced multivariate technique facilitating these two analyses at one go.

2.1. The PLS-SEM estimation algorithm

The algorithm used to estimate a PLS-SEM model consists basically of three sequential stages[6] (Lohmöller 1989). In the first stage, latent variable scores are iteratively estimated for each

[6]As it is common in the literature, we assume that both the latent variables and the indicators are standardized so that the location parameters can be discarded. If this is not the case, a fourth stage should be added in the algorithm described here corresponding to the estimation of the location parameters.

case. Using these scores, in the second stage measurement model parameters (weights/loadings) are estimated. In the same manner, in the third stage structural model parameters (path coefficients) are finally estimated. The first stage is what makes PLS-SEM a novel method in that the second and third stages, as it will be shown below, are about conducting a series of regression analysis using the ordinary least squares method. To help the reader grasp the whole process, we summarize the procedure for PLS-SEM estimation in Algorithm 1. We provide now more details on each stage.

Stage I – Iterative estimation of latent variable scores

The first stage is an iterative process consisting of the following steps, which are carried out to estimate the latent variable scores:

Step 0: Initialization of the latent variable scores.

Step 1: Estimation of the inner weights.

Step 2: Inner approximation of the latent variable scores.

Step 3: Estimation of the outer loadings/weights.

Step 4: Outer approximation of the latent variable scores.

Step 5: Convergence checking.

We now give a brief description of these steps. We will denote the data matrix including all the indicators as $\boldsymbol{X}$, and the block of indicators measuring the jth latent variable $\boldsymbol{y}_j$ as $\boldsymbol{X}_j$. Similarly, we indicate with $\boldsymbol{Y}$ the whole set of latent variables. As it is common in the SEM and PLS-SEM literature, we assume that prior to starting the entire process all indicators are standardized to have a mean zero and unit variance. Additionally, after each step the latent variables are scaled likewise.

Step 0: Initialization of the latent variable scores. In general, we estimate latent variable scores as a weighted sum of the indicators in the corresponding block. In the first step, each latent variable is initialized setting all weights equal to one. In other terms, initially we compute the scores as

$$\widehat{\boldsymbol{Y}} = \boldsymbol{X}\boldsymbol{M},$$

where $\boldsymbol{M}$ is the measurement model adjacency matrix presented in Section 2, such as that reported in Table 1.

Step 1: Estimation of the inner weights. Inner weights are calculated for each latent variable to reflect how strongly the other latent variables are connected to it. The three most common schemes used for computing the inner model weights are the *centroid* scheme originally proposed by Wold (1982), and the *factorial* and *path* schemes introduced by Lohmöller (1989). We provide below a brief description of these schemes assuming that we collect all the inner weights in a matrix denoted as $\boldsymbol{E}$.

Algorithm 1 The PLS-SEM estimation algorithm.

Let data $\boldsymbol{X}$ on indicators, measurement and structural model adjacency matrices $\boldsymbol{M}$ and $\boldsymbol{S}$ be given. Choose the latent variables measured in reflective (mode A) and formative (mode B) way. Set the outer weights initial values $\widehat{\boldsymbol{W}}^{\text{old}}$ to the zero matrix. Fix the tolerance *tol* and the maximum number of iterations $i_{\max}$.
Scale the indicators to have zero mean and unit variance.
Set the scores initial value to

$$\widehat{\boldsymbol{Y}} = \boldsymbol{X}\boldsymbol{M}.$$

Scale the latent variables scores to have zero mean and unit variance.
Set the iteration counter to zero ($i \leftarrow 0$) and the maximum relative difference of the outer weights δ to 1 ($\delta \leftarrow 1$).
while $\delta \geq tol$ and $i < i_{\max}$ **do**
    Estimate the inner weights using either Equations 1, 2 or 3 forming matrix $\boldsymbol{E}$.
    Compute the inner approximation of the latent variable scores as

$$\widetilde{\boldsymbol{Y}} = \widehat{\boldsymbol{Y}}\boldsymbol{E}.$$

    Scale the latent variables scores to have zero mean and unit variance.
    for $j \leftarrow 1, J$ **do**
        if $\boldsymbol{y}_j$ is in the set of mode A latent variables **then**
            Compute the outer weights as

$$\widehat{\boldsymbol{w}}_j^\top = \left(\widetilde{\boldsymbol{y}}_j^\top \widetilde{\boldsymbol{y}}_j\right)^{-1} \widetilde{\boldsymbol{y}}_j^\top \boldsymbol{X}_j.$$

        else if $\boldsymbol{y}_j$ is in the set of mode B latent variables **then**
            Compute the outer weights as

$$\widehat{\boldsymbol{w}}_j = \left(\boldsymbol{X}_j^\top \boldsymbol{X}_j\right)^{-1} \boldsymbol{X}_j^\top \widetilde{\boldsymbol{y}}_j.$$

        end if
    end for
    Compute the outer approximation of the latent variable scores as

$$\widehat{\boldsymbol{Y}} = \boldsymbol{X}\widehat{\boldsymbol{W}},$$

    where $\widehat{\boldsymbol{W}}$ is a diagonal matrix collecting the estimated weights $\widehat{\boldsymbol{w}}_j$, for $j = 1, \ldots, J$.
    Scale the latent variables scores to have zero mean and unit variance.
    Compute

$$\delta = \max_{\substack{k=1,\ldots,K \\ j=1,\ldots,J}} \left| \frac{\hat{w}_{kj}^{\text{old}} - \hat{w}_{kj}^{\text{new}}}{\hat{w}_{kj}^{\text{new}}} \right|.$$

    Increase the iteration counter ($i \leftarrow i + 1$).
end while

22: **for** $j \leftarrow 1$, J **do**
23: **if** $\boldsymbol{y}_j$ is in the set of mode A latent variables **then**
24: Compute the cross loadings as

$$\widehat{\boldsymbol{\lambda}}_j^{\text{cross}} = \mathsf{COR}(\boldsymbol{X}, \widehat{\boldsymbol{y}}_j).$$

25: Compute the outer loadings as

$$\widehat{\lambda}_{kj}^{\text{outer}} = \begin{cases} \widehat{\lambda}_{kj}^{\text{cross}} & \text{if } m_{kj} = 1 \\ 0 & \text{otherwise} \end{cases}.$$

26: **else if** $\boldsymbol{y}_j$ is in the set of mode B latent variables **then**
27: Compute the outer weights as

$$\widehat{\boldsymbol{w}}_j = \left(\boldsymbol{X}_j^\top \boldsymbol{X}_j\right)^{-1} \boldsymbol{X}_j^\top \widehat{\boldsymbol{y}}_j.$$

28: **end if**
29: Compute the structural model parameters (i.e., the path coefficients) as

$$\widehat{\boldsymbol{\beta}}_j = \left(\widehat{\boldsymbol{y}}_j^{\text{pred}\top} \widehat{\boldsymbol{y}}_j^{\text{pred}}\right)^{-1} \widehat{\boldsymbol{y}}_j^{\text{pred}\top} \widehat{\boldsymbol{y}}_j.$$

30: **end for**

Centroid scheme: This scheme produces weights e_{ij} based on the sign of $r_{ij} = \mathsf{COR}(\boldsymbol{y}_i, \boldsymbol{y}_j)$, the empirical linear correlation coefficient between the latent variables $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ resulting from the outer approximation (Step 4 below[7]), assuming they are neighbors. In particular, if $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are adjacent, the weight e_{ij} is set to $+1$ if the correlation is positive and to -1 if the correlation is negative. If $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are nonadjacent, e_{ij} is set to 0. More formally, for $i, j = 1, \ldots, J$,

$$e_{ij} = \begin{cases} \text{sign}\,(r_{ij}) & \text{if } c_{ij} = 1 \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where c_{ij} is the (i, j)th element of the matrix $\boldsymbol{C} = \boldsymbol{S} + \boldsymbol{S}^\top$, with $\boldsymbol{S}$ the adjacency matrix of the structural model introduced in Section 2. Thus, $\boldsymbol{C}$ is a symmetric matrix whose element c_{ij} takes value one if the latent variables $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are neighbors in the structural model, and zero otherwise.

Note that, as implied by Equation 1, correlations very close to zero may cause the weights to take a non-zero value during the iterative process, which may lead to instability. Thus, the centroid scheme should be used when the indicators of a block (latent variable) are strongly correlated to each other, otherwise the factorial scheme is usually recommended (Esposito Vinzi *et al.* 2010).

Factorial scheme: In this scheme the correlation value between each pair of latent variables

[7] At the first iteration of the algorithm the outer proxies of the latent variable scores correspond to the initial values computed in Step 0.

is directly used as the weight, that is

$$e_{ij} = \begin{cases} r_{ij} & \text{if } c_{ij} = 1 \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

with the same interpretation of the notation as above.

