A (Unified) Syntax for Structural Equation Modeling

Manuel J. A. Eugster and Armin Monecke

Institut für Statistik
Ludwig-Maximiliams-Universität München

Psychoco 2012, Universität Innsbruck, 2012

A (Unified) Syntax for Structural Equation Modeling

Manuel J. A. Eugster and Armin Monecke

Institut für Statistik
Ludwig-Maximiliams-Universität München

Psychoco 2012, Universität Inn

- Extensible domain specific language for the specification of structural equation models based on R formula objects.

- Decoupling of the model specification (equal for all packages) from the model representation (partly similar for all packages) and model fitting (specific for each package).

- Using "computing on the language" to satisfy statistical theory, i.e., the confirmatory character of structural equation models.

The 'lavaan model syntax'

- at the heart of the **lavaan** package is the 'model syntax': a formula-based description of the model to be estimated
- a distinction is made between four different formula types: 1) regression formulas, 2) latent variable definitions, 3) (co)variances, and 4) intercepts

1. regression formulas

- in the R environment, a regression formula has the following form:

```
y ~ x1 + x2 + x3 + x4
```

- in **lavaan**, a typical model is simply a set (or system) of regression formulas, where some variables (starting with an 'f' below) may be latent.
- for example:

```
y1 + y2 ~ f1 + f2 + x1 + x2
     f1 ~ f2 + f3
     f2 ~ f3 + x1 + x2
```


(*) See "*lavaan: an R package for structural equation modeling and more*" by Yves Rosseel, Psychoco 2011.

The 'lavaan model syntax'

- at the heart of the **lavaan** package is the 'model syntax': a formula-based description of the model to be estimated
- a distinction is made between four different formula types: 1) regression formulas, 2) latent variable definitions, 3) (co)variances, and 4) intercepts

1. regression formulas

- in the R environment, a regressi

```
y ~ x1 + x2 + x3 + x4
```

5) Constraints
6) Groups
7) Dataset

- in **lavaan**, a typical model is simply a set (or system) of regression formulas, where some variables (starting with an 'f' below) may be latent.
- for example:

```
y1 + y2 ~ f1 + f2 + x1 + x2
     f1 ~ f2 + f3
     f2 ~ f3 + x1 + x2
```


(*) See "*lavaan: an R package for structural equation modeling and more*" by Yves Rosseel, Psychoco 2011.

```
## Model formulas:
y ~ f1 + x1 + x2
```

```
## Structural models:
regression(y ~ f1 + x1 + x2)
```

```
## Structural models:
regression(y ~ f1 + x1 + x2)

Structural equation model specification
        type lhs rhs lhsparam rhsparam group
1 regression   y  f1         y       f1  <NA>
2 regression   y  x1         y       x1  <NA>
3 regression   y  x2         y       x2  <NA>

No dataset and 0 constraint(s) specified
```

```
## Structural models:
regression(y ~ f1 + x1 + x2) +
## Measurement models:
latent(f1 ~ y1 + y2 + y3)
```

```
## Structural models:
regression(y ~ f1 + x1 + x2) +
## Measurement models:
latent(f1 ~ y1 + y2 + y3)
```

```
Structural equation model specification
        type lhs rhs lhsparam rhsparam group
1 regression   y  f1         y       f1  <NA>
2 regression   y  x1         y       x1  <NA>
3 regression   y  x2         y       x2  <NA>
4     latent  f1  y1        f1       y1  <NA>
5     latent  f1  y2        f1       y2  <NA>
6     latent  f1  y3        f1       y3  <NA>

No dataset and 0 constraint(s) specified
```

```
## Structural models:
regression(y ~ f1 + x1 + x2) +
## Measurement models:
latent(f1 ~ y1 + y2 + y3) +
## Covariances and intercepts:
covariance(y1 ~ y2) + intercept(y1 ~ 1)
```

```
## Structural models:
regression(y ~ f1 + x1 + x2) +
## Measurement models:
latent(f1 ~ y1 + y2 + y3) +
## Covariances and intercepts:
covariance(y1 ~ y2) + intercept(y1 ~ 1)
```

```
Structural equation model specification
        type lhs rhs lhsparam rhsparam group
1 regression   y  f1        y       f1  <NA>
2 regression   y  x1        y       x1  <NA>
3 regression   y  x2        y       x2  <NA>
4     latent  f1  y1       f1       y1  <NA>
5     latent  f1  y2       f1       y2  <NA>
6     latent  f1  y3       f1       y3  <NA>
7 covariance  y1  y2       y1       y2  <NA>
8  intercept  y1   1       y1        1  <NA>

No dataset and 0 constraint(s) specified
```

