Parte \mathbf{V}

Software de uso libre

Capítulo 12

Modelos de ecuaciones estructurales con mínimos cuadrados parciales: Un tutorial en R

Jorge Alberto Ruiz Vázquez⁴¹ Cristina Guadalupe Guerrero Sánchez⁴²



⁴¹ Doctor en Psicología por la Universidad Nacional Autónoma de México. ORCID: 0000-0001-9495-2042. Investigador de tiempo completo, adscrito al Centro de Estudios e Investigaciones en Comportamiento y Aprendizaje Humano de la Universidad Veracruzana. Email: ruizvja@yahoo.com

⁴² Maestra en Psicología por la Universidad Nacional Autónoma de México. Estudiante del Doctorado en Ciencia del Comportamiento de la Universidad Veracruzana.

Resumen

En el presente capítulo se presenta un tutorial para realizar modelos de ecuaciones estructurales en condiciones en las que el método de mínimos cuadrados parciales (PLS-SEM) resulta apropiado. Se presenta el código correspondiente para llevar a cabo el modelado de ecuaciones estructurales en el entorno de R, principalmente por sus características de ser de acceso libre y de código abierto. Inicialmente se describen las características del software y se presentan algunos ejercicios para tener un primer acercamiento con este. Posteriormente, se desglosan los pasos a seguir empleando este software y el paquete SEMinR diseñado por otros autores para llevar a cabo modelos de ecuaciones estructurales con diversos métodos, entre ellos el de mínimos cuadrados parciales. Se concluye que la realización de estos modelos mediante lenguajes como R potencia la posibilidad de llevar a cabo modelamientos PLS-SEM sin tener que pagar una licencia costosa, así como la posibilidad de crear scripts que faciliten la replicabilidad de análisis de datos y reportes de resultados.

Introducción

El uso de modelos de ecuaciones estructurales como herramienta de evaluación de relaciones específicas entre variables y constructos teóricos ha sido cada vez más frecuente en las investigaciones de diferentes disciplinas, incluyendo a las ciencias administrativas (Soriano y Mejía-Trejo, 2022). Si bien el modelamiento de ecuaciones estructurales ha estado vigente como método de regresión desde principios del siglo XX (Wright, 1920), su uso se vio potenciado a partir de la contribución de Jöreskog (1970) respecto a un método particular para resolver un sistema de ecuaciones que permitiera estimar los coeficientes para modelar relaciones lineales estructurales, junto con el primer software programado en lenguaje FOR-TRAN-IV y diseñado para desarrollar este tipo de modelos con ayuda de las computadoras (LISREL, por Linear Structural Relations; Jöreskog y Van Thiilo, 1972). Con el paso del tiempo, LISREL ha evolucionado, tanto en sus algoritmos matemáticos como en los algoritmos computacionales que permiten su ejecución (https://ssicentral.com/wp-content/ uploads/2020/07/LISREL_History.pdf). Así mismo, otros desarrolladores han aparecido en la escena con sus respectivos productos, entre los que destacan principalmente EQS (por Equations; Bentler, 1985) y AMOS (por Analysis of Moment Structures; Barnidge y De Zúñiga, 2017) por su uso frecuente en la literatura en la que se emplean modelos de ecuaciones estructurales (Soriano y Mejía-Trejo, 2022).

En los últimos años ha habido un incremento prácticamente exponencial en el uso de la técnica de mínimos cuadrados parciales (PLS, por sus siglas en inglés) como un método alternativo para la estimación de coeficientes en los modelos de ecuaciones estructurales (PLS-SEM, en adelante). Por ejemplo, de acuerdo con Hair y colaboradores (2022), para el año 2020 el número de citas del método PLS-SEM alcanzaba las 1500 menciones tan solo entre publicaciones registradas en Web of Science, mientras que diez años antes apenas eran unos cuantos artículos los que citaban este método. El método PLS-SEM, en relación con los modelos típicos basados en covarianzas (CB-SEM, en adelante), destaca por diferentes razones (Hair, et al. 2022): a) maximiza la varianza explicada; b) es especialmente útil para modelar constructos formativos; c) se obtienen resultados con niveles de potencia estadística aceptable empleando tamaños de muestra más pequeños que los requeridos en los modelos CB-SEM; d) no se asume una distribución normal de los datos y es robusto ante a la asimetría; e) igualmente es robusto en la construcción de modelos complejos con demasiadas variables observadas; f) la evaluación del modelo no depende de la bondad de ajuste; g) permite el modelamiento de constructos de orden superior a partir de un mínimo de dos constructos de orden inferior.

En el campo de investigación de las ciencias administrativas existen diversos ejemplos de la aplicación del método PLS-SEM, en donde destaca el uso principalmente del software Smart-PLS en alguna de sus versiones (Ali et al. 2017; Máynez-Guaderrama y Vargas-Salgado, 2019; Sharma y Kaur, 2024; Takaki et al. 2015). Aunque este software tiene la característica de contar con un entorno de desarrollo integrado (IDE por sus siglas en inglés) sumamente amigable con el usuario y que es posible contar con fuentes bibliográficas (Hair et al., 2019; 2022) como apoyo y guía para llevar a cabo el análisis que se desea, su costo es alto. Por ejemplo, la licencia para usar la versión más reciente del software, Smart-PLS 4.0, en una sola computadora tiene un costo de 560 euros al año (10 223.70 MXN, aproximadamente); para usarlo hasta en cualquier computadora, su costo es de 1120 euros al año (20 447.39 MXN., aproximadamente).

Una alternativa viable para aplicar el método PLS-SEM si no se está en condiciones de pagar una licencia es el uso de software libre. Un caso específico es el del software R (), mediante el cual es posible llevar a cabo un análisis de este tipo mediante el paquete SEMinR (), entre muchos otros tipos de técnicas de análisis y visualización de datos. Otro aspecto importante en el que repercute el uso de R es respecto a la replicabilidad del análisis de datos, cuestión que recientemente se ha discutido en el contexto de la transparencia en el manejo y análisis de datos en las investigaciones en las que se emplea el modelo PLS-SEM (Adler et al., 2023).