Path scheme: In this scheme two types of weight values are produced depending on the relationship between the latent variables. When a latent variable, say $\boldsymbol{y}_i$, is "causing" another latent variable $\boldsymbol{y}_j$ (the so called *successor*), the weight value corresponds to the linear correlation coefficient $r_{ij} = \mathsf{COR}(\boldsymbol{y}_i, \boldsymbol{y}_j)$. If instead the latent variable $\boldsymbol{y}_i$ is "caused" by another latent variable $\boldsymbol{y}_j$ (so called *predecessor*), the weight is determined using a multiple regression model. In particular, the estimated linear regression coefficient on the predecessor will then be used as the weight. More formally, according to the path scheme the weights are computed as follows

$$e_{ij} = \begin{cases} \hat{\gamma}_j & \text{for } j \in \boldsymbol{y}_i^{\text{pred}} \\ r_{ij} & \text{for } j \in \boldsymbol{y}_i^{\text{succ}} \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where $\boldsymbol{y}_i^{\text{pred}}$ indicates the set of predecessors of $\boldsymbol{y}_i$ and $\boldsymbol{y}_i^{\text{succ}}$ represents the corresponding set of successors. The coefficient $\hat{\gamma}_j$ corresponds to the estimate of the $\boldsymbol{y}_j$ coefficient in the linear regression model

$$\boldsymbol{y}_i = \boldsymbol{y}_i^{\text{pred}} \boldsymbol{\gamma} + \varepsilon_i,$$

assuming $\boldsymbol{y}_j$ belongs to the predecessor set of $\boldsymbol{y}_i$.

Step 2: Inner approximation of the latent variable scores. Here, we update the latent variable scores $\widehat{\boldsymbol{y}}_1, \ldots, \widehat{\boldsymbol{y}}_J$ obtained in the previous iteration with new ones, $\widetilde{\boldsymbol{y}}_1, \ldots, \widetilde{\boldsymbol{y}}_J$, which are computed as a weighted sum of their respective adjacent latent variables. More specifically, the inner approximation of the latent variable scores are computed as

$$\widetilde{\boldsymbol{Y}} = \widehat{\boldsymbol{Y}} \boldsymbol{E}, \tag{4}$$

where the matrix $\boldsymbol{E}$ contains the inner weights as obtained from Step 1.

Step 3: Estimation of the outer loadings/weights. So far we did not make any distinction between reflective and formative measures. On the contrary, we now need to take this difference into account to properly estimate the weights/loadings of the measurement model. That is, we need to recalculate the latent variables scores obtained from Step 2 using yet another weighting update. The new weights are called *loadings* when the latent variables are modeled as reflective and just *weights* when the latent variables are modeled as formative. In the classical algorithm, there are two possible choices for updating the outer weights, which are usually referred to as *mode A* and *mode B*. In the marketing literature, mode A refers to a reflective, while mode B refers to a formative measure.

In mode A, we regress each of the indicators onto the corresponding latent variable scores (i.e., using the latent variables included in the indicator block as the predictors of the regression

model). Since the latent variables from Step 2 are standardized, the regression coefficients do correspond to linear correlation coefficients, that is

$$\widehat{\boldsymbol{w}}_j^{\top} = \left(\widetilde{\boldsymbol{y}}_j^{\top}\widetilde{\boldsymbol{y}}_j\right)^{-1}\widetilde{\boldsymbol{y}}_j^{\top}\boldsymbol{X}_j = \mathsf{COR}(\widetilde{\boldsymbol{y}}_j, \boldsymbol{X}_j). \tag{5}$$

In mode B, we regress each latent variable against the indicators in its block. The weights will then correspond to the partial coefficients, that is

$$\widehat{\boldsymbol{w}}_j = \left(\boldsymbol{X}_j^{\top}\boldsymbol{X}_j\right)^{-1}\boldsymbol{X}_j^{\top}\widetilde{\boldsymbol{y}}_j = \mathsf{VAR}(\boldsymbol{X}_j)^{-1}\mathsf{COR}(\boldsymbol{X}_j, \widetilde{\boldsymbol{y}}_j). \tag{6}$$

Step 4: Outer approximation of the latent variable scores. In this step, we estimate the latent variable scores using the weights $\widehat{\boldsymbol{w}}_j$ obtained from Step 3 above by computing

$$\widehat{\boldsymbol{Y}} = \boldsymbol{X}\widehat{\boldsymbol{W}},$$

where $\widehat{\boldsymbol{W}}$ is the matrix that collects all the weights $\widehat{\boldsymbol{w}}_j$, that is

$$\widehat{\boldsymbol{W}} = \begin{pmatrix} \widehat{\boldsymbol{w}}_1 & 0 & \cdots & 0 \\ 0 & \widehat{\boldsymbol{w}}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \widehat{\boldsymbol{w}}_J \end{pmatrix}.$$

Step 5: Convergence checking. The process from Step 1 through Step 4 is then repeated until the maximum relative difference between the outer weights from one iteration to the next falls below a given tolerance value chosen by the analyst (e.g., 10^{-5}). More formally, the algorithm stops when

$$\max_{\substack{k=1,\ldots,K \\ j=1,\ldots,J}} \left| \frac{\hat{w}_{kj}^{\text{old}} - \hat{w}_{kj}^{\text{new}}}{\hat{w}_{kj}^{\text{new}}} \right| < tol.$$

Stage II – Estimation of measurement model parameters

Having estimated the latent variable scores, in the second stage of the PLS-SEM algorithm the loadings for reflective constructs and weights for formative constructs are calculated. These are actually those weights (Equations 5 and 6) at the final iteration. Alternatively, we can use the final latent variables scores ($\widehat{\boldsymbol{Y}}$) predicted after the PLS-SEM estimation to directly compute the loadings, as well as the cross loadings, as the linear correlation between $\boldsymbol{X}$ and $\widehat{\boldsymbol{Y}}$, and the weights by regressing $\widehat{\boldsymbol{Y}}$ on $\boldsymbol{X}$.

Stage III – Estimation of structural model parameters

In this stage, using the final latent variable scores, we estimate the structural model parameters (i.e., the path coefficients) for each endogenous latent variable using ordinary least squares according to the PLS-SEM model specified by the researcher. In particular, for each latent variable ($\widehat{\boldsymbol{y}}_j$) in the model, the path coefficients are obtained as the regression coefficients on its set of predecessors, denoted below as $\widehat{\boldsymbol{y}}_j^{\text{pred}}$, that is

$$\widehat{\boldsymbol{\beta}}_j = \left(\widehat{\boldsymbol{y}}_j^{\text{pred}\top}\widehat{\boldsymbol{y}}_j^{\text{pred}}\right)^{-1}\widehat{\boldsymbol{y}}_j^{\text{pred}\top}\widehat{\boldsymbol{y}}_j = \mathsf{COR}\left(\widehat{\boldsymbol{y}}_j^{\text{pred}}, \widehat{\boldsymbol{y}}_j^{\text{pred}}\right)^{-1}\mathsf{COR}\left(\widehat{\boldsymbol{y}}_j^{\text{pred}}, \widehat{\boldsymbol{y}}_j\right).$$

2.2. Bootstrap-based inference

Since PLS-SEM is a distribution-free method, it is not possible in general to get p values and confidence intervals for the model's parameters. For this reason, inference in PLS-SEM is usually conducted by relying on the (nonparametric) bootstrap (Davison and Hinkley 1997). In the literature on PLS-SEM (see for example Hair *et al.* 2017), the bootstrap is typically used to estimate the standard errors of the estimated parameters. For example, if one needs to test the null hypothesis that a certain outer weight w is equal to zero in the population versus a two-sided alternative, it is possible to calculate the corresponding test statistic by dividing the weight estimate $\widehat{w}$ based on the original full sample by its standard error estimated using the bootstrap. The test statistic value is then compared with the appropriate t distribution percentile to decide upon the rejection of the null hypothesis.

Bootstrap confidence intervals can be computed as well. Among the many approaches available for finding these intervals, it is usually suggested to use bias-corrected and accelerated bootstrap confidence intervals which adjust for biases and skewness in the bootstrap distribution (Henseler, Dijkstra, Sarstedt, Ringle, Diamantopoulos, Straub, Ketchen, Hair, Hult, and Calantone 2009; for a recent survey of the bootstrap methods see Efron and Hastie 2016, Chapter 11).

2.3. Communality, redundancy, goodness-of-fit, reliability coefficients

Assessment of the model goodness for a PLS-SEM model is rather complicated and not yet properly defined. However, many criteria have been proposed; some of them will be briefly presented below.

In addition to R^2 values, the quality of a PLS-SEM model can be assessed using the redundancy and goodness-of-fit (GoF) indices (Tenenhaus, Esposito Vinzi, Chatelin, and Lauro 2005, pp. 172–173).

To compute the average redundancy, we first need to estimate the average communality, which measures the quality of the measurement model for each latent variable $\boldsymbol{y}_j$, with $j = 1, \ldots, J$. The communality for block j is computed as the average of the squared correlations between the indicators in the block and the corresponding latent variable,

$$\text{communality}_j = \frac{1}{p_j} \sum_{h=1}^{p_j} \mathsf{COR}(\boldsymbol{x}_{hj}, \boldsymbol{y}_j)^2,$$

where p_j denotes the number of indicators in the jth block and $\boldsymbol{x}_{hj}$ is the hth indicator in the jth block[8].

The average communality is the average of all $\mathsf{COR}(\boldsymbol{x}_{hj}, \boldsymbol{y}_j)^2$, that is

$$\overline{\text{communality}} = \frac{1}{p} \sum_{j=1}^{J} p_j \times \text{communality}_j = \frac{1}{p} \sum_{j=1}^{J} p_j \sum_{h=1}^{p_j} \mathsf{COR}(\boldsymbol{x}_{hj}, \boldsymbol{y}_j)^2.$$

For each endogenous latent variable, redundancy measures the amount of variance in the indicators measuring the variable that is explained by the exogenous latent variables that

[8]A common measure to establish *convergent validity* on the construct level, that is the extent to which a measure correlates positively with alternative measures of the same construct, is the *average variance extracted* (AVE) measure. The AVE is equivalent to the communality of a construct.

predict the endogenous variable. For an endogenous block j, it is computed as

$$\text{redundancy}_j = \text{communality}_j \times R^2\left(\boldsymbol{y}_j, \boldsymbol{y}_j^{\text{pred}}\right).$$

If more than one endogenous variable is available in the model, then one can also calculate the average redundancy indices for all of the endogenous variables.

