



SEM.NestedData1: Adjusting lavaan results for complex survey design using lavaan.survey.

Jim Grace
USGS

U.S. Department of the Interior
U.S. Geological Survey

1

This module addresses an important question, “What should I do when the data are not a simple random sample?” While there are many variations of, and possible approaches to, this problem, here I summarize a simple approach designed for those using the lavaan package for estimating SE models. This approach uses a companion package, “lavaan.survey”, which implements an adjustment procedure.

Key reference for this material is:

Oberski, D. 2014. lavaan.survey: An R package for complex survey analysis of structural equation models.

Notes: IP-56512; Support provided by USGS Climate & Land Use R&D and Ecosystems Programs. Thanks to Diann Prosser, Chris Patrick, and other members of the “NOAA Shorelines Group” for input on this tutorial, as well as example data. Formal review of the material from which this tutorial was derived was provided by Jesse Miller and Phil Hahn, Univ. Wisconsin. Any use of trade, form, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Last revised 17.02.26.

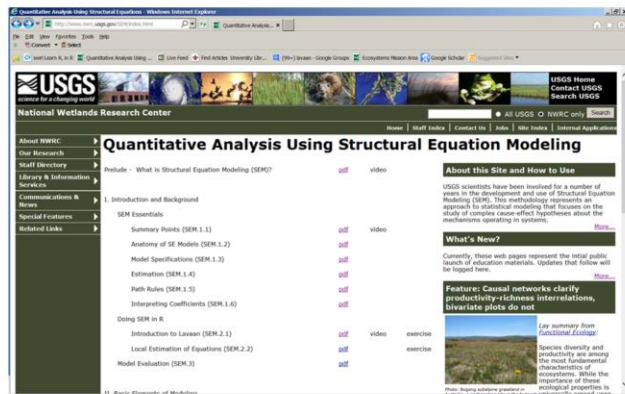
Source: <https://www.usgs.gov/centers/wetland-and-aquatic-research-center/science/quantitative-analysis-using-structural-equation>

A Quick Synopsis

It is possible to adjust lavaan results for data nesting and other complexities using “lavaan.survey”.

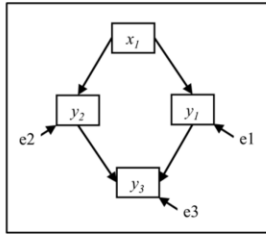
Note: if you are not familiar with lavaan objects or need a refresher, go to <http://www.nwrc.usgs.gov/SEM> and look at tutorial SEM2.1 “Introduction to Lavaan”.

A one-page lavaan refresher is presented on the next page.



Briefly, lavaan.survey is designed to work with lavaan objects. lavaan is an R package that implements classical “global-estimation” approaches to SEM. My website has a bit of information on modeling with lavaan, as well as links to additional resources.

Quick lavaan refresher: Specifying and estimating a model.



```
# Step 1: Specify model equations
```

```
mod.1 <- 'y1 ~ x1
          y2 ~ x1
          y3 ~ y1 + y2'
```

```
# Step 2: Estimate model using the 'sem' function
```

```
mod.1.fit <- sem(mod.1, data=k4.dat)
```

```
# Step 3: Extract results
```

```
summary(mod1.fit)
```



3

This is slide 6 from “SEM.2.1-Intro to Lavaan – version 1.0”.

Note: there are some preliminary steps, of course.

(1) You need the variables that are in the model included in a data object, which in this example is named “k4.dat”.

(2) You need to load the lavaan library

```
# Load lavaan library
library(lavaan)
```

Regarding this slide:

- 1) Specifying a model simply involves an equation for each response variable in the model.
- 2) The “sem” function is used to “fit” the model. This process creates a “fit object”, in this case “mod.1.fit” that contains numerous results.
- 3) You can extract some of the main results from the fit object using the “summary()” function.

What is “lavaan.survey”?



Journal of Statistical Software

March 2014, Volume 57, Issue 1.