Primer contacto con R y RStudio

Ante la posibilidad de que el nivel de dominio sobre R de algunos de los posibles lectores de este capítulo sea el de principiante, consideramos oportuno incluir una sección introductoria al proceso de instalación y ejecución del software R y el editor RStudio. En primer lugar, el usuario debe dirigirse a la página de descarga de Rstudio (https://posit.co/download/ rstudio-desktop/), donde encontrará las ligas correspondientes para descargar el archivo de instalación de R, así como la liga para descargar el archivo de instalación de Rstudio. Al dar clic en la liga correspondiente a R (https://cran.rstudio.com/), el usuario será redirigido al espejo del sitio web del Proyecto R (Comprehensive R Archive Network, CRAN en adelante) en el que será posible seleccionar el archivo de instalación que mejor corresponda con el sistema operativo instalado en el ordenador del usuario. Por ejemplo, se puede dar clic en la liga "Download R for Windows" (https://cran.rstudio.com/bin/windows/), y enseguida dar clic en la opción "Install R for the first time" (https://cran.rstudio.com/ bin/windows/base/), y será redirigido a la página en la que se encuentra disponible la versión actualizada del software (https://cran.rstudio.com/ bin/windows/base/R-4.3.3-win.exe). En esta página se encontrará información importante sobre el software, por ejemplo, el archivo READ-ME o las preguntas frecuentes sobre el software (FAQs).

Una vez que el archivo se hava descargado, el procedimiento de instalación es semejante a la instalación de cualquier otro archivo ejecutable con el que el usuario del ordenador haya tenido experiencia. Una vez instalado R, el usuario puede usar la consola de este programa, aunque esto podría representar la opción menos viable para alguien que recién inicia en el mundo de la programación en R. En cambio, existe la posibilidad de emplear un editor especializado como es el caso de RStudio para programar en un entorno más amigable con el usuario, particularmente porque hace explícitas diversas funciones de R que en el caso de cuando se usa solo la consola de R no son evidentes y eso podría dificultar el aprendizaje de su uso para alguien con nula experiencia. Para descargar el archivo de instalación de Rstudio, se debe dar clic en la opción correspondiente desde la página https://posit.co/download/rstudio-desktop/ y se descargará de manera automática el archivo para Windows. Si el usuario requiere una versión para un sistema operativo diferente, puede explorar en la lista que se muestra en la misma página principal de descarga. Una vez que el archivo correspondiente se hava descargado, el usuario debe ejecutarlo y seguir las indicaciones durante el proceso de instalación.

Una vez que ambos archivos han sido instalados, es suficiente con que el usuario ejecute Rstudio y este se enlazará automáticamente con la consola de R. En la Figura 1 se muestra una captura de pantalla de la apariencia del IDE de Rstudio cuando se inicia el programa por primera vez. El entorno se divide originalmente en tres ventanas (B, C y D), aunque en la imagen que se muestra se incluye la Ventana A, debido a que al iniciar por primera vez una sesión de trabajo es necesario abrir un archivo nuevo para comenzar a programar (para esto solo es necesario seguir la ruta: Menu File > New file > R Script).

Figura 1



Apariencia del IDE de RStudio.

Nota: Al abrir el IDE por primera vez sólo aparecen tres ventanas (B, C y D); sin embargo, la Ventana A aparece en cuanto se solicita abrir un nuevo archivo siguiendo la ruta: Menu File > New File > R script. *Fuente*: Captura de pantalla del IDE de Rstudio tomada por los autores.

En la Ventana A, arriba a la izquierda, se encuentra la ventana de escritura de comandos o conjuntos de estos para realizar una tarea que implica varios pasos. En la Ventana B, abajo a la izquierda, se muestra una ventana que corresponde a la consola de R, que es el entorno en el que propiamente se llevarán a cabo las tareas que son ejecutadas desde la ventana de script, aunque también se pueden ejecutar tareas directamente en esta ventana. En la Ventana D, arriba a la derecha, se muestra una ventana en la que las dos primeras pestañas son de particular ayuda cuando se programa en R: la de Environment, que es en donde se va mostrando cualquier objeto creado en el entorno de programación, ya sea por importación de un archivo externo o como resultado de la ejecución de alguna instrucción programada, y la de History, que muestra el historial de trabajo. En la Ventana D, abajo a la derecha, se muestra una ventana con varias pestañas

importantes. Las primeras cuatro permiten visualizar los archivos disponibles en el directorio de trabajo vinculado con R (por default, la carpeta de Documentos), los paquetes instalados (activos o inactivos) en el entorno de R, los gráficos que se producen como parte del código que se ejecuta, así como la documentación de cada paquete y funciones instalados en el entorno de R. La Tabla 1 muestra una descripción breve de las diferentes pestañas incluidas en cada ventana, así como una breve descripción de su función.

Pestaña Función Ventana Área de código Se muestra el archivo con el código de la rutina que se А ejecuta. В Consola Se muestra la consola del software R, que es en donde se ejecutan los algoritmos programados. Permite acceso al sistema interno de Rstudio. В Terminal В Trabajos de res-Permite que algunos códigos de R trabajen en sesiones paldo diferentes. С Ambiente Se muestran los datos, listas y variables que están activos para trabajar; así como las características. С Historia Se muestra el registro de entradas realizadas en la consola en orden cronológico. Conexiones Permite realizar conexiones a bases de datos. С С Tutorial Se muestran los tutoriales de RStudio, requiere la instalación del paquete learnr. D Archivos Se muestran los archivos y carpetas del equipo. Se puede elegir en esta pestaña el Directorio de trabajo. Gráficos D Se presentan los gráficos realizados en R en orden cronológico. D Paquetes Se muestra la lista de paquetes del usuario (biblioteca). Pueden activarse o desactivarse al dar clic en la casilla. D Permite la búsqueda en la documentación de R. Ayuda D Visor Permite la visualización de contenido localizado en la red local. Algunos gráficos permiten la interacción con los datos mostrados. D Presentación Permite la visualización de presentaciones dinámicas.