Finally, the goodness-of-fit (GoF) index, which takes into account both the measurement and structural model performance, is used for judging the quality of a PLS-SEM model as a whole. GoF is calculated as the geometric mean of the average communality and the average R^2:

$$\text{GoF} = \sqrt{\overline{\text{communality}} \times \overline{R}^2},$$

where the average R^2 is computed using only the endogenous latent variables in the model.

A well known theoretical deficiency of PLS-SEM is that it lacks an overall optimization criterion (such as for example the sum of squared residuals in linear regression or the likelihood function in COV-SEM). Therefore, no index for the assessment of the global validation of the model is available. The GoF index represents an operational solution to this problem which is very often used in the practical application of PLS-SEM[9].

In PLS-SEM it is assumed that the block of indicators for a reflective measure is unidimensional in the same sense of factor analysis. To check the unidimensionality of a reflective block, some reliability indexes are typically computed, the most popular ones being the *Cronbach's alpha* (α_j) and the *Dillon-Goldstein's coefficient* (ρ_j). When standardized indicators and latent variables are used, these indices are defined respectively[10] as

$$\alpha_j = \frac{\sum_{h\neq k} \mathsf{COR}(\boldsymbol{x}_{hj}, \boldsymbol{x}_{kj})}{p_j + \sum_{h\neq k} \mathsf{COR}(\boldsymbol{x}_{hj}, \boldsymbol{x}_{kj})} \times \frac{p_j}{p_j - 1},$$

and

$$\rho_j = \frac{\left(\sum_{h=1}^{p_j} \widehat{\lambda}_{hj}^{\text{outer}}\right)^2}{\left(\sum_{h=1}^{p_j} \widehat{\lambda}_{hj}^{\text{outer}}\right)^2 + \sum_{h=1}^{p_j}\left\{1 - \left(\widehat{\lambda}_{hj}^{\text{outer}}\right)^2\right\}},$$

where λ_{hj} is the outer loading for indicator h in block j. Since Cronbach's alpha tends to underestimate the internal consistency reliability, the Dillon-Goldstein's coefficient is often preferred in practice (Chin 1998, p. 320).

For more details on the assessment of PLS-SEM results and rules of thumb for evaluating the quality of a fitted model, we refer the reader to the literature (e.g., an updated and comprehensive survey is available in Hair *et al.* 2017).

[9] The lack of an explicit optimization criterion is a critical drawback of the PLS-SEM approach, which has some unpleasant consequences. The most serious is the impossibility to statistically test the relative superiority of a PLS-SEM model over any other. However, we also notice that in recent years successful attempts to derive the criteria optimized by PLS-SEM have been made (for a review see Esposito Vinzi and Russolillo 2013).

[10] Without loss of generality, in the calculation of the Cronbach's alpha it is usually assumed that all the indicators in the block are positively correlated. This is not really a big issue, since the indicators can always be built in this way.

3. The plssem package

3.1. Syntax

The syntax of **plssem** reflects the measurement and structural part of a PLS-SEM model, and accordingly requires the user to specify both of these parts simultaneously. Since a full PLS-SEM model would include a structural model, i.e., the relationship between latent variables (LV), we need to have at least two latent variables specified in the measurement part. Each latent variable will be defined by a block of indicators (say, `indblock`). For example, if we have two latent variables in our PLS-SEM model, the `plssem` syntax requires to specify the measurement part by typing

```
plssem (LV1 > indblock1) (LV2 > indblock2).
```

Clearly, one can specify as many LVs as it is needed in the model. The specification of reflective measures in the measurement model require to use the greater-than sign between a latent variable and its associated indicators (e.g., `LV1 > indblock1`), while the less-than sign needs to be provided when one needs to include latent variables measured in a formative way (e.g., `LV1 < indblock1`).

To specify the structural part[11], one needs to provide the endogenous/dependent latent variable (say, `LV2`) first followed by the exogenous latent variables (say, `LV1`) by typing

```
plssem (LV1 > indblock1) (LV2 > indblock2), structural(LV2 LV1).
```

One can specify further structural relationships following the same approach. For example, suppose one has two further latent variables in the model, `LV3` and `LV4`, still measured in a reflective way, with `LV4` endogenous and `LV3` exogenous. Then, the syntax for the structural part should be

```
plssem (LV1 > indblock1) (LV2 > indblock2) (LV3 > indblock3) ///
   (LV4 > indblock4), structural(LV2 LV1, LV4 LV3).
```

In addition, in line with most of the Stata commands, we can fit a full PLS-PM model by sub-setting the data directly in the syntax using the `if` and `in` qualifiers.

More generally, the syntax for the `plssem` command is provided by[12]

```
[by groupvar:] plssem (LV1 > indblock1) (LV2 > indblock2) (... ...)  ///
   [if exp] [in range] [, structural(LV2 LV1, ... ...) options],
```

where square brackets distinguish optional qualifiers and options from required ones, *groupvar* denotes a variable name in the data set, *exp* denotes an algebraic expression, *range* denotes an observation range, and *options* denotes a list of available options. The optional `by` prefix causes Stata to repeat a command for each subset of the data for which the values of the *groupvar* variable are equal. In other words, when prefixed with `by`, the result of the command will be the same as if one had formed separate data sets for each group of

[11] While the measurement part is mandatory, the **plssem** package allows to fit models that do not include the structural part.

[12] The **plssem** package is compatible with Stata version 10 and above.

observations, saved them, and then gave the command on each data set separately. The list of available options for the `plssem` command are illustrated in the next section.

3.2. Options

The options allowed by the `plssem` command are detailed below:

`wscheme(weighting_scheme)` provides the choice of the weighting scheme. The default is `path` for the path scheme as given in Equation 3. Alternative choices are `factorial` or `centroid` for the corresponding scheme.

`binary(LV)` indicates the latent variables that are defined by a single binary variable. This allows essentially for estimating a model with a binary dependent variable using a logistic regression model. The latent variable `LV` needs to be specified in the measurement part of the syntax at the same time (e.g., `LV > binaryvar`)[13].

`boot(#)` sets the number of bootstrap replications.

`seed(#)` sets the seed number for the bootstrap calculations. This option may be useful if reproducibility is one of the analyst's concerns.

`tol(#)` sets the tolerance value used for checking convergence attainment (see Step 5 in Stage I described in Section 2.1). The default tolerance value is `1e-7`.

`maxiter(#)` indicates the maximum number iterations the algorithm runs. The default is 100 iterations. Note that usually the algorithm requires a very limited number of iterations to reach convergence, typically less than 10.

`missing(imputation_method)` provides the choice of the imputation method for the indicator missing values. Possible choices are `mean` (i.e., the mean of the available indicators) or `knn` (i.e., the kth nearest neighbor method).

`k(#)` sets the number of nearest neighbors to use with `missing(knn)`. The default number of nearest neighbors is 5.

`init(init_method)` lets the user choose between two options for initialization. These are `indsum`[14] (default) and `eigen`[15]. The `eigen` option is required if the user wants to estimate only the measurement part of the model[16].

`digits(#)` sets the number of decimals to display the model estimates. The default is 3.

[13] This is in fact showing how we can work with single indicators using the `plssem` command. We can include both continuous and dichotomous single indicators in the model by linking them to latent variables in the measurement part of the syntax. Unless any of these latent variables is specified as binary using the `binary()` option, the structural part will apply linear (`regress`), otherwise logistic (`logit`) regression will be used. However, we stress that the same algorithm is used for the measurement part regardless of the nature of the indicators.

[14] The initial values in this option are 1s for all of the indicators.

[15] The initial values (i.e., the weights) in this option are the values associated with the first eigenvector in factor extraction's iterative process.

[16] What this initialization does is essentially running separate factor analyses with principal component extraction method (`factor, pcf`) for each latent variable in Stata. Thus, `plssem` command can conveniently be used as an alternative to the `factor, pcf` command as **plssem** would provide the user with some further estimations (i.e., reliability coefficients and discriminant validity assessment).

`noheader` suppresses the output header.

`nodiscrimtable` suppresses the discriminant validity assessment section of the output.

`nomeastable` suppresses the measurement model section of the output.

`nostructtable` suppresses the structural model section of the output.

`loadpval` shows the table of loadings' p values.

`stats` displays some summary statistics (mean, standard deviations, etc.) for each indicator.

`group(grouping variable, [suboptions])` provides both the structural and the measurement part of the estimation results for each category of the grouping variable as well as the comparison between the categories based on normal theory (default). As an alternative to normal-based theory estimations, the user can choose between two resampling techniques. More specifically, by adding the suboptions `method(permutation)` or `method(bootstrap)` one can get the results based on permutation or bootstrap resampling. The default number of replications for both permutation and bootstrap is 100. However, this can be changed by adding the suboption `reps(#)`. Further, with the suboption `groupseed(#)` one can also set a certain seed number to be able reproduce the bootstrap or permutation results. Finally, by using the suboption `plot` one can get a graphical output showing the estimates differences between the groups based on alpha level of 0.05 (default). The significance level can also be changed by adding the suboption `alpha(#)`.