The power of R model formulas!

```
## Interactions:
regression(y ~ f1 + x1*x2)

Structural equation model specification
        type lhs   rhs lhsparam rhsparam group
1 regression   y    f1        y       f1  <NA>
2 regression   y    x1        y       x1  <NA>
3 regression   y    x2        y       x2  <NA>
4 regression   y x1:x2        y    x1:x2  <NA>

No dataset and 0 constraint(s) specified
```

```
## Arithmetic expressions:
regression(y ~ f1 + x1 + I(3.1415 * x2))

Structural equation model specification
        type lhs             rhs lhsparam       rhsparam group
1 regression   y              f1        y             f1  <NA>
2 regression   y              x1        y             x1  <NA>
3 regression   y I(3.1415 * x2)        y I(3.1415 * x2)  <NA>

No dataset and 0 constraint(s) specified
```

```
## Arithmetic expressions:
regression(y ~ f1 + x1 + I(3.1415 * x2))

Structural equation model specification
        type lhs             rhs lhsparam       rhsparam group
1 regression   y              f1        y             f1  <NA>
2 regression   y              x1        y             x1  <NA>
3 regression   y I(3.1415 * x2)        y I(3.1415 * x2)  <NA>

No dataset and 0 constraint(s) specified


## Parameter labels:
regression(y ~ f1 + x1 + I(3.1415 * x2),
           param = c("I(3.1415 * x2)" = "pix2"))

Structural equation model specification
        type lhs             rhs lhsparam rhsparam group
1 regression   y              f1        y       f1  <NA>
2 regression   y              x1        y       x1  <NA>
3 regression   y I(3.1415 * x2)        y     pix2  <NA>

No dataset and 0 constraint(s) specified
```

```
## Groups:
regression(y ~ f1 + x1) + latent(f1 ~ y1 + y2 | g1)

Structural equation model specification
        type lhs rhs lhsparam rhsparam group
1 regression   y  f1         y       f1  <NA>
2 regression   y  x1         y       x1  <NA>
3     latent  f1  y1        f1       y1    g1
4     latent  f1  y2        f1       y2    g1

No dataset and 0 constraint(s) specified
```

```
## Groups:
regression(y ~ f1 + x1) + latent(f1 ~ y1 + y2 | g1)

Structural equation model specification
        type lhs rhs lhsparam rhsparam group
1 regression   y  f1         y       f1  <NA>
2 regression   y  x1         y       x1  <NA>
3     latent  f1  y1        f1       y1    g1
4     latent  f1  y2        f1       y2    g1

No dataset and 0 constraint(s) specified

## Global group:
regression(y ~ f1 + x1) + latent(f1 ~ y1 + y2 | g1) + group(g2)

Structural equation model specification
        type lhs rhs lhsparam rhsparam group
1 regression   y  f1         y       f1    g2
2 regression   y  x1         y       x1    g2
3     latent  f1  y1        f1       y1    g1
4     latent  f1  y2        f1       y2    g1

No dataset and 0 constraint(s) specified
```