Distribución de ventanas, pestañas y funciones en el IDE de RStudio.

Tabla 1

Antes de describir como desarrollar un análisis PLS-SEM en R, consideramos oportuno ejemplificar algunas nociones básicas sobre la manera en la que se programa en R. Este lenguaje funciona a partir de líneas de comandos que esencialmente se construyen a partir de dos partes: objetos y funciones, unidas por el símbolo "<-" (que significa "creado a partir de"). Un objeto en R puede tomar un valor específico (e.g., una constante), una variable (númerica, lógica, categórica, etc.), un conjunto de variables (del mismo o de diferente tipo), una base de datos completa, entre otros. Una función en R es toda operación posible, ya sea que se empleen las funciones incluidas en el paquete Base de R, de paquetes descargados explícitamente por el usuario (como el que aquí se describe), o bien, funciones creadas por el propio usuario a partir de su programación en el entorno de R.

En la Figura 2 se muestra un ejemplo que el lector podría ejecutar en su computadora como práctica introductoria al uso de R. Al escribir y ejecutar la primera línea, el programa no devuelve un resultado en la consola de R, aunque sí en la pestaña de Environment (Ventana C, de acuerdo con la Figura 1), en la que ahora aparecerá un objeto (example1) creado a partir de la concatenación de los valores 1, 2, 3, 4 y 5. De la misma manera en la que se ha creado este objeto, se puede crear cualquier otro. Más adelante se ilustra cómo puede crearse un objeto que contenga una base de datos completa proveniente de un archivo externo. Una vez que un objeto existe en el entorno de R, se pueden ejecutar funciones respecto de él. El tipo de acciones que pueden llevarse a cabo depende de las características del objeto, por ejemplo, si es un valor único o si es un conjunto de valores, si se trata de valores numéricos o de otra categoría, entre otros aspectos. En la segunda línea se ejemplifica el cálculo de la media de los valores concatenados en el objeto example1 y se muestra el resultado que aparece en la consola de R en la línea que inicia con los símbolos ##, es decir, la media es 3. Si el resultado de la aplicación de una función quisiera guardarse en un objeto, solo se debe emplear la misma lógica que en el caso de la creación de example1, pero esta vez el objeto será creado a partir del cálculo de la media.

La tercera línea de comandos ejemplifica una de las funciones más empleadas en R para la creación de gráficos. En este caso, se ejemplifica la situación más simple en la que puede solicitarse una gráfica. Puede apreciarse en el dispersograma que el gráfico se produce considerando algunos aspectos de manera automática, como que el tipo de gráfico es de puntos (círculos vacíos), se asigna el nombre del objeto que contiene los datos graficados como la etiqueta para el eje de las ordenadas, mientras que se usa el indicador de posición de cada valor en el eje de las abscisas, además de la etiqueta Index por defecto. La función plot() admite una serie de argumentos que permiten dar un formato especializado al gráfico que se busca producir.

Figura 2

Ejemplo de programación y productos arrojados en el software R.

```
example1<-c(1,2,3,4,5)
mean(example1)
## [1] 3
plot(example1)</pre>
```



Una de las características más importantes del entorno de R es que se puede acceder a la documentación de cualquier paquete o funciones específicas directamente desde el entorno de programación. Para obtener información de los argumentos que permiten especificar los atributos con los que puede (o debe) operar cualquier función, es suficiente con escribir el nombre de la función en el campo de búsqueda en la pestaña Help (véase Ventana D, en la Figura 1). También puede escribirse directamente en la consola el signo de interrogación "?", seguido del nombre de la función de la que se desea conocer su documentación. Por ejemplo, *?install. packages()* devuelve en la pestaña Help la información sobre la descripción, el formato de uso, los argumentos e incluso ejemplos de uso de la función que permite instalar paquetes en el entorno de R, entre otros rubros de información. En la Figura 3 se muestra una captura de pantalla del resultado de la ejecución de la instrucción ejemplificada. En la siguiente sección se ejemplificará el uso de esta función para los casos concretos de los paquetes que permiten importar datos desde archivos externos guardados desde Microsoft Excel (con extensión .xlsx) o desde SPSS (con extensión .sav).

Figura 3





Nota: Se muestra un fragmento de la documentación total disponible sobre la función. *Fuente*: captura de pantalla del IDE de Rstudio tomada por los autores.

Ejemplo de aplicación

En esta sección se presenta la ejecución de un análisis llevado a cabo con datos recolectados por el primer autor como parte de un proyecto de investigación relacionado con el concepto de identidad universitaria (Ruiz et al., 2020). Debido al interés principalmente didáctico del presente capítulo, no se profundiza en los detalles teóricos y conceptuales sobre la identidad universitaria, por lo que se recomienda acudir a la citada publicación para leer esos detalles y también para comparar los resultados obtenidos con la serie de comandos que se describen a continuación y los reportados previamente (obtenidos usando el software Smart-PLS 2). Debe tomarse en cuenta que la base de datos actual es una muestra de la base completa empleada en la publicación previa. La base de datos puede descargarse en cualquiera de tres formatos de archivo desde el sitio: https://github.com/ruizvja/datos_ejemplo_PLS-SEM.

Importación y análisis exploratorio de datos

El punto de partida es importar la base de datos con la que se llevará a cabo el análisis correspondiente. Para ello, es necesario atender a la extensión del archivo de origen en el que se encuentran almacenados los datos, pues dependiendo de la extensión, será necesario usar la función de uno u otro paquete para hacer el llamado del archivo. La extensión del archivo de origen depende completamente del programa que se haya empleado para la captura y guardado de los datos cuando estos fueron recolectados. Por ejemplo, la base de datos que se utiliza para el ejemplo aquí descrito fue creada en el software SPSS, por lo que su extensión es .sav, mientras que cualquier otro usuario podría crear sus bases de datos en Excel o incluso en un editor de texto simple como Notepad y guardarlas como un archivo de valores separados por comas (.csv).