`correlate(mv lv cross, cutoff())` lets the user ask for correlations among the indicators or manifest variables (`mv`), latent variables (`lv`) as well as cross-loadings (`cross`) between the indicators and latent variables[17]. When doing so, the user can also set a certain cut-off value for the correlations to be displayed by using the suboption `cutoff()`. For instance, `cutoff(0.3)` will display the correlations above 0.3 in absolute terms.

`rawsum` uses the sum of the raw indicators and the resulting aggregated scores (also called *summated scales*) are used directly for estimating the structural part. In this sense, `rawsum` is an alternative procedure to the PLS-algorithm for estimating the latent variable scores.

`noscale` if chosen, the manifest variables are not standardized before running the algorithm.

`convcrit(convergence_criterion)` the convergence criterion to use. Alternative choices are `relative` or `square`. The former corresponds to

$$\max_{\substack{k=1,\ldots,K\\ j=1,\ldots,J}} \left| \frac{\hat{w}_{kj}^{\text{old}} - \hat{w}_{kj}^{\text{new}}}{\hat{w}_{kj}^{\text{new}}} \right|,$$

while the latter to

$$\max_{\substack{k=1,\ldots,K\\ j=1,\ldots,J}} \left(\hat{w}_{kj}^{\text{old}} - \hat{w}_{kj}^{\text{new}} \right)^2.$$

The default is `relative`.

[17] These correlations are computed using the original indicators and estimated latent variable scores.

3.3. Postestimation commands

The following are the postestimation commands that can be used after fitting a PLS-SEM model with the `plssem` command. These commands can basically be categorized under two rubrics, `estat` and `plssemplot`.

`estat indirect, effects(dep med ind, ...)` estimates the specified (standardized) indirect effects and tests the significance of these effects using either the Sobel's z statistic (default) as well as the bootstrap approach[18] (Sobel 1982; Baron and Kenny 1986; VanderWeele 2015). The command can estimate up to five different indirect effects at a time. Each of these should be specified by sequentially typing the dependent (`dep`), mediator (`med`) and independent (`ind`) variable from any PLS-SEM model. By adding the suboption `boot(#)`, you can obtain the results based on the bootstrap. To facilitate the reproducibility of results, the suboption `seed(#)` can further be added to set the seed for the bootstrap calculations. Confidence intervals for the estimated indirect effects are also provided. The default confidence level is 95%, but one can change it by adding the suboption `level(#)`. To change the number of decimals used to display the estimates, one can change the default (3 digits) to another value by adding the suboption `digits(#)`.

`estat total` produces the decomposition of the total effects in (standardized) direct and indirect effects[19]. Adding the suboption `plot` will generate a bar plot of the effect decomposition. You can here too change the decimals by making use of the suboption `digits(#)`.

`estat vif` computes the variance inflation factors (VIFs) for the independent variables specified in the structural part of a PLS-SEM model. With the `digit(#)` suboption, one can change the decimal display.

`estat unobshet` assesses the presence of unobserved heterogeneity. Currently, the command implements only the REBUS-PLS approach proposed by Trinchera (2007) and Esposito Vinzi *et al.* (2008).

`plssemplot, loadings` provides a bar plot of the loadings of indicators for their respective latent variables.

`plssemplot, crossloadings` provides bar plots of the loadings of indicators for not only their respective but all the other latent variables (i.e., the cross loadings; see line 24 of Algorithm 1).

`plssemplot, scores` provides the scatterplot matrix of the scores for the latent variables defined in the PLS-SEM model.

`plssemplot, stats(LV)` provides the scatterplot matrix for the indicators in the block defining the latent variable `LV`.

[18] `estat indirect` provides the indirect effects mediated by only one latent variable.
[19] In particular, the overall indirect effects via more than one mediator variable are provided.

`plssemplot, innermodel` produces a graphical representation of the structural (inner) part of the PLS-SEM model. This command requires the installation of the `nwcommands` suite[20].

`plssemplot, outermodel` produces a visualized version of the measurement (outer) part of the PLS-SEM model. This feature is still under development, but will be available soon.

`predict, xb residuals` creates new variables containing linear predictions (option `xb`, the default) and residuals (option `residuals`). These quantities are provided only for reflective blocks of manifest variables in the measurement/outer model and for endogenous latent variables in the structural/inner model.

3.4. Stored results

Since `plssem` is built as a Stata estimation command, many of the results are stored after fitting a model. These objects might be used for further analyses after a model has been fitted. In particular, `plssem` stores the following objects accessible through the Stata's `e()` function:

- The stored scalar objects are given by:

  `e(N)`: number of observations.

  `e(reps)`: number of bootstrap replications.

  `e(iterations)`: number of iterations to reach convergence.

  `e(tolerance)`: chosen tolerance value.

  `e(maxiter)`: maximum number of iterations allowed.

  `e(converged)`: scalar equal to 1 if convergence is achieved; 0 otherwise.

- The stored macros are:

  `e(cmd)`: this is just the command name, i.e., `plssem`.

  `e(cmdline)`: the command as typed.

  `e(estat_cmd)`: the name of the program used to implement `estat`.

  `e(predict)`: program used to implement `predict`.

  `e(title)`: title in estimation output.

  `e(mvs)`: list of manifest variables (indicators) used.

  `e(lvs)`: list of latent variables used.

  `e(binarylvs)`: sublist of binary latent variables only.

  `e(datasignaturevars)`: variables used in calculation of checksum.

  `e(datasignature)`: the checksum.

  `e(reflective)`: list of latent variables measured in a reflective way.

  `e(formative)`: list of latent variables measured in a formative way.

[20]This can be achieved by executing the code `net install nwcommands-ado.pkg`.

`e(struct_eqs)`: equations defining the structural model.

`e(properties)`: choices of initialization, weighting scheme, imputation method, whether the bootstrap has been used, whether the model has a structural part, whether the `rawsum` option has been used, and whether the manifest variables have been scaled or not.

- The matrix objects saved for later use are:

`e(loadings)`: outer loadings matrix.

`e(loadings_bs)`: bootstrap-based outer loadings matrix (available only if the `boot()` option is chosen).

`e(loadings_se)`: matrix of the outer loadings standard errors.

`e(cross_loadings)`: cross loadings matrix.

`e(cross_loadings_bs)`: bootstrap-based cross loadings matrix (available only if the `boot()` option is chosen).

`e(cross_loadings_se)`: matrix of the cross loadings standard errors.

`e(adj_meas)`: adjacency matrix for the measurement (outer) model.

`e(outerweights)`: matrix of outer weights.

`e(ow_history)`: matrix of outer weights evolution.

`e(relcoef)`: matrix of reliability coefficients.

`e(sqcorr)`: matrix of squared correlations among the latent variables.

`e(ave)`: vector of average variances extracted.

`e(struct_b)`: path coefficients matrix (short form).

`e(struct_se)`: matrix of path coefficients' standard errors (in short form).

`e(struct_table)`: table combining estimation results for the structural (inner) model.

`e(pathcoef)`: path coefficients matrix (in extended form).

`e(pathcoef_bs)`: bootstrap-based path coefficients matrix (available only if the `boot()` option is chosen).

`e(adj_struct)`: adjacency matrix for the structural (inner) model.

`e(rsquared)`: vector of R^2 for reflective latent variables.

`e(redundancy)`: vector of redundancy indices.

`e(assessment)`: vector of model quality indices, that is the average R^2, the average communality, the average redundancy and the goodness-of-fit as discussed in Section 2.3.

`e(reldiff)`: vector containing the history of weights' relative differences.

`e(imputed_data)`: matrix of imputed indicators; available only if the `missing` option has been used.

- Finally, `plssem` saves a function returning an indicator that marks the observations used for fitting the model; this function is accessible through:

`e(sample)`: marks the estimation sample.

Together with the above objects, the `plssem` command also saves the latent variable scores as new columns in the active data set. These columns are labeled as the latent variables specified in the model syntax.

3.5. Additional features

The `plssem` command is also able to deal with binary latent variables, even when these are used as endogenous in the structural part of the model. This can be achieved by specifying the binary latent variables with the `binary()` option. In this case, `plssem` uses the `logit` command for fitting the logistic regression models having the binary latents as the dependent variable. Even if the corresponding path coefficients cannot be directly compared with those obtained using a linear regression model, for completeness we decided to collect and report all the coefficients in a single table.

The package also has the potential to estimate higher-order construct models entailing higher-order structure (usually second-order) that contains several layers of constructs (Lohmöller 1989, Section 3.5). In particular, one can use the so called *repeated indicators approach* (Sanchez 2013, Chapter 8) according to which one simply uses the estimated latent variable scores added to the current data set as indicators for the higher-order latent variables. This approach can be easily accomplished with the **plssem** package.

As a final note, we mention that the current release of the package provides two different approaches to deal with missing values imputation, that is mean and k-nearest neighbors imputation, through the `missing()` option. Clearly, as most Stata statistical commands do, if the `missing()` option is not specified, `plssem` treats missing values by simply disregarding observations with one or more missing values. This trivial approach to missing values is generally known as *listwise deletion*. We remind that listwise deletion provides unbiased estimates of means, variances and regression coefficients only under the restrictive assumption that the data are missing completely at random (see for example Van Buuren 2012).