Data for models.

```
## Model specification:
regression(y ~ f1 + x1) +
latent(f1 ~ y1 + y2)

Structural equation model specification
        type lhs rhs lhsparam rhsparam group
1 regression   y  f1         y       f1  <NA>
2 regression   y  x1         y       x1  <NA>
3     latent  f1  y1        f1       y1  <NA>
4     latent  f1  y2        f1       y2  <NA>

No dataset and 0 constraint(s) specified
```

```
## Model specification:
regression(y ~ f1 + x1) +
latent(f1 ~ y1 + y2) +
## Dataset:
dataset(dat)
```

```
Structural equation model specification
        type lhs rhs lhsparam rhsparam group level param free
1 regression   y  f1         y       f1  <NA>  <NA>  y_f1 TRUE
2 regression   y  x1         y       x1  <NA>  <NA>  y_x1 TRUE
3     latent  f1  y1        f1       y1  <NA>  <NA> f1_y1 TRUE
4     latent  f1  y2        f1       y2  <NA>  <NA> f1_y2 TRUE

A dataset and 0 constraint(s) specified
```

```
## Model specification:
regression(y ~ f1 + x1 | g1) +
latent(f1 ~ y1 + y2) +
## Dataset:
dataset(dat)
```

```
Structural equation model specification
        type lhs rhs lhsparam rhsparam group level  param free
1 regression   y  f1        y       f1    g1     1 y_f1:1 TRUE
2 regression   y  f1        y       f1    g1     2 y_f1:2 TRUE
3 regression   y  x1        y       x1    g1     1 y_x1:1 TRUE
4 regression   y  x1        y       x1    g1     2 y_x1:2 TRUE
5     latent  f1  y1       f1       y1  <NA>  <NA>  f1_y1 TRUE
6     latent  f1  y2       f1       y2  <NA>  <NA>  f1_y2 TRUE

A dataset and 0 constraint(s) specified
```

```
## Model specification:
regression(y ~ f1 + x1 | g1) +
latent(f1 ~ y1 + y2) +
## Dataset:
dataset(dat) +
## Constraints:
constraint(f1_y1 == 10)

Structural equation model specification
        type lhs rhs lhsparam rhsparam group level  param  free
1 regression   y  f1         y       f1    g1     1 y_f1:1  TRUE
2 regression   y  f1         y       f1    g1     2 y_f1:2  TRUE
3 regression   y  x1         y       x1    g1     1 y_x1:1  TRUE
4 regression   y  x1         y       x1    g1     2 y_x1:2  TRUE
5     latent  f1  y1        f1       y1  <NA>  <NA>  f1_y1 FALSE
6     latent  f1  y2        f1       y2  <NA>  <NA>  f1_y2  TRUE

A dataset and 1 constraint(s) specified
```

```
## Model specification:
regression(y ~ f1 + x1 | g1) +
latent(f1 ~ y1 + y2) +
## Dataset:
dataset(dat) +
## Constraints:
constraint(f1_y1 == 10) +
constraint(y_f1:2 == y_f1:1)

Structural equation model specification
        type lhs rhs lhsparam rhsparam group level  param  free
1 regression   y  f1        y       f1    g1     1 y_f1:1  TRUE
2 regression   y  f1        y       f1    g1     2 y_f1:2 FALSE
3 regression   y  x1        y       x1    g1     1 y_x1:1  TRUE
4 regression   y  x1        y       x1    g1     2 y_x1:2  TRUE
5     latent  f1  y1       f1       y1  <NA>  <NA>  f1_y1 FALSE
6     latent  f1  y2       f1       y2  <NA>  <NA>  f1_y2  TRUE

A dataset and 2 constraint(s) specified
```