A continuación se ilustran las líneas de comandos que habría que escribir y ejecutar para importar la base de datos después de que se haya descargado del sitio web en alguna de las versiones disponibles. Si el archivo empleado es el que tiene la extensión .sav, debe instalarse el paquete foreign (R Core Team, 2022) en primer lugar y posteriormente activar la librería en la sesión de trabajo en el entorno de R. En tercer lugar, se debe emplear la función read.spss() con dos de sus argumentos debidamente especificados: file, para señalar el nombre del archivo (y su ruta de localización en el ordenador); y to.data.frame = TRUE, para que los datos importados se configuren como un data frame en el entorno de R. Si es el caso del archivo con extensión .xlsx, se debe instalar el paquete readxl (), activar su librería en la sesión de R y emplear la función read_excel(), con los argumentos: file, para especificar el nombre el archivo que se va a importar; y na = "NA", para establecer que aquellas celdas en las que aparezcan las siglas NA se debe interpretar como datos faltantes (si el usuario usa otro símbolo para identificar los datos faltantes en sus propias bases de datos, ese símbolo es el que habría que escribir entre comillas para el argumento na). Si se trata de un archivo de valores separados por comas, se puede emplear directamente la función read.csv() sin requerir la instalación de ningún paquete ni la activación de ninguna librería especializada, y cuyos argumentos necesarios son: file, para especificar el nombre del archivo que se va a importar; y header = TRUE, para señalar que el primer renglón de la base de datos contiene los encabezados (nombres) de cada variable (columna).

En cualquiera de los tres casos se asume que los datos que se importan quedan alojados en el objeto dt, el cual debe aparecer en la pestaña Environment (véase Ventana C de la Figura 1), con un aspecto en el que se informa que se trata de un objeto del tipo data frame, con 122 observaciones y 26 variables. Al hacer clic sobre el nombre del objeto, o al escribir y ejecutar en la consola el comando View(dt), se visualiza la base de datos correspondiente en la Ventana A, como se ilustra en la Figura 4. Es importante resaltar que la instalación de cualquier paquete solo es necesaria la primera vez que se emplea, mientras que la activación de la librería correspondiente se debe llevar a cabo cada vez que se inicia una nueva sesión de trabajo. Otro detalle importante es que aquí se ejemplifican tres posibilidades para importar un mismo conjunto de datos (guardados originalmente en diferentes programas), solo con fines didácticos; en la práctica, lo recomendable es usar un solo tipo de programa (el de la preferencia del lector o el que esté a su alcance si no cuenta con una licencia para usar un software especializado).

```
# EJEMPLO PARA EL CASO DE UN ARCHIVO CON EXTENSIÓN *.sav
# Instalar el paquete foreign
install.packages("foreign")
# activar la librería del paquete foreign
library(foreign)
# crear el objeto dt, a partir de los datos importados
dt<-read.spss("C:/Users/ruizv/Dropbox/IDENTIDAD sub.sav",to.data.frame = T)
# sólo para el caso del archivo *.sav se ejecuta la siguiente línea, ya que e
s la que proporciona los nombres de las variables que posteriormente se usan
en la creación y estimación del modelo PLS-SEM
colnames(dt)<-c("IDVI_1","IDVI_2","IDVI_3","COMU_1","COMU_2","COMU_3","COMP_1</pre>
","COMP_2","COMP_3","COMP_4","CULT_1","CULT_2","CULT_3", "ATK_1", "ATK_2","ID
_1","ID_2","ID_3","ID_4","ACT_1","ACT_2","ACT_3","ACT_4","EXTRA_1","EXTRA_2",
"EXTRA 3")
# EJEMPLO PARA EL CASO DE UN ARCHIVO CON EXTENSIÓN *.xlsx
# Instalar el paquete readxl
install.packages("readxl")
# activar La Librería del paquete readxL
library(readx1)
# crear el objeto dt, a partir de los datos importados
dt3<-read excel("C:/Users/ruizv/Downloads/IDENTIDAD sub.xlsx",na="NA")
# EJEMPLO PARA EL CASO DE UN ARCHIVO CON EXTENSIÓN *.csv
```

```
# crear el objeto dt, a partir de Los datos importado, directamente con la fu
nción read.csv
```

```
dt2<-read.csv("C:/Users/ruizv/Downloads/IDENTIDAD_sub.csv",header = T)</pre>
```

Figura 4

Visualización del objeto dt creado a partir de los datos importados.

Un	titled1* ×	E	df×	Script_PL	S-SEM.R* ×	•	ejemplo1.Rm	nd 🗌 dt	× PLS_	SEM.Rmd	K 3	»	Environment History Connections Tutorial
	10	18	Filter					-	-	Q,			🚰 📊 📅 Import Dataset • 🌖 128 MiB • 🔏 📃 List • 🕲 •
*	IDVI_1	-	IDVI_2	IDVI_3	COMU_1	° 0	OMU_2	COMU_3	COMP_1	COMP	2 0	COMP_3	R • 🚳 Global Environment • Q
1		5	7		5	7	6	6		5	6		Data
2		6	e		5	6	6	6		6	5		💿 dt 122 obs. of 26 variables
-										-			\$ IDVI_1 : num 5 6 5 5 5 1 5 4 6 5
3		5	1		>	3	4	2		4	1		\$ IDVI_2 : num 7 6 5 5 7 4 5 6 5 6
4		5	5	1	3	4	5	4		4	4		\$ IDVI_3 : num 5653612556
5		5	7		5	4	5	3		5	5		\$ COMU_1 : num 7 6 3 4 4 2 2 6 5 6
6		1	4		1	2	3	2		2	3		\$ COMU_2 : num 6 6 4 5 5 3 6 6 5 5
-		•				2	6				2		\$ COMU_3 : num 6 6 2 4 3 2 2 4 5 3
		5				-	0			-	-		\$ COMP_1 : num 5 6 4 4 5 2 4 6 6 5
8		4	6	1	5	6	6	4		6	6		\$ COMP_2 : num 6 5 1 4 5 3 2 6 NA 6
9		6	5		5	5	5	5		6	NA		\$ COMP_3 : num 5 6 4 5 7 2 4 5 5 5
10		5	6		5	6	5	3		5	6	*	\$ COMP_4 : num 5 6 6 5 7 3 5 6 5 4
												•	<pre>\$ CULT_1 : num 4645734654 \$</pre>
Shov	ing 1 to 1	0 of 1	122 entries	, 26 total col	umns								

