

modsem: An R Package for Estimating Latent Interactions and Quadratic Effects

Kjell S Slupphaug

modsem

- ▶ `modsem` is an R package for estimating latent interaction and quadratic effects, in *Structural Equation Models* (SEMs).

Frameworks

- ▶ Product Indicator (PI) approaches
- ▶ Distribution Analytic (DA) approaches

Product Indicator (PI) Approaches

Product Indicator (PI) Approaches

- ▶ First attempt at latent interaction in SEMs (Kenny and Judd 1984)
- ▶ Creates product indicators, used as indicators for latent interaction terms.
- ▶ Has traditionally required manual model specification, and manual construction of product indicators.
- ▶ Early approaches involved complicated model constraints.
- ▶ Manual specification (partially) led to a continual simplification of constraints.

Product Indicator (PI) Approaches (History)

- ▶ Constrained Approach (orthogonal specification) (Kenny and Judd 1984)
- ▶ Constrained Approach (oblique specification) (Jöreskog and Yan 1996)
- ▶ Constrained Approach (mean-centered indicators) (Algina and Moulder 2001)
- ▶ Unconstrained Approach (constrained latent mean) (Martin and Marsh 1999)
- ▶ Residual Centering Approach (no constraints) (Little, Bovaird, and Widaman 2006)
- ▶ Double Centering Approach (no constraints) (Lin et al. 2010)

Product Indicator (PI) Approaches (Software)

- ▶ `semTools` offers tools for creating the product indicators, but does not help specify the model.
- ▶ Thus the PI approaches largely require manual specification.
- ▶ `modsem` automatically handles the creation of product indicators, and model specification.
- ▶ Model specification becomes exponentially more complicated for models with more indicators, more interaction terms. Especially for the PI approaches with model constraints.

Example: Double Centering Approach, using semTools

```
model <- '  
# Measurement Model  
X  =~ x1 + x2 + x3  
Z  =~ z1 + z2 + z3  
Y  =~ y1 + y2 + y3  
XZ =~ x1.z1 + x2.z1 + x3.z1 +  
      x1.z2 + x2.z2 + x3.z2 +  
      x1.z3 + x2.z3 + x3.z3  
  
# Structural Model  
Y ~ X + Z + XZ  
  
# Residual Covariances  
x1.z1 ~~ x1.z2 + x1.z3 + x2.z1 + x3.z1  
x1.z2 ~~ x1.z3 + x2.z2 + x3.z2  
x2.z1 ~~ x2.z2 + x2.z3 + x3.z1  
  
x1.z3 ~~ x2.z3 + x3.z3  
x2.z2 ~~ x2.z3 + x3.z2  
x3.z1 ~~ x3.z2 + x3.z3  
  
x2.z3 ~~ x3.z3  
x3.z2 ~~ x3.z3  
,
```


Example: Double Centering Approach, using semTools

```
library(semTools)
data.prod <- indProd(data = oneInt,
                     var1 = c("x1", "x2", "x3"),
                     var2 = c("z1", "z2", "z3"),
                     match = FALSE)

fit <- sem(model, data = data.prod)
summary(fit)
```

Truncated Output:

Regressions:

	Estimate	Std.Err	z-value	P(> z)
Y ~				
X	0.675	0.027	25.379	0.000
Z	0.561	0.026	21.606	0.000
XZ	0.702	0.027	26.360	0.000

Example: Double Centering Approach, using modsem

```
library(modsem)

model <- '
# Measurement Model
  X =~ x1 + x2 + x3
  Z =~ z1 + z2 + z3
  Y =~ y1 + y2 + y3

# Structural Model
  Y ~ X + Z + X:Z
'

fit <- modsem(model, data = oneInt)
summary(fit)
```

Truncated Output:

Regressions:

	Estimate	Std.Err	z-value	P(> z)
Y ~				
X	0.675	0.027	25.379	0.000
Z	0.561	0.026	21.606	0.000
XZ	0.702	0.027	26.360	0.000

Distribution Analytic (DA) Approaches

Distribution Analytic (DA) Approaches

- ▶ Latent Moderated Structural Equations (LMS) Method, introduced by (A. Klein and Moosbrugger 2000).
- ▶ Quasi Maximum Likelihood (QML) Method, introduced by (A. G. Klein and Muthén 2007).
- ▶ No formation of product indicators.
- ▶ Non-normal distribution of η is explicitly modelled.
- ▶ LMS and QML perform very similarly. But QML performs slightly better under normality violations.
- ▶ QML is less computationally intensive, with multiple moderators.