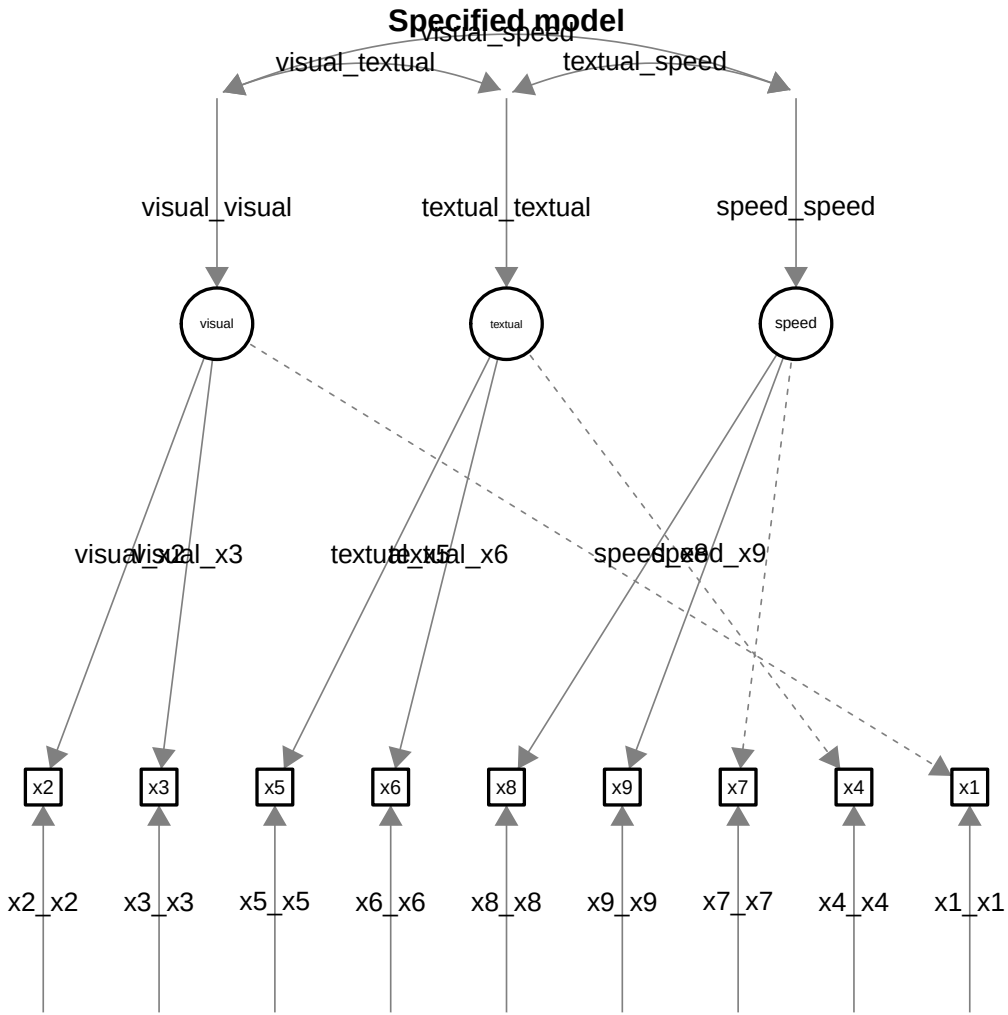
Model checking.

```
## Measurement model
m <- latent(visual ~ x1 + x2 + x3) +
     latent(textual ~ x4 + x5 + x6) +
     latent(speed ~ x7 + x8 + x9)
m <- m + dataset(HolzingerSwineford1939)
## MV variances:
m <- m + covariance(x1 ~ x1) + covariance(x2 ~ x2) +
         covariance(x3 ~ x3) + covariance(x4 ~ x4) +
         covariance(x5 ~ x5) + covariance(x6 ~ x6) +
         covariance(x7 ~ x7) + covariance(x8 ~ x8) +
         covariance(x9 ~ x9)
## LV variances:
m <- m + covariance(visual ~ visual) +
         covariance(textual ~ textual) +
         covariance(speed ~ speed)
## LV covariance:
m <- m + covariance(visual ~ textual) +
         covariance(visual ~ speed) +
         covariance(textual ~ speed)
## Constraints:
m <- m + constraint(visual_x1 == 1) +
         constraint(textual_x4 == 1) +
         constraint(speed_x7 == 1)
```