Especificación del modelo

En la Figura 5 se muestra la representación del modelo propuesto por Ruiz et al. (2020) sobre la identidad universitaria, como un apoyo visual para que el lector identifique los constructos que se asumieron como parte del modelo y así como la dirección de las relaciones entre ellos. En las siguientes líneas se ejemplifica la secuencia de comandos para crear el mo-

delo de medición, es decir, la especificación de la participación de cada indicador (ítem) del instrumento utilizado para la recolección de los datos en cada uno de los constructos que se representan en la Figura 5. En primer lugar, se debe instalar y activar el paquete seminr (Ray et al., 2022), que es el que permitirá realizar la construcción y estimación del modelo que se ejemplifica. Posteriormente, se crea el objeto modelo_medicion a partir de la función constructs(), la cual admite como argumentos a la función composite() un número de veces determinado por el número de constructos contemplados en el modelo, ocho en este caso. La función composite() requiere de la especificación de al menos dos argumentos: el primero para asignar el nombre a cada constructo y el segundo para señalar que se trata de un constructo compuesto por uno (single_item) o varios ítems (multi items). En el caso de que se trate un factor univariado, es suficiente con escribir el nombre de la variable correspondiente; mientras que, si se trata de un factor multivariado, se debe escribir el segmento del nombre que comparten los distintos reactivos que se incluyen en el factor, seguido del número de ítems. Téngase en cuenta que el nombre del objeto creado para alojar el modelo de medición es completamente arbitrario y que el lector puede usar el nombre que mejor le parezca. Como resultado de la ejecución de este código, en la pestaña Environment (Ventana C) debe aparecer el objeto creado para alojar el modelo de medición propuesto.

```
# Instalar el paquete SEMinR (sólo la primera vez que usa).
install.packages("seminr")
# Cargar el paquete SEMinR en el entorno de trabajo (cada vez que se inicia l
a sesión de trabajo en RStudio).
library(seminr)
#Creación del modelo de medición
modelo_medicion<-constructs(
   composite("IDVI", multi_items("IDVI_", 1:3)),
   composite("COMU", multi_items("COMU_", 1:3)),
   composite("COMP", multi_items("COMP_", 1:4)),
   composite("COMP", multi_items("CULT_", 1:3)),
   composite("ATR", multi_items("ATR_", 1:2)),
   composite("ATC", multi_items("ATR_", 1:4)),
   composite("EXTRA", multi_items("EXTRA_", 1:3)))</pre>
```

Figura 5



Representación del modelo estructural de la identidad universitaria.

Fuente: elaboración propia a partir de la propuesta de Ruiz et al. (2020)

El siguiente paso es especificar el modelo estructural, es decir, las relaciones que se asumen entre los diferentes constructos que se consideran en el modelo. En las siguientes líneas se ejemplifica el código que tendría que usarse para este propósito. La función principal es relationships(), dentro de la cual se debe incluir a la función path() tantas veces como sea necesario para establecer la dirección de las relaciones que se plantean en el modelo entre los múltiples constructos. Así, en la primera línea del ejemplo se ilustra la relación que se asume de los constructos "Identidad visual", "Comunicación", "Comportamiento organizacional" y "Cultura" y el constructo "Atractivo de la identidad". En las siguientes líneas se ejemplifican el resto de relaciones asumidas en el modelo estructural.

```
# Creación del modelo estructural
modelo_estructural <- relationships(
   paths(from = c("IDVI", "COMU","COMP","CULT"), to = c("ATR")),
   paths(from = c("ATR"), to = c("ID","ACT")),
   paths(from = c("ID"), to = c("ACT")),
   paths(from = c("ID","ACT"), to = c("EXTRA"))
   )</pre>
```

Estimación del modelo

En las siguientes líneas se muestra el código correspondiente a la estimación del modelo y los resultados que arroja el programa para la ejecución de cada segmento de instrucciones. En primer lugar, se determina mediante la función estimate_pls() cuál es el conjunto de datos sobre el que se va a evaluar el modelo propuesto, en este caso aquellos contenidos en el objeto dt. Se debe indicar también cuáles son el modelo de medición y el modelo estructural que se han de asumir para proceder con el cálculo de coeficientes y otros estadísticos; en este caso se trata de los objetos creados en el apartado previo. Un par de instrucciones importantes son las dos últimas, que permiten señalar cuál es el valor o símbolo que se debe interpretar como un dato faltante en la base de datos y cuál es el tratamiento empleado sobre esos datos faltantes; en este caso se sustituye a los NA por el valor promedio. Al ejecutar la primera parte del código, el programa devuelve un par de mensajes para informar que se genera el modelo en el paquete seminr y cuántas observaciones se consideran casos válidos para llevar a cabo los diferentes cálculos.