Distribution Analytic (DA) Approaches (Software)

- ▶ R package nlsem.
- ▶ Mplus
- ▶ modsem

nlsem

- ▶ Pros:
 - ▶ Free open source R package.
 - ▶ Has both LMS and QML approaches.
- ▶ Cons:
 - ▶ Very slow
 - ▶ Hard to use
 - ▶ Only handles models with a single endogenous variable (η).
 - ▶ Doesn't handle missing data
 - ▶ Biased LMS standard errors
 - ▶ Unstable estimates
 - ▶ Cannot produce standardized estimates
 - ▶ Virtually no fit measures
 - ▶ Does not handle linear and nonlinear constraints.
 - ▶ Generally unpolished

Mplus

- ▶ Pros:
 - ▶ Extended implementation of LMS approach (multiple endogenous variables, endogenous interactions)
 - ▶ Integrates well with other Mplus features (e.g., Multilevel SEM)
 - ▶ Handles three-way interaction terms
 - ▶ Handles higher order models (and interactions)
 - ▶ Easy to use
 - ▶ Full Information Maximum Likelihood (FIML) estimation.
 - ▶ Linear and nonlinear constraints.
- ▶ Cons:
 - ▶ Scales poorly with multiple interaction effects
 - ▶ Does not include QML
 - ▶ Proprietary (expensive) software package.
 - ▶ No Multigroup LMS.

modsem

- ▶ Pros:
 - ▶ Free open source R package.
 - ▶ Extended implementation of LMS and QML approach (multiple endogenous variables, endogenous interactions).
 - ▶ Computational optimized implementation of LMS and QML approaches.
 - ▶ Better scaling in LMS approach, compared to Mplus.
 - ▶ Handles higher order models (and interactions)
 - ▶ Easy to use
 - ▶ Full Information Maximum Likelihood (FIML) estimation.
 - ▶ Multigroup LMS and QML.
 - ▶ Linear and nonlinear constraints.
- ▶ Cons:
 - ▶ Missing some advanced features available in Mplus (e.g., Multilevel SEM).

Example: LMS approach, using nlsem (213.2 seconds)

```
library(nlsem)
set.seed(3248927)

model <- "
  X =~ x1 + x2 + x3
  Z =~ z1 + z2 + z3
  Y =~ y1 + y2 + y3

  Y ~ X + Z + X:Z
"

nlsem_model_1 <- lav2nlsem(model)
oneIntSorted <- oneInt[c("x1", "x2", "x3", "z1", "z2",
                        "z3", "y1", "y2", "y3")]
start <- runif(count_free_parameters(nlsem_model_1))

specs <- as.data.frame(nlsem_model_1)
specs[grepl("tau|alpha", specs$label), "class1"] <- 0
specs[grepl("nu\\. (x|y)", specs$label), "class1"] <- NA

nlsem_model_2 <- create_sem(specs)

fit_nlsem <- em(nlsem_model_2, data = oneIntSorted, m = 16,
               convergence = 1e-4, max.iter = 500, start = start)
```

Example: LMS approach, using nlsem (213.2 seconds)

```
summary(fit_nlsem)
```

Truncated Output:

	Estimate	Std. Error	z value	Pr(> z)
Gamma1	0.6750993	0.03022066	22.33900	1.544561e-110
Gamma2	0.5965296	0.02516093	23.70857	2.941752e-124
Omega3	0.7269645	0.02676991	27.15603	2.149299e-162

Example: LMS approach, using Mplus (4.2 seconds)

```
model <- '  
# Measurement Model  
  X =~ x1 + x2 + x3  
  Z =~ z1 + z2 + z3  
  Y =~ y1 + y2 + y3  
  
# Structural Model  
  Y ~ X + Z + X:Z  
,  
  
fit <- modsem(model, data = oneInt, method = "mplus")  
summary(fit)
```

Truncated Output:

Regressions:

	Estimate	Std.Error	z.value	P(> z)
Y ~				
X	0.673	0.031	21.666	0.000
Z	0.569	0.030	18.710	0.000
X:Z	0.718	0.028	25.828	0.000

Example: LMS approach, using modsem (1.2 seconds)

```
model <- '  
# Measurement Model  
  X =~ x1 + x2 + x3  
  Z =~ z1 + z2 + z3  
  Y =~ y1 + y2 + y3  
  
# Structural Model  
  Y ~ X + Z + X:Z  
,  
  
fit <- modsem(model, data = oneInt, method = "lms")  
summary(fit)
```

Truncated Output:

Regressions:

	Estimate	Std.Error	z.value	P(> z)
Y ~				
X	0.673	0.031	21.666	0.000
Z	0.569	0.030	18.710	0.000
X:Z	0.718	0.028	25.828	0.000