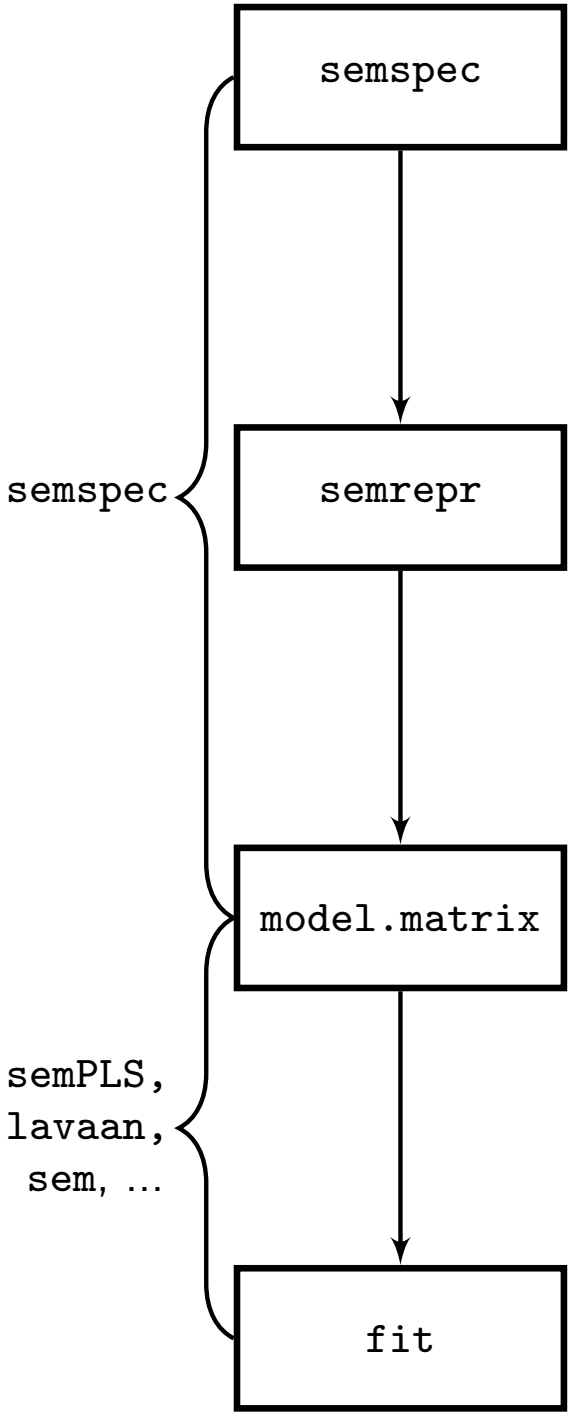
```
## Model specification summary:
summary(m)
Structural equation model specification

  latent(formula = visual ~ x1 + x2 + x3)
  latent(formula = textual ~ x4 + x5 + x6)
  latent(formula = speed ~ x7 + x8 + x9)
...

Variables:     Latent   Manifest
        12          3          9
  Latent:
      visual, textual, speed
  Manifest:
      x1, x2, x3, x4, x5, x6, x7, x8, x9

Parameters:        Free       Fixed  Restricted
         24          21           3           0
  Free:
      visual_x2, visual_x3, textual_x5, textual_x6, speed_x8,
      speed_x9, x1_x1, x2_x2, x3_x3, x4_x4, x5_x5, x6_x6, x7_x7,
      x8_x8, x9_x9, visual_visual, textual_textual, speed_speed,
      visual_textual, visual_speed, textual_speed
```

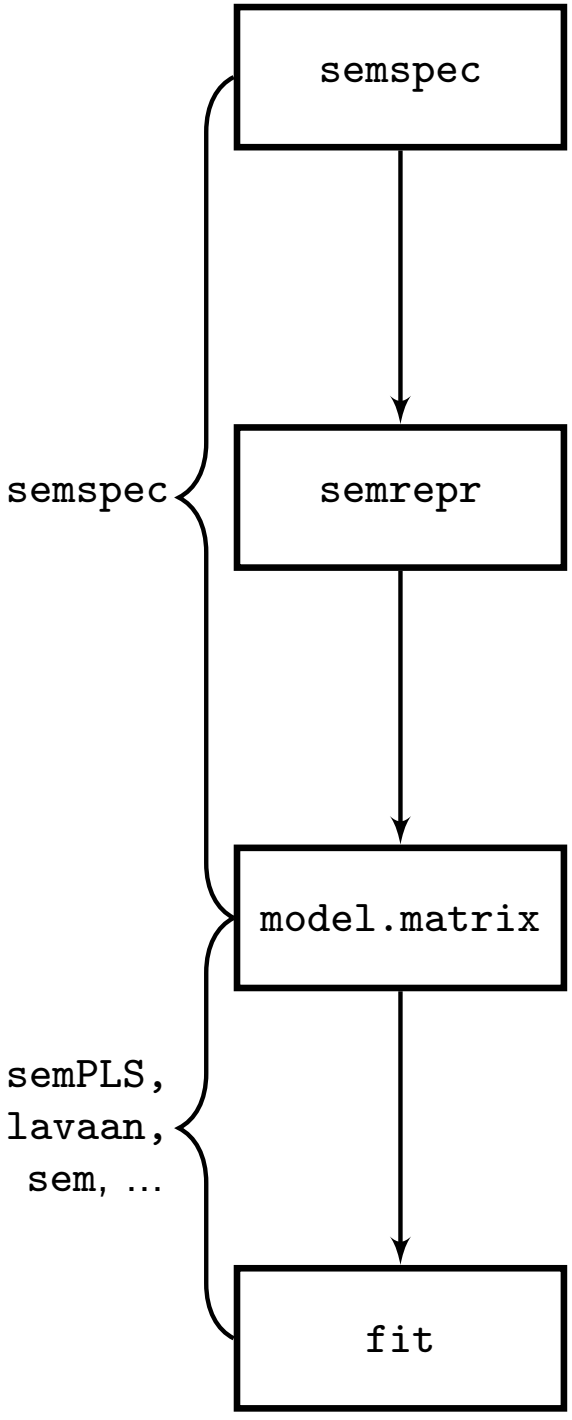
```
...
  Fixed:
      visual_x1, textual_x4, speed_x7
  Restricted:


Constraints:       Active     Inactive
           3            3            0
  Active:
     visual_x1 == 1
     textual_x4 == 1
     speed_x7 == 1
  Inactive:


Data: 301 obs. of 9 variables, 0 grouping variables
  Variable Level Group Mean Median  SD Kurtosis Skewness   N NAs
        x1    NA    NA  4.9    5.0 1.2     0.31    -0.25 301   0
        x2    NA    NA  6.1    6.0 1.2     0.33     0.47 301   0
...

Degrees of freedom: 24
```

```
## Model specification plot (via qgraph):
plot(m)
```