En las siguientes líneas se solicita la creación del resumen de resultados del modelo evaluado. A partir de tal resumen se pueden solicitar valores de estadísticos específicos, como es el caso de los coeficientes estimados para las relaciones entre constructos, así como sus valores de R cuadrada (summary_estimacion\$paths). Así mismo, es posible solicitar los valores obtenidos para los indicadores de fiabilidad y validez (summary_estimacion\$reliability).

```
# Resumen de los resultados del modelo
summary estimacion <- summary(estimacion)</pre>
# Inspección de los coeficientes para las relaciones entre constructos y sus
respectivos valores de R^2
summary_estimacion$paths
##
                      ACT EXTRA
           ATR
                  ID
## R^2 0.571 0.663 0.429 0.407
## AdiR^2 0.556 0.660 0.419 0.397
## IDVI
        0.256
## COMU
        0.084
## COMP 0.248
## CULT 0.304
          . 0.814 0.182
## ATR
## ID
             . 0.498 0.148
## ACT
                         . 0.532
# Inspección de la fiabilidad y validez
summary_estimacion$reliability
        alpha rhoC AVE rhoA
##
## IDVI 0.669 0.820 0.606 0.704
## COMU 0.810 0.888 0.725 0.811
## COMP 0.790 0.864 0.614 0.808
## CULT 0.764 0.861 0.675 0.821
## ATR 0.899 0.952 0.908 0.899
## ID
        0.899 0.930 0.768 0.908
## ACT 0.886 0.921 0.745 0.900
## EXTRA 0.860 0.914 0.779 0.881
##
## Alpha, rhoC, and rhoA should exceed 0.7 while AVE should exceed 0.5
```

Una de las características del método PLS-SEM es que carece de la posibilidad de llevar a cabo una evaluación de la bondad de ajuste del modelo, debido principalmente a que no se asume una distribución particular para los estadísticos que se calculan. En su lugar, se emplea el método bootstrapping para generar muestreos mediante un proceso iterativo y calcular a partir de los valores obtenidos en cada muestra una serie de valores que se sirven de referencia para establecer la media, la desviación estándar, intervalo de confianza, etcétera, para los valores originalmente estimados para los coeficientes del modelo estructural. ## Bootstrapping model using seminr...

SEMinR Model successfully bootstrapped

Resumen de los resultados del bootstrapping summary_boot_estimacion <- summary(boot_estimacion) # Inspección de los valores estimados a partir del bootstraping summary_boot_estimacion\$bootstrapped_paths

##		Original Est.	Bootstrap Mean	Bootstrap SD	T Stat.	2.5% CI
##	IDVI -> ATR	0.256	0.258	0.096	2.678	0.073
##	COMU -> ATR	0.084	0.077	0.089	0.942	-0.107
##	COMP -> ATR	0.248	0.259	0.094	2.628	0.084
##	CULT -> ATR	0.304	0.303	0.095	3.206	0.108
##	ATR -> ID	0.814	0.813	0.036	22.752	0.738
##	ATR -> ACT	0.182	0.185	0.169	1.080	-0.134
##	ID -> ACT	0.498	0.499	0.159	3.131	0.185
##	ID -> EXTRA	0.148	0.154	0.085	1.737	-0.015
##	ACT -> EXTRA	0.532	0.532	0.082	6.496	0.365
##		97.5% CI				
##	IDVI -> ATR	0.443				
##	COMU -> ATR	0.246				
##	COMP -> ATR	0.449				
##	CULT -> ATR	0.489				
##	ATR -> ID	0.878				
##	ATR -> ACT	0.514				
##	ID -> ACT	0.791				
##	ID -> EXTRA	0.322				
##	ACT -> EXTRA	0.691				

Inspección de Las cargas factoriales de cada ítem, según los indicadores ob tenidos en el proceso de boostrapping summary boot estimacion\$bootstrapped loadings

## CT				Original Est.	Bootstrap Mean	Bootstrap SD	T Stat.	2.5%
## 88	IDVI_1	->	IDVI	0.863	0.862	0.030	29.145	0.7
## 20	IDVI_2	->	IDVI	0.821	0.818	0.044	18.686	0.7
## 45	IDVI_3	->	IDVI	0.632	0.630	0.082	7.740	0.4
## 07	COMU_1	->	COMU	0.866	0.867	0.028	31.463	0.8
##	COMU_2	->	COMU	0.842	0.844	0.041	20.461	0.7

53				
## COMU_3 -> COMU	0.846	0.846	0.032 26.119	0.7
74				
## COMP_1 -> COMP 21	0.696	0.690	0.075 9.272	0.5
## COMP_2 -> COMP	0.785	0.782	0.048 16.270	0.6
## COMP_3 -> COMP	0.843	0.845	0.027 31.199	0.7
## COMP_4 -> COMP	0.803	0.803	0.040 19.993	0.7
13 ## CULT_1 -> CULT	0.869	0.871	0.019 45.637	0.8
30 ## CULT_2 -> CULT	0.879	0.876	0.030 29.067	0.8
06 ## CULT_3 -> CULT	0.707	0.701	0.073 9.680	0.5
45 ## ATR_1 -> ATR	0.953	0.953	0.010 96.021	0.9
29				
## ATR_2 -> ATR	0.953	0.953	0.009 106.528	0.9
34				
## ID_1 -> ID	0.849	0.850	0.026 32.211	0.7
## ID_2 -> ID	0.831	0.830	0.046 18.165	0.7
25 ## ID_3 -> ID	0.902	0.904	0.018 48.868	0.8
## ID_4 -> ID	0.919	0.919	0.015 61.963	0.8
## ACT_1 -> ACT	0.880	0.875	0.035 24.844	0.7
## ACT_2 -> ACT	0.902	0.902	0.016 55.520	0.8
## ACT_3 -> ACT	0.881	0.879	0.027 32.076	0.8
## ACT_4 -> ACT	0.785	0.775	0.066 11.855	0.6
## EXTRA_1 -> EXTRA	0.887	0.888	0.019 45.755	0.8
48 ## EXTRA_2 -> EXTRA	0.899	0.898	0.033 27.556	0.8
## EXTRA_3 -> EXTRA	0.862	0.858	0.040 21.315	0.7
## O	7 5% CT			
ידער איז	0 906			
$\frac{1}{2} \frac{1}{2} \frac{1}$	0.900			
$\frac{1}{2} = \frac{1}{2} = \frac{1}$	0.365			
## COMU 1 -> COMU	0.914			
	~ • ~ - 1			