Visualizing Interaction Effects

Example Model

```
model <- '
  visual  =~ x1 + x2 + x3
  textual =~ x4 + x5 + x6
  speed   =~ x7 + x8 + x9

  visual ~ speed + textual + speed:textual
'

fit <- modsem(model, data = HolzingerSwineford1939, method = "lms")
summary(fit)
```

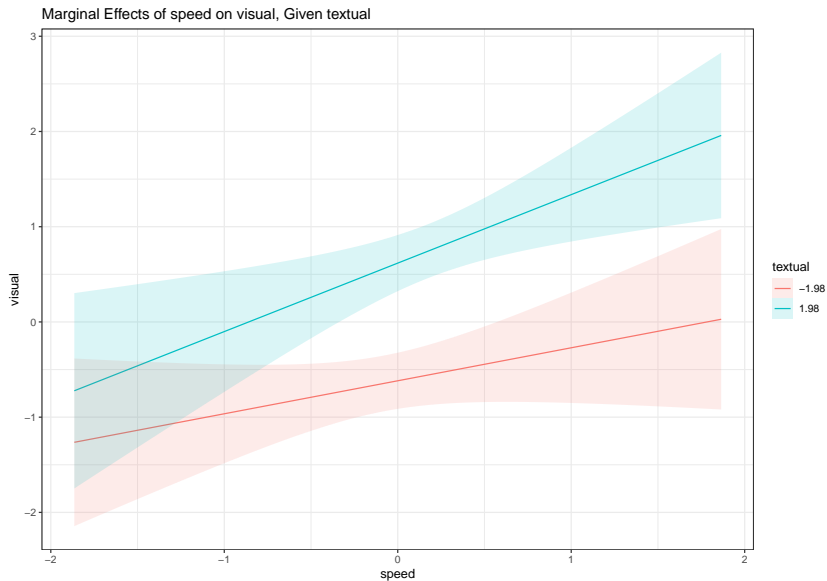
Truncated Output:

Regressions:

	Estimate	Std.Error	z.value	P(> z)
visual ~				
textual	0.312	0.076	4.107	0.000
speed	0.533	0.158	3.366	0.001
speed:txtl	0.094	0.093	1.016	0.309

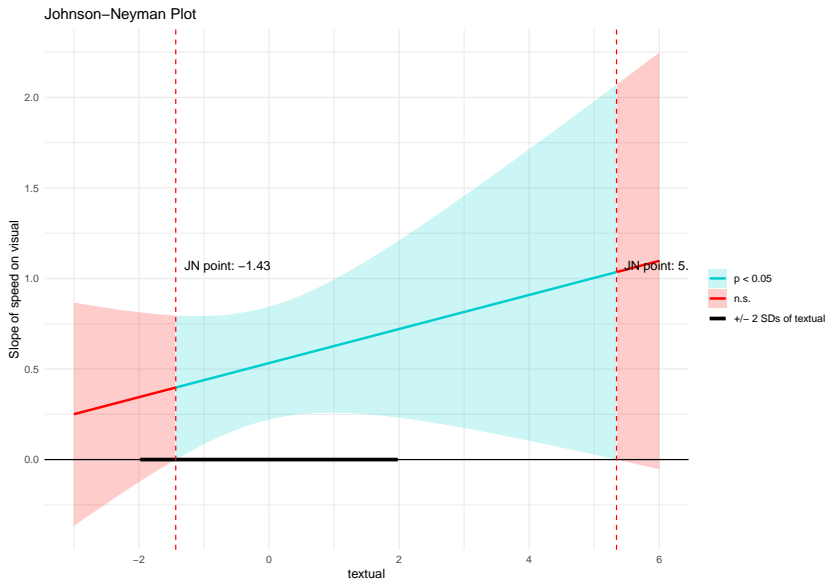
Plotting Margins (Simple Slopes)

```
plot_interaction(x = "speed", z = "textual", y = "visual", model = fit, vals_z
```



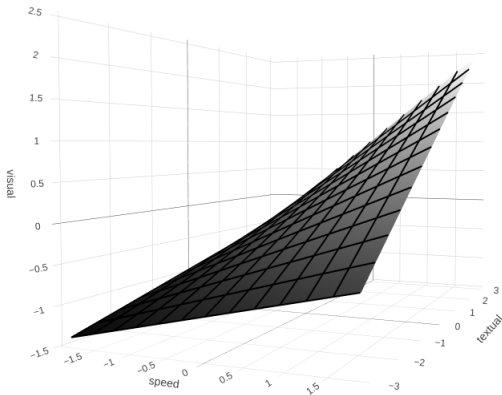
Johnson-Neyman Plot

```
plot_jn(x = "speed", z = "textual", y = "visual", model = fit, max_z = 6)
```