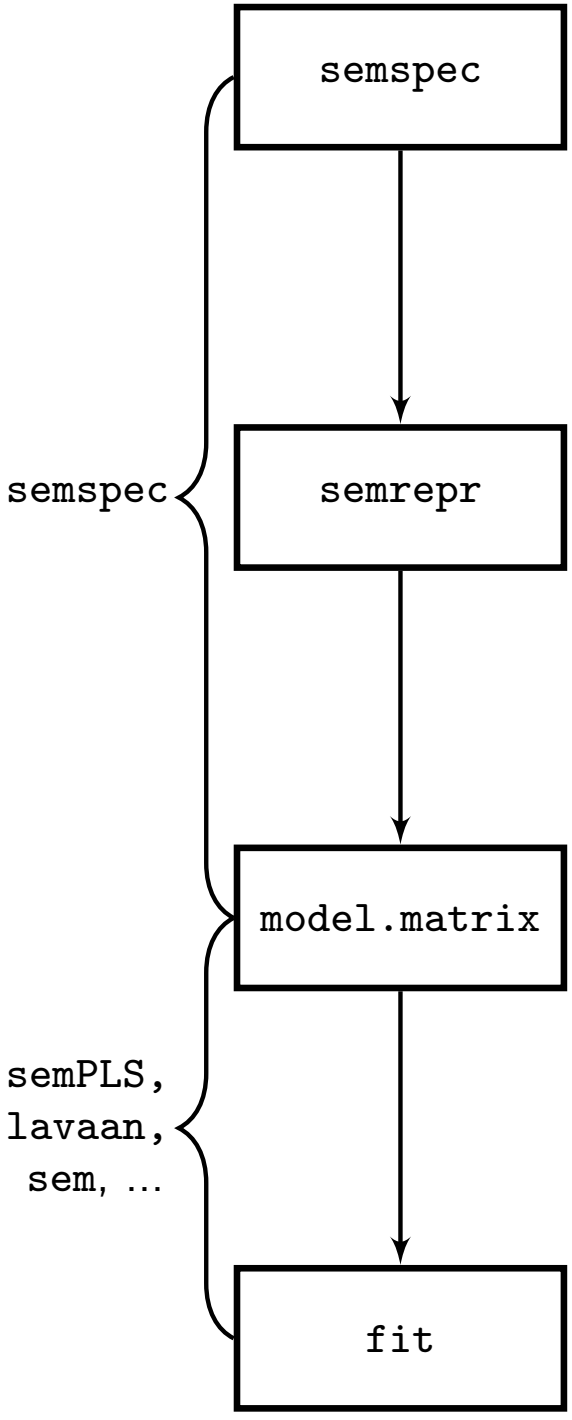
Model fitting: our initial design idea ...


semspec
semrepr
model.matrix
fit
semspec
semPLS,
lavaan,
sem, ...