COMU_2 -> COMU ## COMU_3 -> COMU

0.912 0.900

<pre>## COMP_1 -> COMP ## COMP_2 -> COMP ## COMP_3 -> COMP ## COMP_4 -> COMP ## CULT_1 -> CULT ## CULT_2 -> CULT ## CULT_3 -> CULT ## ATR_1 -> ATR ## ATR_2 -> ATR ## ID_1 -> ID ## ID_2 -> ID ## ID_3 -> ID ## ID_4 -> ID ## ACT_1 -> ACT ## ACT_1 -> ACT ## ACT_2 -> ACT</pre>	0.812 0.861 0.889 0.869 0.904 0.925
<pre>## COMP_2 -> COMP ## COMP_3 -> COMP ## COMP_4 -> COMP ## CULT_1 -> CULT ## CULT_2 -> CULT ## CULT_3 -> CULT ## ATR_1 -> ATR ## ATR_2 -> ATR ## ID_1 -> ID ## ID_2 -> ID ## ID_3 -> ID ## ID_4 -> ID ## ACT_1 -> ACT ## ACT_1 -> ACT ## ACT_2 -> ACT</pre>	0.861 0.889 0.869 0.904 0.925
<pre>## COMP_3 -> COMP ## COMP_4 -> COMP ## CULT_1 -> CULT ## CULT_2 -> CULT ## CULT_3 -> CULT ## ATR_1 -> ATR ## ATR_2 -> ATR ## ID_1 -> ID ## ID_2 -> ID ## ID_3 -> ID ## ID_4 -> ID ## ACT_1 -> ACT ## ACT_2 -> ACT</pre>	0.889 0.869 0.904 0.925
<pre>## COMP_4 -> COMP ## CULT_1 -> CULT ## CULT_2 -> CULT ## CULT_3 -> CULT ## ATR_1 -> ATR ## ATR_2 -> ATR ## ID_1 -> ID ## ID_2 -> ID ## ID_3 -> ID ## ID_4 -> ID ## ACT_1 -> ACT ## ACT_2 -> ACT</pre>	0.869 0.904 0.925
<pre>## CULT_1 -> CULT ## CULT_2 -> CULT ## CULT_3 -> CULT ## ATR_1 -> ATR ## ATR_2 -> ATR ## ID_1 -> ID ## ID_2 -> ID ## ID_3 -> ID ## ID_4 -> ID ## ACT_1 -> ACT ## ACT_2 -> ACT</pre>	0.904 0.925
<pre>## CULT_2 -> CULT ## CULT_3 -> CULT ## ATR_1 -> ATR ## ATR_2 -> ATR ## ID_1 -> ID ## ID_2 -> ID ## ID_3 -> ID ## ID_4 -> ID ## ACT_1 -> ACT ## ACT_2 -> ACT</pre>	0.925
<pre>## CULT_3 -> CULT ## ATR_1 -> ATR ## ATR_2 -> ATR ## ID_1 -> ID ## ID_2 -> ID ## ID_3 -> ID ## ID_4 -> ID ## ACT_1 -> ACT ## ACT_2 -> ACT</pre>	
<pre>## ATR_1 -> ATR ## ATR_2 -> ATR ## ID_1 -> ID ## ID_2 -> ID ## ID_3 -> ID ## ID_4 -> ID ## ACT_1 -> ACT ## ACT_2 -> ACT</pre>	0.823
<pre>## ATR_2 -> ATR ## ID_1 -> ID ## ID_2 -> ID ## ID_3 -> ID ## ID_4 -> ID ## ACT_1 -> ACT ## ACT_2 -> ACT</pre>	0.968
<pre>## ID_1 -> ID ## ID_2 -> ID ## ID_3 -> ID ## ID_4 -> ID ## ACT_1 -> ACT ## ACT_2 -> ACT</pre>	0.967
## ID_2 -> ID ## ID_3 -> ID ## ID_4 -> ID ## ACT_1 -> ACT ## ACT_2 -> ACT	0.898
## ID_3 -> ID ## ID_4 -> ID ## ACT_1 -> ACT ## ACT_2 -> ACT	0.903
## ID_4 -> ID ## ACT_1 -> ACT ## ACT_2 -> ACT	0.936
## ACT_1 -> ACT ## ACT_2 -> ACT	0.944
## ACT_2 -> ACT	0.930
	0.930
## ACT_3 -> ACT	0.923
## ACT_4 -> ACT	0.878
## EXTRA_1 -> EXTRA	0.926
## EXTRA_2 -> EXTRA	0.948
## EXTRA_3 -> EXTRA	

Finalmente, en la Figura 6 se ilustra el resultado de ejecutar la función plot() en el contexto de la estimación del modelo estructural con el método PLS-SEM.

plot(summary_estimacion\$reliability)



Visualización de los indicadores de fiabilidad para el modelo estructural de la identidad universitaria.



Conclusiones

El objetivo del presente capítulo fue desarrollar una guía lo más clara posible para llevar a cabo el modelado de ecuaciones estructurales con el método de mínimos cuadrados parciales empleando software libre. Concretamente, se desarrolló la actividad didáctica empleando el software R debido a sus características de ser libre, de código abierto y de una lógica de programación relativamente fácil de aprender y de ejecutar. Como pudo notarse en el cuerpo del capítulo, los comandos incluidos en el paquete seminr son lo suficientemente intuitivos como para permitir al usuario el desarrollo de sus propios modelos a partir de unas cuantas funciones.

Consideramos oportuno resaltar un par de aspectos en relación con el tópico que se abordó en el presente capítulo y con la metodología para el análisis llevado a cabo. Por un lado, es una gran ventaja emplear una lógica de programación como la de R, basada en scripts, lo cual facilita la posibilidad de replicar el análisis de datos llevado a cabo de una manera completamente idéntica al análisis llevado a cabo por el autor del script. Cualquier lector puede descargar del sitio https://github.com/ruizvja/datos_ejemplo_PLS-SEM la copia del script que los autores diseñaron para ejemplificar el uso de R para llevar a cabo el análisis correspondiente al modelo estructural de la identidad universitaria con el método PLS-SEM. Por otro lado, el uso de lenguajes de programación como R no resuelve las dudas planteadas por algunos autores sobre la falta de investigación estadística sobre los procedimientos empleados en la lógica del PLS-SEM, ni de los resultados que este método arroja (Westland, 2019), aunque indudablemente potencia la eficiencia del investigador para llevar a cabo los análisis correspondientes a este tipo de técnica.