Surface Plot

```
plot_surface(x = "speed", z = "textual", y = "visual", model = fit,  
             colorscale = "Greys", grid = TRUE, grid_color = "black")
```



Visualizing Models With Quadratic Effects

Example Model

```
model <- '  
  X =~ x1 + x2 + x3  
  Z =~ z1 + z2 + z3  
  Y =~ y1 + y2 + y3  
  
  Y ~ X + Z + X:X + Z:Z + X:Z  
,  
  
fit <- modsem(model, data = data.quadratic, method = "qml")  
summary(fit)
```

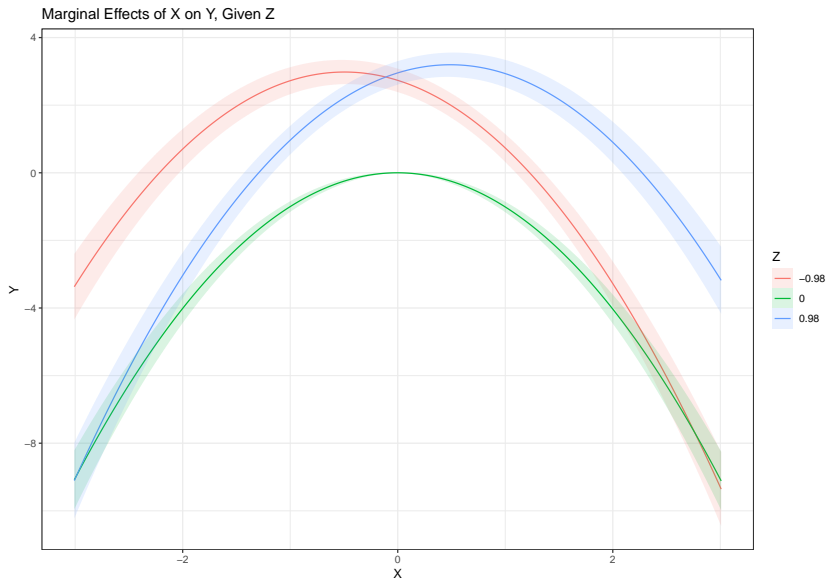
Truncated Output:

Regressions:

	Estimate	Std.Error	z.value	P(> z)
Y ~				
X	-0.005	0.067	-0.081	0.935
Z	0.113	0.163	0.693	0.488
X:X	-1.006	0.044	-23.040	0.000
X:Z	1.011	0.080	12.571	0.000
Z:Z	2.969	0.079	37.362	0.000

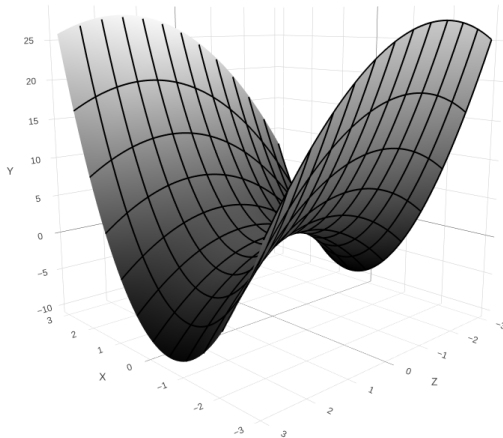
2D Plot

```
plot_interaction(x = "X", z = "Z", y = "Y", model = fit, vals_z = c(-1, 0, 1))
```



3D (Response Surface) Plot

```
plot_surface(x = "X", z = "Z", y = "Y", model = fit,  
             colorscale = "Greys", grid = TRUE, grid_color = "black")
```



References

- Algina, James, and Bradley C. Moulder. 2001. "A Note on Estimating the Jöreskog-Yang Model for Latent Variable Interaction Using LISREL 8.3." *Structural Equation Modeling* 8 (1): 40–52. https://doi.org/10.1207/S15328007SEM0801_3.
- Jöreskog, Karl Gustav, and Fan Yan. 1996. "Nonlinear Structural Equation Models: The Kenny-Judd Model With Interaction Effects."
- Kenny, David A., and Charles M. Judd. 1984. "Estimating the Nonlinear and Interactive Effects of Latent Variables." *Psychological Bulletin* 96 (1): 201–10. <https://doi.org/10.1037/0033-2909.96.1.201>.
- Klein, Andreas G, and Bengt O Muthén. 2007. "Quasi-Maximum Likelihood Estimation of Structural Equation Models with Multiple Interaction and Quadratic Effects." *Multivariate Behavioral Research* 42 (4): 647–73.
- Klein, Andreas, and Helfried Moosbrugger. 2000. "Maximum