Formula representation

```
regression(y ~ f1 + x1 + x2) +
latent(f1 ~ y1 + y2 + y3) +
constraint(y1_y2 == 10) + dataset(dat)
```

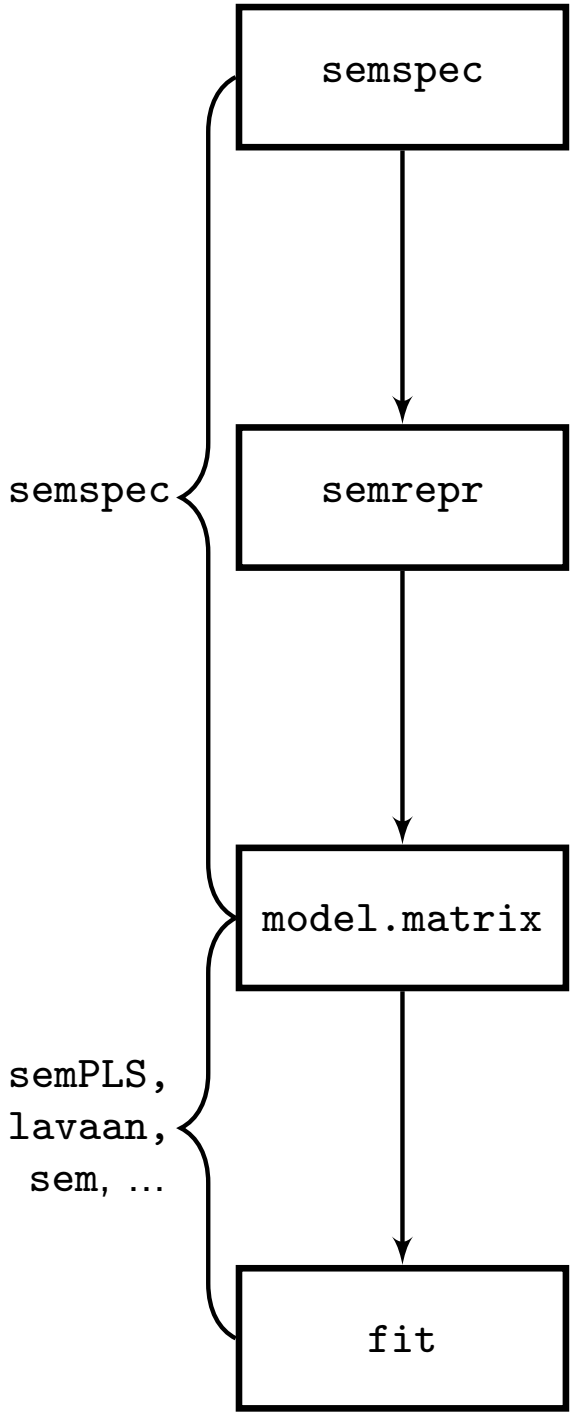


Formula representation

```
regression(y ~ f1 + x1 + x2) +
latent(f1 ~ y1 + y2 + y3) +
constraint(y1_y2 == 10) + dataset(dat)
```

List representation

```
Structural equation model specification
        type lhs rhs lhsparam rhsparam group
1 regression   y  f1         y       f1  <NA>
2 regression   y  x1         y       x1  <NA>
...
```