<http://www.jstatsoft.org/>

lavaan.survey: An R Package for Complex Survey Analysis of Structural Equation Models

Daniel Oberski
Tilburg University

Abstract

This paper introduces the R package **lavaan.survey**, a user-friendly interface to design-based complex survey analysis of structural equation models (SEMs). By leveraging existing code in the **lavaan** and **survey** packages, the **lavaan.survey** package allows for SEM analyses of stratified, clustered, and weighted data, as well as multiply imputed complex survey data. **lavaan.survey** provides several features such as SEMs with replicate weights, a variety of resampling techniques for complex samples, and finite population corrections, features that should prove useful for SEM practitioners faced with the common situation of a sample that is not *iid*.

Keywords: complex survey analysis, structural equation modeling, clustering, stratification, sampling weights, multiple imputation, resampling, jackknife, bootstrap, replicate weights, R.



4

Direct link to the paper can be found at
<http://www.jstatsoft.org/v57/i01/paper>

How do you use “lavaan.survey”?

```
# Assuming we have a model fit object (mod.1.fit)
mod.1.fit <- sem(mod.1, data=k4.dat)

# Adjust for Nested Design
# first, describe the design using “svydesign”
# (imagine obs are nested within sites)

design <- svydesign(ids = ~sites, nest=TRUE,
data = k4.dat)

# second, post-process the lavaan object
fit.adj <- lavaan.survey(lavaan.fit = mod.1.fit,
survey.design = design)

# third, request adjusted results
summary(fit.adj, standardized=T)
```



End of Synopsis

5

Remember, before using any R package, you need to load it. So,

```
# Load lavaan.survey library
library(lavaan.survey)
```

There are additional commands we can use for extracting results from a post-processed object. For example,

```
fit.adj # gives chi-square
modindices(fit.adj) # extracts mod indices
resid(fit.adj, type="standardized") # resids
```

An Ecological Example

“The Shorelines Project”

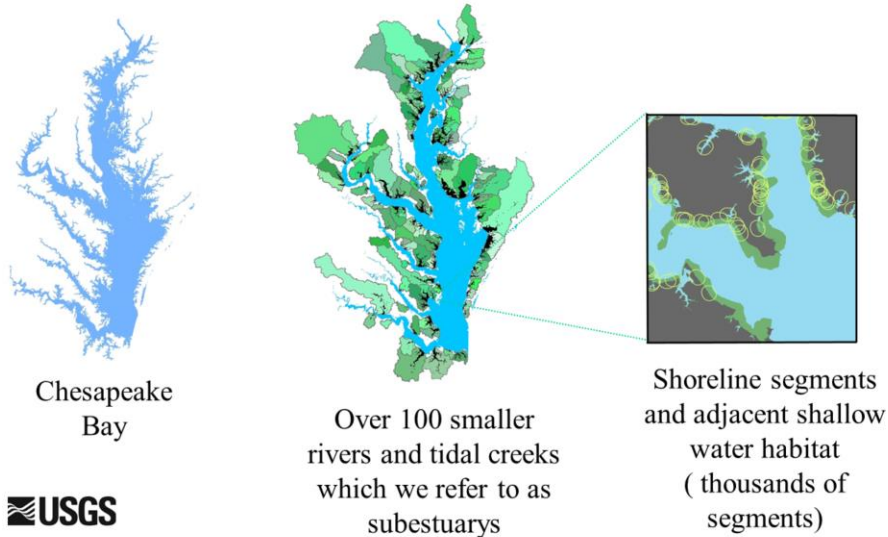
Thomas Jordan, Smithsonian Environmental Research Center
and many, many others.



6

Special thanks to the Shorelines Project folks for suggesting this tutorial and for providing the modeling questions and associated data.

Spatial Scales of Data: samples within subestuaries



Here is a slide from a presentation given by Chris Patrick, Smithsonian Environmental Research Center. This illustrates the conceptual hierarchy of samples being collected at locations that fall within larger spatial units, “subestuaries”.

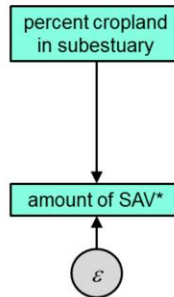
As far as nesting effects goes, this is perhaps one of those cases where the grouping variable, “subestuary” may or may not impose a major grouping effect on the sample. It behooves us to consider this as cluster sampling, nonetheless.