4. Empirical application

In this section we illustrate the use of the **plssem** package through an example taken from our research agenda. More specifically, we use a real-life data set collected from members of a training/fitness center in 2014 in a medium-sized city in Norway. The members were asked to indicate how well having an attractive face and being sexy described them as a person using an ordinal scale (1 = very badly to 6 = very well). Using a similar scale (1 = not at all important to 6 = very important), the members were also asked to indicate how important each of the following measures was for working out:

- to have a good body;
- to improve my appearance;
- to look more attractive;
- to develop my muscles;
- to get stronger;

Indicator	Variable name	Latent variable
Attractive face	`face`	`Attractive`
Sexy	`sexy`	
To have a good body	`body`	`Appearance`
To improve my appearance	`appear`	
To look more attractive	`attract`	
To develop my muscles	`muscle`	`Muscle`
To get stronger	`strength`	
To increase my endurance	`endur`	
To lose weight	`lweight`	`Weight`
To burn calories	`calories`	
To control my weight	`cweight`	

Table 3: List of indicators collected and latent variables they measure for the empirical application described in Section 4.

- to increase my endurance;
- to lose weight;
- to burn calories;
- to control my weight.

Table 3 reports the list of indicators, the variable name in the data set and the corresponding latent construct they measure.

Specification of the PLS-SEM model

Based on relevant evolutionary psychology literature (see for example Markland and Ingledew 1997 and Kirsner, Figueredo, and Jacobs 2003), we propose the following hypotheses:

H1: The more attractive one perceives herself/himself, the more the person wants to work out to improve her/his physical appearance (i.e., Attractive → Appearance).

H2: The more the person wants to work out to improve her/his physical appearance, the more s/he wants to work out to build up muscles (i.e., Appearance → Muscle).

H3: The more the person wants to work out to improve her/his physical appearance, the more s/he wants to work out to lose weight (i.e., Appearance → Weight).

H4: The more attractive one perceives herself/himself will indirectly influence the person to work out more to build up muscles (i.e., Attractive → Appearance → Muscle).

H5: The more attractive one perceives herself/himself will indirectly influence the person to work out more to lose weight (i.e., Attractive → Appearance → Weight).

It is usual in SEM-based publications to represent these hypotheses using a path diagram to ease the understanding of the relationships (see for example Jöreskog *et al.* 2016; Kline 2016). We do this for our set of hypotheses in Figure 2.

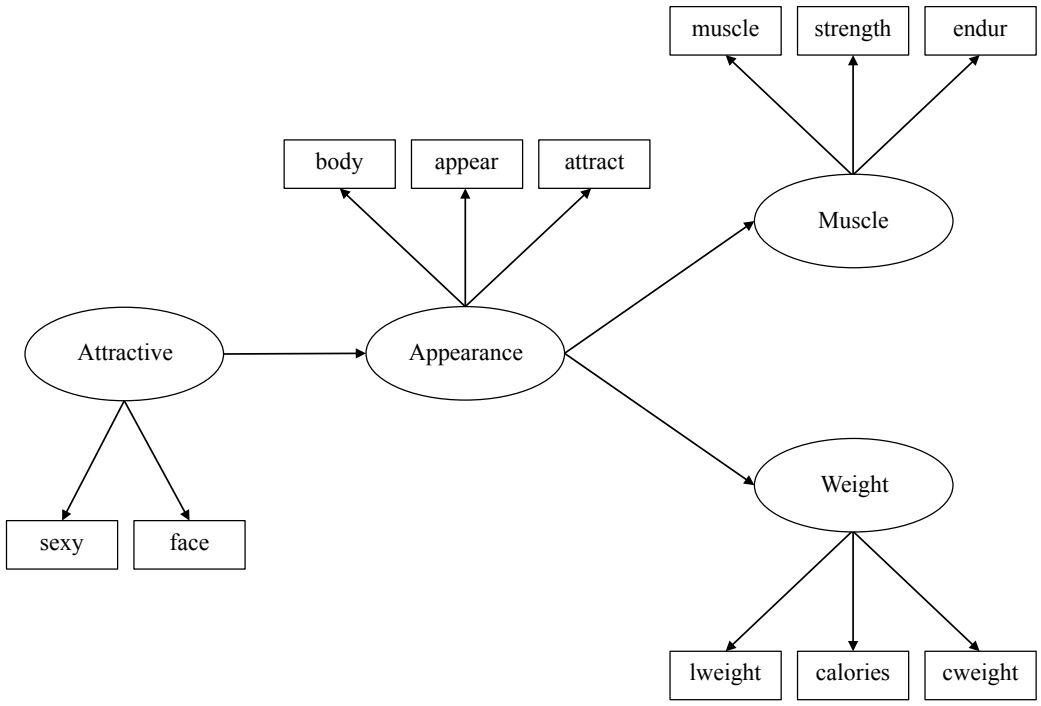

Figure 2: The hypothesized PLS-SEM model according to the hypotheses described in Section 4. `Attractive`, `Appearance`, `Muscle` and `Weight` are the latent variables defined in the model, with `Attractive` being the only exogenous variable. All the latent variables are measured in a reflective way.

Model estimation

Following the syntax and options described in Section 3, we specify and estimate our research model represented in Figure 2 with the following code:

```
. use workout2.dta,clear
. plssem (Attractive > face sexy) ///
         (Appearance > body appear attract) ///
         (Muscle > muscle strength endur) ///
         (Weight > lweight calories cweight), ///
         structural(Appearance Attractive, ///
                    Muscle Appearance, ///
                    Weight Appearance) ///
         boot(200) seed(123) stats correlate(lv)
```

The above lines of code produce the following results[21]:

```
Bootstrap replications (200)
```

[21] We compared the results for the example shown here, as well as the results for many other examples not reported in this paper, with those provided by the software mentioned in Section 1. In all cases we found an agreement in the order of at least 10^{-6}. However, we note that a perfect agreement is never possible because of minor implementation differences of the PLS-SEM algorithm in the different software.

```
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
..................................................   150
..................................................   200

Iteration 1:   outer weights rel. diff. = 6.31e-01
Iteration 2:   outer weights rel. diff. = 1.49e-02
Iteration 3:   outer weights rel. diff. = 1.34e-03
Iteration 4:   outer weights rel. diff. = 7.76e-05
Iteration 5:   outer weights rel. diff. = 6.80e-06
Iteration 6:   outer weights rel. diff. = 4.00e-07
Iteration 7:   outer weights rel. diff. = 3.49e-08

Partial least squares path modeling         Number of obs       =     187
                                            Average R-squared   =     0.15795
Weighting scheme: path                      Average communality =     0.79165
Tolerance: 1.00e-07                         GoF                 =     0.35361
Initialization: indsum                      Average redundancy  =     0.11941

Table of summary statistics for indicator variables
------------------------------------------------------------------------------
  Indicator |    mean       sd   median      min      max        N   missing
------------+-----------------------------------------------------------------
       face |   3.290    1.005    3.000    1.000    6.000      200        46
       sexy |   2.592    1.113    3.000    1.000    6.000      196        50
       body |   4.034    1.470    4.000    1.000    6.000      205        41
     appear |   3.365    1.672    3.000    1.000    6.000      203        43
    attract |   3.059    1.707    3.000    1.000    6.000      204        42
     muscle |   3.853    1.587    4.000    1.000    6.000      204        42
   strength |   4.779    1.159    5.000    1.000    6.000      208        38
      endur |   4.976    1.111    5.000    1.000    6.000      209        37
    lweight |   3.604    1.759    4.000    1.000    6.000      207        39
   calories |   4.053    1.638    4.000    1.000    6.000      207        39
    cweight |   4.048    1.666    4.000    1.000    6.000      207        39
------------------------------------------------------------------------------

Measurement model - Standardized loadings
------------------------------------------------------------------------
              |  Reflective:    Reflective:    Reflective:    Reflective:
              |   Attractive     Appearance         Muscle         Weight
--------------+---------------------------------------------------------
         face |        0.908
         sexy |        0.919
         body |                       0.899
       appear |                       0.949
      attract |                       0.923
```

```
       muscle |                                          0.886
     strength |                                          0.873
        endur |                                          0.623
      lweight |                                                        0.916
     calories |                                                        0.937
      cweight |                                                        0.911
--------------+-----------------------------------------------------------
     Cronbach |        0.801          0.914          0.734          0.912
           DG |        0.909          0.946          0.842          0.944
-------------------------------------------------------------------------
```

Discriminant validity - Squared interfactor correlation vs. Average variance extracted (AVE)

```
-------------------------------------------------------------------------
             |   Attractive     Appearance         Muscle         Weight
-------------+-----------------------------------------------------------
   Attractive |        1.000          0.080          0.021          0.002
   Appearance |        0.080          1.000          0.217          0.177
       Muscle |        0.021          0.217          1.000          0.041
       Weight |        0.002          0.177          0.041          1.000
-------------+-----------------------------------------------------------
          AVE |        0.834          0.854          0.645          0.849
-------------------------------------------------------------------------
```

Structural model - Standardized path coefficients (Bootstrap)

```
-------------------------------------------------------
     Variable |   Appearance         Muscle         Weight
--------------+----------------------------------------
   Attractive |        0.283
              |      (0.000)
   Appearance |                       0.466          0.420
              |                     (0.000)        (0.000)
--------------+----------------------------------------
         r2_a |        0.075          0.213          0.172
-------------------------------------------------------
 p-values in parentheses
```

Correlation of latent variables

```
-------------------------------------------------
             | Attrac~e Appear~e   Muscle   Weight
-------------+-----------------------------------
  Attractive |   1.0000
  Appearance |   0.2830   1.0000
      Muscle |   0.1435   0.4658   1.0000
      Weight |  -0.0414   0.4204   0.2032   1.0000
-------------------------------------------------
```

As one can see, the output commences with some summary statistics followed by the measurement part of the estimation results including the bootstrap standardized loadings. We then see a table showing the discriminant validity assessment[22] before displaying the structural part of the estimation results including bootstrap standardized path coefficients. Finally, we get a table showing the correlations among the latent variables of our model.