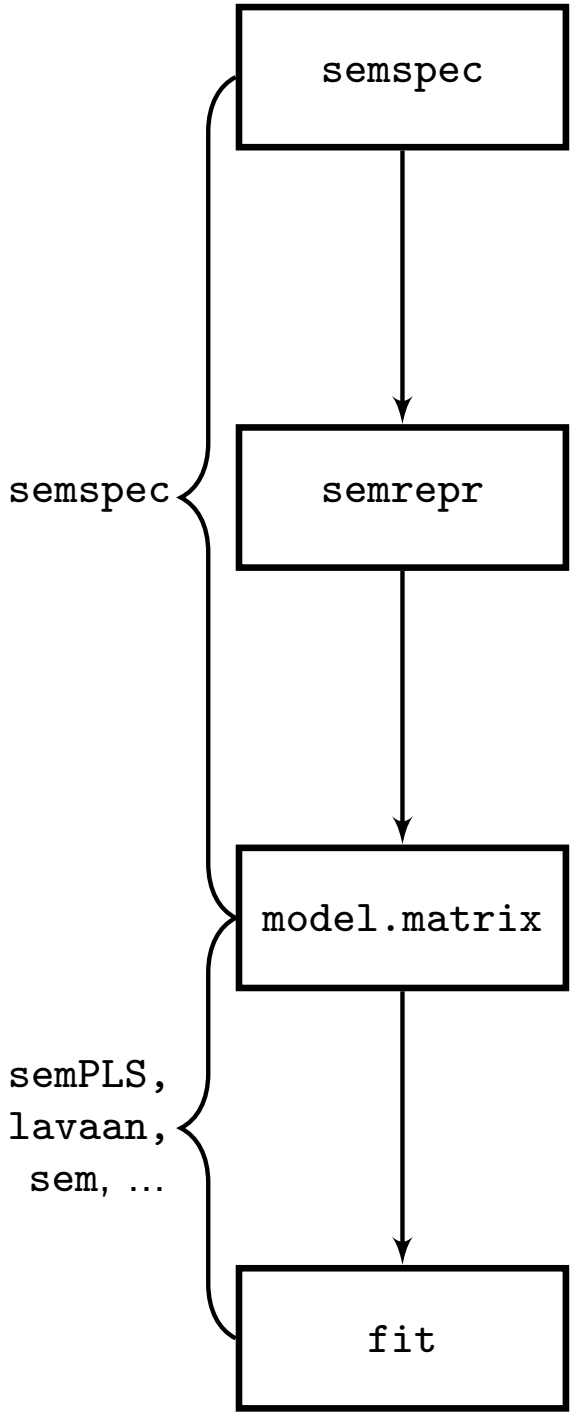
Formula representation

```
regression(y ~ f1 + x1 + x2) +
latent(f1 ~ y1 + y2 + y3) +
constraint(y1_y2 == 10) + dataset(dat)
```

List representation

```
Structural equation model specification
        type lhs rhs lhsparam rhsparam group
1 regression   y  f1         y       f1  <NA>
2 regression   y  x1         y       x1  <NA>
...
```

Matrix representation



Formula representation

```
regression(y ~ f1 + x1 + x2) +
latent(f1 ~ y1 + y2 + y3) +
constraint(y1_y2 == 10) + dataset(dat)
```

List representation

```
Structural equation model specification
        type lhs rhs lhsparam rhsparam group
1 regression   y  f1         y       f1  <NA>
2 regression   y  x1         y       x1  <NA>
...
```

Matrix representation

Fitting methods

Model translator: proof of concept ...

```
## Translation for the sem package:
as_sem_syntax(m)
```

```
x2 = visual_x2 * visual
x3 = visual_x3 * visual
x5 = textual_x5 * textual
x6 = textual_x6 * textual
x8 = speed_x8 * speed
x9 = speed_x9 * speed
x7 = 1 * speed
x4 = 1 * textual
x1 = 1 * visual
C(x1, x1) = x1_x1
C(x2, x2) = x2_x2
C(x3, x3) = x3_x3
...
```

```
## Model fit with the sem package:
semfit_sem(m)
```

```
## Translation for the sem package:
as_sem_syntax(m)
x2 = visual_x2 * visual
x3 = visual_x3 * visual
x5 = textual_x5 * textual
x6 = textual_x6 * textual
x8 = speed_x8 * speed
x9 = speed_x9 * speed
x7 = 1 * speed
x4 = 1 * textual
x1 = 1 * visual
C(x1, x1) = x1_x1
C(x2, x2) = x2_x2
C(x3, x3) = x3_x3
...

## Model fit with the sem package:
semfit_sem(m)

## ... semPLS and lavaan packages:
as_semPLS_syntax(m); semfit_semPLS(m)
as_lavaan_syntax(m); semfit_lavaan(m)
```

A Unified Syntax for SEM?

One Syntax to rule them all!

Adding semantics to the formulas using descriptive functions and seeing model specifications as programs allows

- to create easy and easily extensible model specification "user-interfaces" with on-the-fly error checking;
- to maintain a clean separation of model specification, model representation and model fitting;
- and to satisfy statistical theory.

Prototype implementation available as package `semspec` from `https://r-forge.r-project.org/projects/sempls/`.