A question about submerged aquatic vegetation:
Is it related to the amount of cropland in the
subestuary?



*SAV = submersed aquatic vegetation,
epsilon = influence of other factors.

Two-level data:
I. 112 subestuaries (each with 1
cropland estimate).
II. 8069 segments nested in
subestuaries (each with an SAV est.)

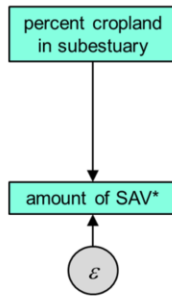


8

More about the sav research can be found in:

Patrick, C.J., Weller, D.E., Li, X., and Ryder, M. 2014. Effects of Shoreline Alteration and Other Stressors on Submerged Aquatic Vegetation in Subestuaries of Chesapeake Bay and the Mid-Atlantic Coastal Bays. Estuaries and Coasts.

A lavaan approach to this simple model.



```
# Step 1: Specify model equations  
savmod <- 'sav ~ crop'
```

```
# Step 2: Estimate model using the 'sem' function  
savmod.fit <- sem(savmod, data=sav1.dat,  
  meanstructure=T)
```



9

Note that this code is in the R file “SEM.NestedData_code.R”.

Adjusting for sample design.

```
# Step 3: Describing sampling using "svydesign"

design1 <- svydesign(ids = ~subest, nest=TRUE,
  data=sav1.dat)
```

```
# Step 4: Post-processing original lavaan object

savmod.adj <- lavaan.survey(lavaan.fit =
  savmod.fit, survey.design = design1)
```

```
# Step 5: Get summary of adjusted results

summary(savmod.adj, rsq=T)
```



10

A key step in the adjustment process is to describe the sampling design for `lavaan.survey` to use. There are lots of possibilities and one should consult Oberski (2014) to understand the greater range of potentials for this function. Here we have the simplest case, samples are nested within “subest” (subestuaries).

Once we have the design described, we use it in the “`lavaan.survey`” function.

Comparing Unadjusted to Adjusted Output

# Unadjusted Results (from savmod)				
	Estimate	Std.err	Z-value	P(> z)
Regressions:				
sav ~				
crop	0.005	0.000	20.543	0.000
Intercepts:				
sav	0.109	0.005	22.399	0.000
Variances:				
sav	0.079	0.001		
R-Square:				
sav	0.050			
# Adjusted Results (from savmod.adj)				
	Estimate	Std.err	Z-value	P(> z)
Regressions:				
sav ~				
crop	0.005	0.002	2.828	0.005
Intercepts:				
sav	0.109	0.037	2.997	0.003
Variances:				
sav	0.079	0.010		
R-Square:				
sav	0.050			

Parameter estimates unchanged.

Only Z-values and Std.errs get adjusted when nesting accounted for.

There has been an adjustment of the “effective sample size” (just like with spatial autocorrelation issue)



11

Here I first show results returned by the command:

```
summary(savmod, rsq=T)
```

I don't show the model fit statistics in this case because the model is saturated and, thus, fit is not computed.

Comparing the unadjusted results to the adjusted ones obtained by

```
summary(savmod.adj, rsq=T)
```

is interesting. Note that the parameter estimates are unchanged for this case (no unequal weighting of samples). What is very different is the standard errors, which are much bigger after adjustment. The reason? Nesting creates non-independence in the data and one effect of that is an effective sample size that is much smaller than the actual sample. This issues is also found with spatial autocorrelation.

Additional Information

The general topic of “complex sample design” relates to another current hot topic in quantitative analysis, “multi-level” or “hierarchical” modeling.

I will cover some other approaches to hierarchical modeling elsewhere.

References related to SEM include:

Hox, J.J. 2010. Multilevel Analysis. Routledge Publishers.

Bollen et al. 2013. Issues in the Structural Equation Modeling of Complex Survey Data. Proc. 59th Int. Stats. Inst. World Stats. Cong. (<http://www.2013.isiproceedings.org/Files/STS010-P1-S.pdf>)



12

Here, my treatment of the subject is very brief. More information can be found in Bollen’s paper. Beyond that, there is a large literature on this topic.