The output provided gives us the necessary information to test the first three hypotheses, namely **H1**, **H2** and **H3**. To be able to test mediational hypotheses (**H4** and **H5**), we make further use of the following code to estimate the indirect effects and test their statistical significance using the bootstrap method.

```
. estat indirect, effects(Muscle Appearance Attractive, ///
                          Weight Appearance Attractive) ///
                  boot(200) seed(456)

Computing indirect effects bootstrap distribution...

Significance testing of (standardized) indirect effects (Bootstrap)
------------------------------------------------------------------
                        |        Muscle <- |        Weight <-
 Statistics             |    Appearance <- |    Appearance <-
                        |       Attractive |       Attractive
------------------------+------------------+------------------
 Indirect effect        |            0.132 |            0.119
 Standard error         |            0.040 |            0.033
 Z statistic            |            3.285 |            3.564
 P-value                |            0.001 |            0.000
 Conf. interval (N)     |   (0.053, 0.210) |   (0.054, 0.184)
 Conf. interval (P)     |   (0.066, 0.228) |   (0.067, 0.197)
 Conf. interval (BC)    |   (0.071, 0.240) |   (0.068, 0.198)
------------------------------------------------------------------
 confidence level: 95%
 (N)    normal confidence interval
 (P)    percentile confidence interval
 (BC)   bias-corrected confidence interval
```

We can further ask for a graphical output showing the size of the outer loadings for each latent variable using the following code, which yields the graph shown in Figure 3:

```
. plssemplot, loadings
```

Stored results

`plssem` stores the following objects in `e()` after estimating our proposed model.

```
. ereturn list
```

[22] To be able to demonstrate discriminant validity, the average variance extracted (AVE) values should be larger than the squared correlations among the latent variables.

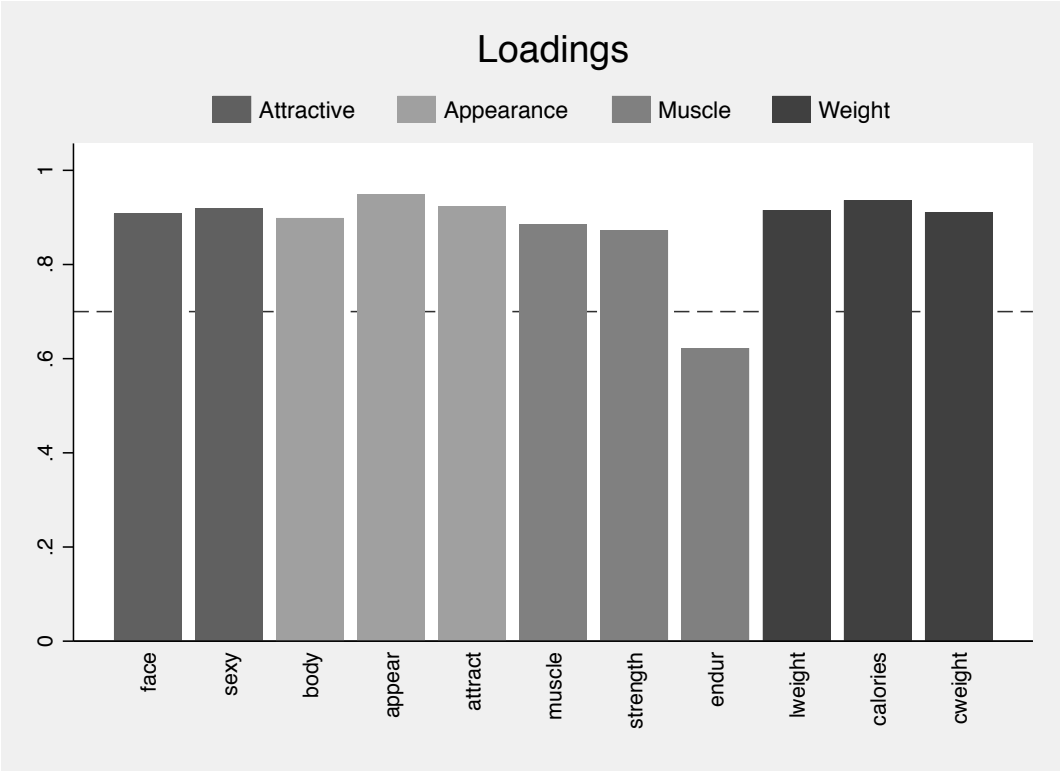

Figure 3: Bar chart reporting the outer loadings by blocks. Colors denote different indicator blocks. The dashed line provides a value (i.e., 0.7) frequently used in the literature to assess the quality of the fit.

```
. ereturn list

scalars:
          e(converged) =  1
          e(tolerance) =  1.00000000000e-07
         e(iterations) =  7
            e(maxiter) =  100
               e(reps) =  200
                  e(N) =  187

macros:
               e(cmd) : "plssem"
           e(cmdline) : "plssem (Attractive > face sexy) (Appearance > body
                         appear  ..."
          e(estat_cmd) : "plssem_estat"
           e(predict) : "plssem_p"
             e(title) : "Partial least squares structural equation modeling"
  e(datasignaturevars) : "face sexy body appear attract muscle strength endur
                         lweight ..."
      e(datasignature) : "187:15:2197803537:1706307640"
```

```
                e(mvs) : "face sexy body appear attract muscle strength endur
                         lweight ..."
                e(lvs) : "Attractive Appearance Muscle Weight"
          e(reflective) : "Attractive Appearance Muscle Weight"
          e(struct_eqs) : "(Appearance Attractive) (Muscle Appearance) (Weight
                         Appearance)"
          e(properties) : "indsum path bootstrap structural scaled relative"

matrices:
            e(indstats) :  11 x 7
            e(loadings) :  11 x 4
         e(loadings_bs) :  11 x 4
         e(loadings_se) :  11 x 4
      e(cross_loadings) :  11 x 4
   e(cross_loadings_bs) :  11 x 4
   e(cross_loadings_se) :  11 x 4
            e(adj_meas) :  11 x 4
             e(relcoef) :  2 x 4
              e(sqcorr) :  4 x 4
                 e(ave) :  1 x 4
            e(struct_b) :  2 x 3
           e(struct_se) :  2 x 3
        e(struct_table) :  5 x 3
            e(pathcoef) :  4 x 4
         e(pathcoef_bs) :  4 x 4
          e(adj_struct) :  4 x 4
       e(total_effects) :  4 x 4
            e(rsquared) :  1 x 4
          e(redundancy) :  1 x 4
          e(assessment) :  1 x 4
          e(ow_history) :  8 x 11
        e(outerweights) :  11 x 4
             e(reldiff) :  1 x 7

functions:
              e(sample)
```

Multigroup analysis

To demonstrate a further feature of the **plssem** package, in this section we perform a multigroup analysis based on the model depicted in Figure 2. More specifically, we now check whether the model estimates (path coefficients and loadings) differ between male and female respondents in our sample. As described earlier in the paper, **plssem** offers two approaches for comparing model estimates across groups: permutation and bootstrap (as well as the standard one based on normal theory). Here, we show the results using the bootstrap option with 200 replications. For reproducibility purposes, we set an arbitrary seed. We also set a significance level (`alpha`) of 0.1 to display significant path coefficients or loadings in the

resulting plot. The grouping variable, `women`, is a dummy-coded variable in which men are coded as 0.

```
. plssem (Attractive > face sexy) ///
         (Appearance > body appear attract) ///
         (Muscle > muscle strength endur) ///
         (Weight > lweight calories cweight), ///
         structural(Appearance Attractive, ///
                    Muscle Appearance, ///
                    Weight Appearance) ///
         group(women, reps(200) groupseed(123) method(bootstrap) alpha(.1)
               plot)

Bootstrap replications (200)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
..................................................   150
..................................................   200

Partial least squares path modeling

Weighting scheme: path
Tolerance: 1.00e-07
Initialization: indsum

Multigroup comparison (women) - Bootstrap t-test
------------------------------------------------------------------------------
    Measurement effect | Global Group 1 Group 2 Abs Diff Statistic P-value
-----------------------+------------------------------------------------------
    Attractive -> face |  0.908   0.816   0.943    0.127     1.573   0.117
    Attractive -> sexy |  0.919   0.936   0.910    0.026     0.355   0.723
    Appearance -> body |  0.899   0.883   0.909    0.026     1.103   0.272
  Appearance -> appear |  0.949   0.950   0.954    0.004     0.166   0.869
 Appearance -> attract |  0.923   0.946   0.911    0.035     1.239   0.217
      Muscle -> muscle |  0.886   0.887   0.882    0.004     0.070   0.945
    Muscle -> strength |  0.873   0.860   0.883    0.022     0.295   0.769
       Muscle -> endur |  0.623   0.616   0.640    0.025     0.060   0.953
     Weight -> lweight |  0.916   0.941   0.911    0.030     0.000   1.000
    Weight -> calories |  0.937   0.924   0.938    0.014     0.534   0.594
     Weight -> cweight |  0.911   0.866   0.924    0.058     0.751   0.454
------------------------------------------------------------------------------
 number of replications: 200
 group labels:
   Group 1: men
   Group 2: women
 group sizes:
```