Finalmente, consideramos que la información aquí presentada es útil, aunque no es exhaustiva en cuanto a todo lo que implica la aplicación del método PLS-SEM, pero es un intento por acercar el uso de lenguajes de programación versátiles y de licencia libre a la comunidad de investigadores en ciencias administrativas, quienes podrían encontrar un beneficio de esta labor.

Referencias

- Adler, S. J., Sharma, P. N., & Radomir, L. (2023). Toward open science in PLS-SEM: Assessing the state of the art and future perspectives. *Journal of Business Research*, 169, 114291. https://doi.org/10.1016/j. jbusres.2023.114291
- Ali, M., Lei, S., & Hussain, S. T. (2017). Relationship of external knowledge management and performance of Chinese Manufacturing Firms: The mediating role of talent management. *International Business Research*, 10(6), 248. https://doi.org/10.5539/ibr.v10n6p248
- Barnidge, M., & De Zúñiga, H. G. (2017). Amos (software). The International Encyclopedia of Communication Research Methods, 1–3. https://doi. org/10.1002/9781118901731.iecrm0003
- Bentler, P. M. (1985). Theory and implementation of EQS: A structural equations program. BMDP Statistical Software. Multivariate Software Inc.
- Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2019). *Multivariate data analysis*. 8th edition. Cengage.

- Hair, J. F., Hult, G. T., Ringle, C. M., Sarstedt, M., Danks, N. P., & Ray, S. (2021). Partial least squares structural equation modeling (PLS-SEM) using R. A workbook. Springer.
- Jöreskog, K. G. (1970). A general method for estimating a linear structural equation system. *ETS Research Bulletin Series*, 1970(2). https://doi. org/10.1002/j.2333-8504.1970.tb00783.x
- Jőreskog, K. G., & van Thiilo, M. (1972). Lisrel a general computer program for estimating a linear structural equation system involving multiple indicators of unmeasured variables. *ETS Research Bulletin Series*, 1972(2). https://doi.org/10.1002/j.2333-8504.1972.tb00827.x
- Khan, G. F., Sarstedt, M., Shiau, W.-L., Hair, J. F., Ringle, C. M., & Fritze, M. P. (2019). Methodological research on partial least squares structural equation modeling (PLS-SEM). *Internet Research*, 29(3), 407–429. https://doi.org/10.1108/intr-12-2017-0509
- Máynez-Guaderrama, A.I. & Vargas-Salgado, M.M. (2019). Modelos de ecuaciones estructurales mediante mínimos cuadrados parciales: un ejemplo de su aplicación en el campo de las ciencias administrativas. En: A. I. Maynez-Guaderrama y V. G. López-Torres (Coords.), *La práctica de la investigación en las ciencias administrativas* (pp. 113-135). Pearson Educación.
- R Core Team (2022). foreign: Read Data Stored by 'Minitab','S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ... R package version 0.8-84, <https://CRAN.R-project.org/package=foreign>.
- Ray, S., Danks, N., & Calero-Valdez A (2022). seminr: Building and Estimating Structural Equation Models. R package version 2.3.2, <https:// CRAN.R-project.org/package=seminr>.
- Ruiz, J. A., Bermúdez, K. A., & González-García, D. A. (2020). Adaptación y validación de un instrumento para evaluar la identidad universitaria. En: E. Vargas, S. Cruz y V.G. López (Comps.), *Interdisciplina en temas de relevancia social* (pp. 81-91). MA Porrúa.
- Sharma, R. K., & Kaur, S. (2024). Analysing the mediating role of organisational citizenship behaviour between Transformational Leadership and Education 4.0 using PLS-SEM approach. *International Journal* of Educational Management, 38(2), 391–412. https://doi.org/10.1108/ ijem-07-2023-0322

- Soriano, J. L., & Mejía-Trejo, J. (2022). Modelado de Ecuaciones Estructurales en el Campo de las ciencias de la administración. Revista de Métodos Cuantitativos para la Economía y la Empresa. https://doi.org/10.46661/ revmetodoscuanteconempresa.5414
- Takaki, M., Bravo, R., & Martínez, E. (2015). La Gestión de la Identidad Corporativa en la Universidad: Análisis Y consecuencias desde la perspectiva del profesorado. Revista Europea de Dirección y Economía de La Empresa, 24(1), 25–34. https://doi.org/10.1016/j.redee.2014.05.001
- Westland, C.J. (2019). Structural equation models. From paths to networks. Springer
- Wright, S. (1920). The relative importance of heredity and environment in determining the piebald pattern of Guinea-Pigs. *Proceedings of the National Academy of Sciences*, 6(6), 320–332. https://doi.org/10.1073/ pnas.6.6.320

Modelos de ecuaciones estructurales como técnica de análisis en estudios empíricos Se terminó de editar en diciembre de 2024 en los talleres gráficos de Astra Ediciones Av. Acueducto 829, Colonia Santa Margarita, C. P. 45140, Zapopan, Jalisco, México E-mail: edicion@astraeditorial.com.mx www.astraeditorialshop.com En este libro se describen aplicaciones del uso de PLS-SEM de una forma simple y detallada para que futuros investigadores puedan replicar esta técnica en sus investigaciones y, además, los capítulos de este libro sirvan como guía a otros investigadores que necesiten familiarizarse más con su aplicación.

La obra, también presenta investigaciones en ciencias administrativas que contribuyen a la literatura y, sus resultados son valiosos para los tomadores de decisiones en las áreas de finanzas, mercadotecnia, manufactura, administración educativa, recursos humanos y emprendimiento.