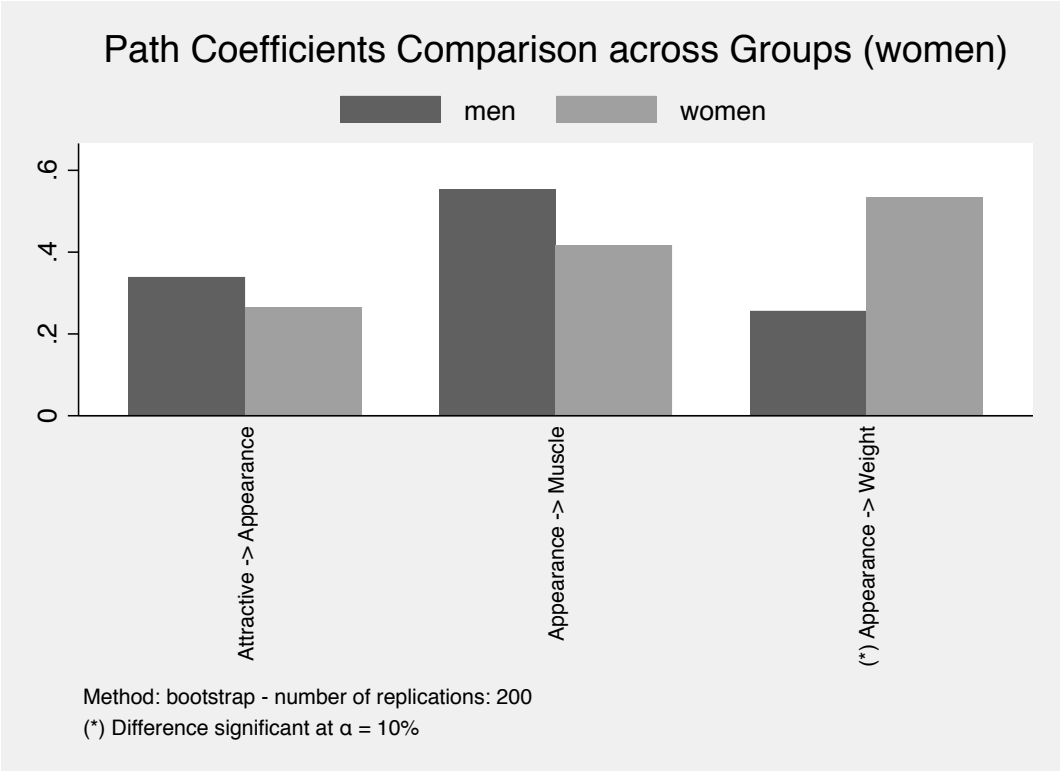

Figure 4: Comparison of path coefficients using multigroup analysis. The statistically significant differences at the given `alpha` level are highlighted with `(*)`.

```
   Group 1: 71
   Group 2: 116

Multigroup comparison (women) - Bootstrap t-test
------------------------------------------------------------------------------
       Structural effect | Global Group 1 Group 2 Abs Diff Statistic P-value
-------------------------+----------------------------------------------------
Attractive -> Appearance |  0.283   0.339   0.265   0.074      0.618   0.537
    Appearance -> Muscle |  0.466   0.554   0.417   0.137      1.362   0.175
    Appearance -> Weight |  0.420   0.257   0.533   0.276      1.799   0.074
------------------------------------------------------------------------------
 number of replications: 200
 group labels:
   Group 1: men
   Group 2: women
 group sizes:
   Group 1: 71
   Group 2: 116
```

The results show the path coefficients for the whole sample (`Global`) as well as those for the samples containing the men (`Group 1`) and women (`Group 2`). More importantly, a

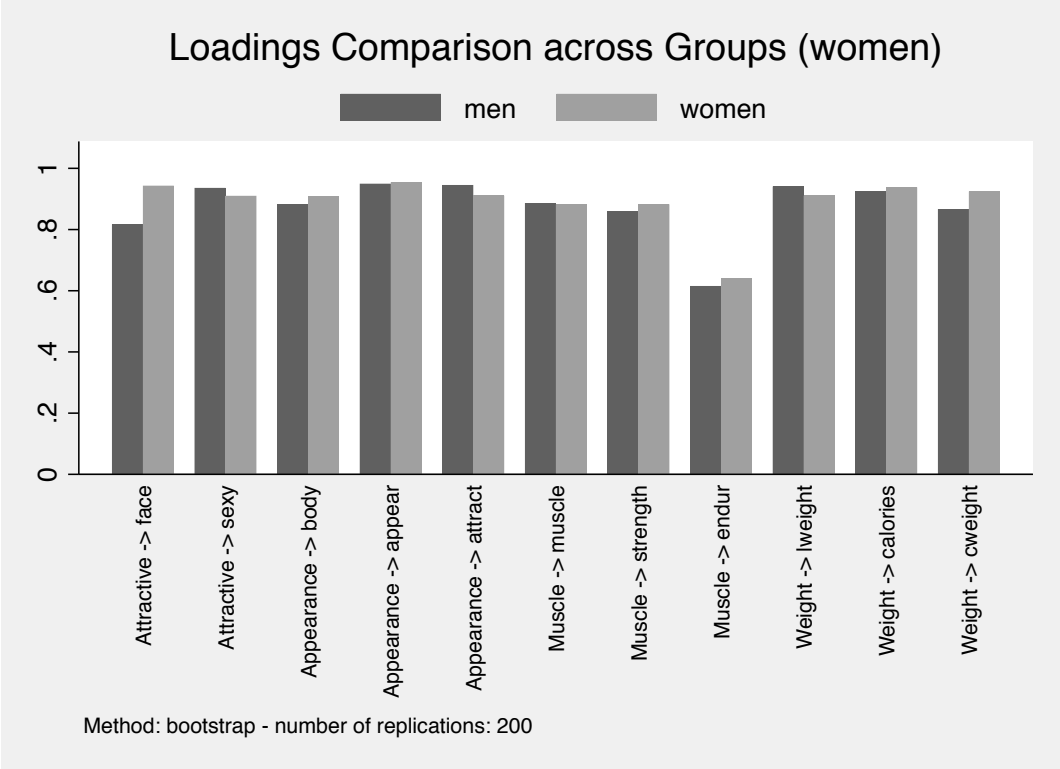

Figure 5: Comparison of outer loadings using multigroup analysis. In this case, none of the differences is significant at the given `alpha` level.

bootstrapped t test is run based on these estimates. We can conclude that the effect of `Appearance` on `Weight` is larger among women than men. This difference is significant at the 0.1 significance level though. The remaining path coefficients are not significantly different between the two groups. Figure 4 reports the plot produced by the above code showing the magnitudes of the differences among the path coefficients. The plot also shows the path coefficients (if any) that are significantly different between groups according to the chosen `alpha` level by marking them with a star.

The same command also provides the comparison of the model's loadings between men and women represented in Figure 5. None of the loadings is significantly different between men and women. Equality of loadings is indeed an important condition that must be met for establishing measurement invariance before comparing path coefficients of different models. Thus, ideally and as done in real-life research practice, the comparison of the measurement model parameters should precede the comparison of the structural model parameters.

5. Conclusion

In this article, we introduced the **plssem** package for estimation of partial least squares structural equation modeling (PLS-SEM). We demonstrated the capabilities of the package using a common and multi-featured empirical application. **plssem** can as easily be used to estimate more complex PLS-SEM models such as higher-order latent variable models. Future releases

of the command will include further more advanced features, in particular we plan to add capabilities for multilevel modeling, more options for missing values imputation and more elaborate approaches for dealing with observed and unobserved heterogeneity.

References

Addinsoft (2007). ***XLSTAT*** – *Statistical Software for MS Excel*. URL https://www.xlstat.com/.

Arbuckle JL (2014). ***Amos*** *23.0 User's Guide*. IBM SPSS, Chicago.

Baron RM, Kenny DA (1986). "The Moderator-Mediator Variable Distinction in Social Psychological Research: Conceptual, Strategic, and Statistical Considerations." *Journal of Personality and Social Psychology*, **51**(6), 1173–1182. doi:10.1037//0022-3514.51.6.1173.

Bentler PM (2008). ***EQS*** *6 Structural Equations Program Manual*. Multivariate Software, Inc.

Bollen KA (1989). *Structural Equation with Latent Variables*. John Wiley & Sons. doi:10.1002/9781118619179.

Brown TA (2015). *Confirmatory Factor Analysis for Applied Research*. Guilford Press.

Canty A, Ripley BD (2017). ***boot***: *Bootstrap Functions (Originally by Angelo Canty for S)*. R package version 1.3-20, URL https://CRAN.R-project.org/package=boot.

Chin WW (1995). "Partial Least Squares is to **LISREL** as Principal Components Analysis is to Common Factor Analysis." *Technology Studies*, **2**, 315–319.

Chin WW (1998). "The Partial Least Squares Approach for Structural Equation Modeling." In GA Marcoulides (ed.), *Modern Methods for Business Research*, pp. 295–336. Lawrence Erlbaum Associates.

Davison AC, Hinkley DV (1997). *Bootstrap Methods and Their Applications*. Cambridge University Press. URL http://statwww.epfl.ch/davison/BMA/.

Dijkstra TK, Henseler J (2015a). "Consistent and Asymptotically Normal PLS Estimators for Linear Structural Equations." *Computational Statistics & Data Analysis*, **81**, 10–23. doi:10.1016/j.csda.2014.07.008.

Dijkstra TK, Henseler J (2015b). "Consistent Partial Least Squares Path Modeling." *MIS Quarterly*, **39**(2), 297–316. doi:10.25300/misq/2015/39.2.02.

Efron B, Hastie T (2016). *Computer Age Statistical Inference*. Cambridge University Press.

Esposito Vinzi V, Russolillo G (2013). "Partial Least Squares Algorithms and Methods." *WIREs Computational Statistics*, **5**(1), 1–19. doi:10.1002/wics.1239.

Esposito Vinzi V, Trinchera L, Amato S (2010). "PLS Path Modeling: From Foundations to Recent Developments and Open Issues for Model Assessment and Improvement." In V Esposito Vinzi, WW Chin, J Henseler, H Wang (eds.), *Handbook of Partial Least Squares: Concepts, Methods and Applications*, pp. 47–82. Springer-Verlag.

Esposito Vinzi V, Trinchera L, Squillacciotti S, Tenenhaus M (2008). "REBUS-PLS: A Response-Based Procedure for Detecting Unit Segments in PLS Path Modeling." *Applied Stochastic Models in Business and Industry*, **24**(5), 439–458. doi:10.1002/asmb.728.

Fox J, Nie Z, Byrnes J (2017). ***sem**: Structural Equation Models*. R package version 3.1-9, URL https://CRAN.R-project.org/package=sem.

Hahn C, Johnson MD, Herrmann A, Huber F (2002). "Capturing Customer Heterogeneity Using a Finite Mixture PLS Approach." *Schmalenbach Business Review*, **54**(3), 243–269. doi:10.1007/bf03396655.

Hair JF, Hult GTM, Ringle CM, Sarstedt M (2017). *A Primer on Partial Least Squares Structural Equation Modeling (PLS-SEM)*. 2nd edition. Sage.

Henseler J (2017). ***ADANCO** 2.0.1 User Manual*. Composite Modeling GmbH & Co. KG, Kleve. URL https://www.composite-modeling.com/.

Henseler J, Dijkstra TK, Sarstedt M, Ringle CM, Diamantopoulos A, Straub DW, Ketchen DJ, Hair JF, Hult GTM, Calantone RJ (2009). "The Use of Partial Least Squares Path Modeling in International Marketing." *Advances in International Marketing*, **20**, 277–320.

Henseler J, Dijkstra TK, Sarstedt M, Ringle CM, Diamantopoulos A, Straub DW, Ketchen DJ, Hair JF, Hult GTM, Calantone RJ (2014). "Common Beliefs and Reality About PLS: Comments on Rönkkö and Evermann (2013)." *Organizational Research Methods*, **17**(2), 182–209. doi:10.1177/1094428114526928.

Hwang H, Takane Y (2004). "Generalized Structured Component Analysis." *Psychometrika*, **69**(1), 81–99. doi:10.1007/bf02295841.

Hwang H, Takane Y (2014). *Generalized Structured Component Analysis: A Component-Based Approach to Structural Equation Modeling*. CRC Press. doi:10.1201/b17872.

IBM Corporation (2017). *IBM SPSS Statistics 25*. IBM Corporation, Armonk. URL http://www.ibm.com/software/analytics/spss/.

Jöreskog KG (1969). "A General Approach to Confirmatory Maximum Likelihood Factor Analysis." *Psychometrika*, **34**(2), 183–202. doi:10.1007/bf02289343.

Jöreskog KG, Olsson UH, Wallentin FY (2016). *Multivariate Analysis with **LISREL***. Springer-Verlag.

Jöreskog KG, Sörbom D (2015). ***LISREL** 9.20 for Windows*. Scientific Software International, Inc., Skokie.

Kirsner BR, Figueredo AJ, Jacobs WJ (2003). "Self, Friends, and Lovers: Structural Relations Among Beck Depression Inventory Scores and Perceived Mate Values." *Journal of Affective Disorders*, **75**(2), 131–148. doi:10.1016/s0165-0327(02)00048-4.

Kline RB (2016). *Principles and Practice of Structural Equation Modeling*. 4th edition. The Guilford Press.

Kock N (2018). ***WarpPLS** User Manual: Version 6.0*. ScriptWarp Systems, Laredo. URL http://www.scriptwarp.com/warppls/.

Lohmöller JB (1989). *Latent Variable Path Modeling with Partial Least Squares*. Physica. doi:10.1007/978-3-642-52512-4.

Markland D, Ingledew DK (1997). "The Measurement of Exercise Motives: Factorial Validity and Invariance Across Gender of a Revised Exercise Motivations Inventory." *British Journal of Health Psychology*, **2**(4), 361–376. doi:10.1111/j.2044-8287.1997.tb00549.x.

Mevik BH, Wehrens R, Liland KH (2018). ***pls**: Partial Least Squares and Principal Component Regression*. R package version 2.7-0, URL https://CRAN.R-project.org/package=pls.

Monecke A, Leisch F (2012). "**semPLS**: Structural Equation Modeling Using Partial Least Squares." *Journal of Statistical Software*, **48**(3), 1–32. doi:10.18637/jss.v048.i03.

Muthén LK, Muthén BO (2017). ***Mplus** User's Guide*. 8th edition. Muthén & Muthén, Los Angeles.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Ringle CM, Wende S, Becker JM (2015). ***SmartPLS** 3*. **SmartPLS**, Bönnigstedt.

Rönkkö M (2016). ***StataPLS***. URL https://github.com/mronkko/StataPLS.

Rönkkö M (2017). ***matrixpls**: Matrix-Based Partial Least Squares Estimation*. R package version 1.0.5, URL https://CRAN.R-project.org/package=matrixpls.

Rönkkö M, Evermann J (2013). "A Critical Examination of Common Beliefs About Partial Least Squares Path Modeling." *Organizational Research Methods*, **16**(3), 425–448. doi:10.1177/1094428112474693.

Sanchez G (2013). *PLS Path Modeling with R*. Trowchez Editions. URL http://www.gastonsanchez.com/PLSPathModelingwithR.pdf.

Sanchez G, Trinchera L, Russolillo G (2017). ***plspm**: Tools for Partial Least Squares Path Modeling (PLS-PM)*. R package version 0.4.9, URL https://CRAN.R-project.org/package=plspm.

Sarstedt M, Becker JM, Ringle CM, Schwaiger M (2011). "Uncovering and Treating Unobserved Heterogeneity with FIMIX-PLS: Which Model Selection Criterion Provides an Appropriate Number of Segments?" *Schmalenbach Business Review*, **63**(1), 34–62. doi:10.1007/bf03396886.

SAS Institute Inc (2013). *The SAS System, Version 9.4*. SAS Institute Inc., Cary. URL http://www.sas.com/.

Sobel MN (1982). "Asymptotic Confidence Intervals for Indirect Effects in Structural Equations Models." In S Leinhart (ed.), *Sociological Methodology*, pp. 290–312. Jossey-Bass.

StataCorp (2017). *Stata Statistical Software: Release 15*. StataCorp LLC, College Station. URL http://www.stata.com/.

Temme D, Kreis H, Hildebrandt L (2010). "A Comparison of Current PLS Path Modeling Software: Features, Ease-of-Use, and Performance." In V Esposito Vinzi, WW Chin, J Henseler, H Wang (eds.), *Handbook of Partial Least Squares: Concepts, Methods and Applications*, pp. 737–756. Springer-Verlag.

Tenenhaus M, Esposito Vinzi V, Chatelin YM, Lauro C (2005). "PLS Path Modeling." *Computational Statistics & Data Analysis*, **48**(1), 159–205. doi:10.1016/j.csda.2004.03.005.

Trinchera L (2007). *Unobserved Heterogeneity in Structural Equation Models: A New Approach to Latent Class Detection in PLS Path Modeling*. Ph.D. thesis, University of Naples "Federico II".

Van Buuren S (2012). *Flexible Imputation of Missing Data*. CRC Press. doi:10.1201/b11826.

VanderWeele TJ (2015). *Explanation in Causal Inference*. Oxford University Press.

Wehrens R (2011). *Chemometrics with R*. Springer-Verlag. doi:10.1007/978-3-642-17841-2.

Wold HOA (1975). "Path Models with Latent Variables: The NIPALS Approach." In HM Blalock, A Aganbegian, FM Borodkin, R Boudon, V Cappecchi (eds.), *Quantitative Sociology*, pp. 307–359. Academic Press.

Wold HOA (1982). "Soft Modeling: The Basic Design and Some Extensions." In KG Jöreskog, HOA Wold (eds.), *Systems under Indirect Observations, Part II*, pp. 1–54. North-Holland.

Affiliation:

Sergio Venturini
Department of Decision Sciences
Università Commerciale L. Bocconi
20136 Milan, Italy
E-mail: sergio.venturini@unibocconi.it

Mehmet Mehmetoglu
Department of Psychology
Norwegian University of Science and Technology
7491 Trondheim, Norway
E-mail: mehmetm@svt.ntnu.no

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
March 2019, Volume 88, Issue 8 *Submitted:* 2017-01-07
doi:10.18637/jss.v088.i08 *Accepted:* 2017-10-09