2021 ABCD Workshop: SEM Practical

Ethan McCormick and John Flournoy

2021-07-13

Contents

Intro

Loading Relevant Packages

```
packages =  c('lavaan','nlme','ggplot2','patchwork',
              'kableExtra', 'psych', 'dplyr', 'tidyr',
              'semPlot')
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
```

```r
  install.packages(setdiff(packages, rownames(installed.packages())),
                   repos = "http://cran.us.r-project.org")
}
invisible(lapply(packages, library, character.only = TRUE))
```

Read In Data

```r
ABCD = read.csv('abcd_sem.csv')
str(ABCD)
```

```
## 'data.frame':    964 obs. of  13 variables:
##  $ id    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ female: int  1 1 0 0 0 0 0 0 1 1 ...
##  $ advers: int  0 0 0 1 0 1 0 1 0 1 ...
##  $ VS.1  : num  6.67 5.24 6.19 6.19 7.62 ...
##  $ VS.2  : num  NA 4.29 9.05 4.76 5.71 ...
##  $ VS.3  : num  NA NA 8.57 4.76 7.14 ...
##  $ VS.4  : num  NA NA 7.14 6.19 8.57 ...
##  $ VS.5  : num  NA NA 8.82 5.31 7.08 ...
##  $ EXT.1 : num  1.08 3.25 2.17 NA 3.25 ...
##  $ EXT.2 : num  0 2.17 2.17 2.53 2.53 ...
##  $ EXT.3 : num  NA 1.083 2.528 0.722 3.25 ...
##  $ EXT.4 : num  NA NA 2.89 1.08 3.25 ...
##  $ EXT.5 : num  NA NA 4.494 0.809 1.606 ...
```

These data have been synthesized from other data and then addition simulations were performed to make the data behave. The original data labels were changed for use in the workshop. The relevant variables include:

`id`: unique identifier
`female`: self-identified sex (binary: 0 = male, 1 = female)
`advers`: did the individual experience early-childhood adversity (binary: 0 = no, 1 = yes)
`VS`: measures of ventral striatum response during reward anticipation
`EXT`: measures of parent-reported externalizing behavior

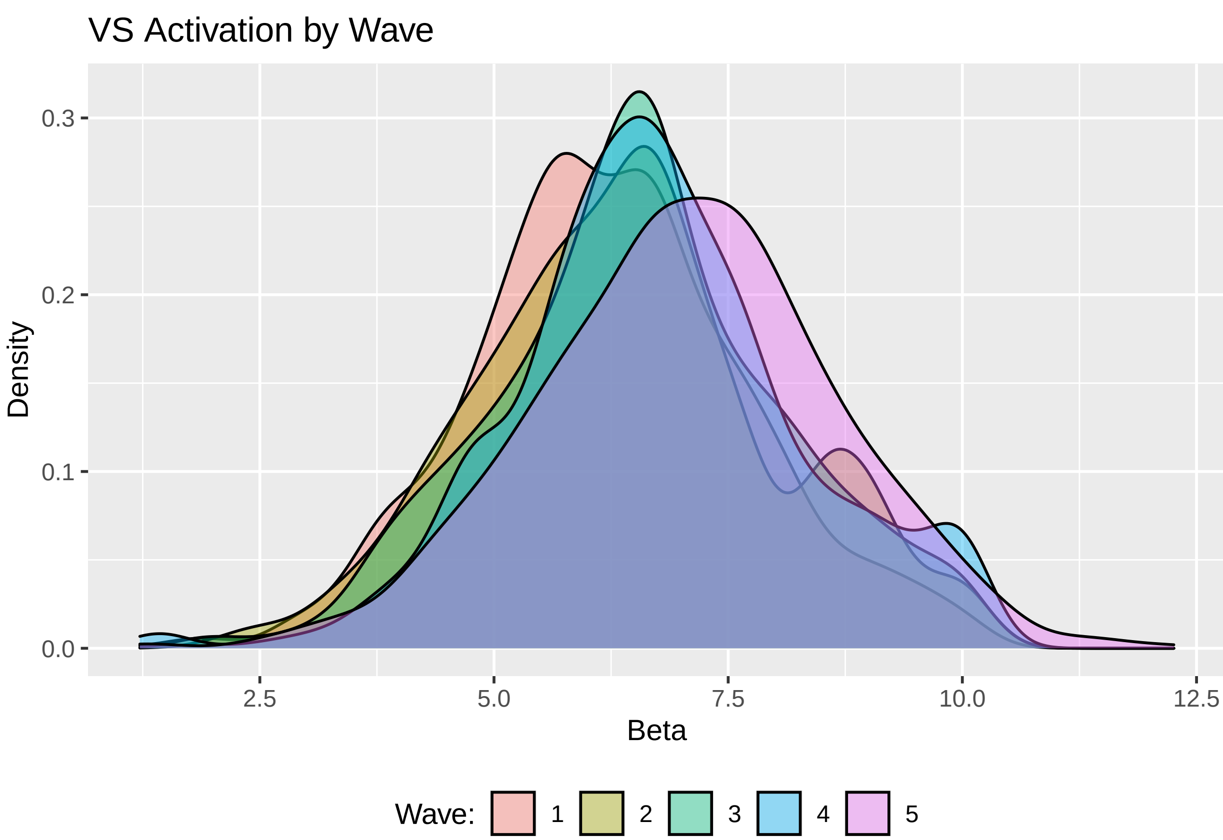
Descriptives

```r
describe(ABCD[,2:ncol(ABCD)], fast = TRUE) %>%
  kbl() %>%
  kable_styling(full_width = F)
```

```r
#knitr::kable(describe(ABCD[,2:ncol(ABCD)], fast = TRUE))

# Density by Wave: VS Activation
ggplot(ABCD %>%
         pivot_longer(cols=starts_with('VS'),
                      names_to='wave',
                      values_to='VS'),
       aes(x=VS, group=wave, fill=wave)) +
  geom_density(alpha=.4) +
  labs(title = 'VS Activation by Wave',
       x='Beta',
       y = 'Density',
       fill='Wave:') +
  scale_fill_discrete(labels = c('1','2','3','4','5')) +
```

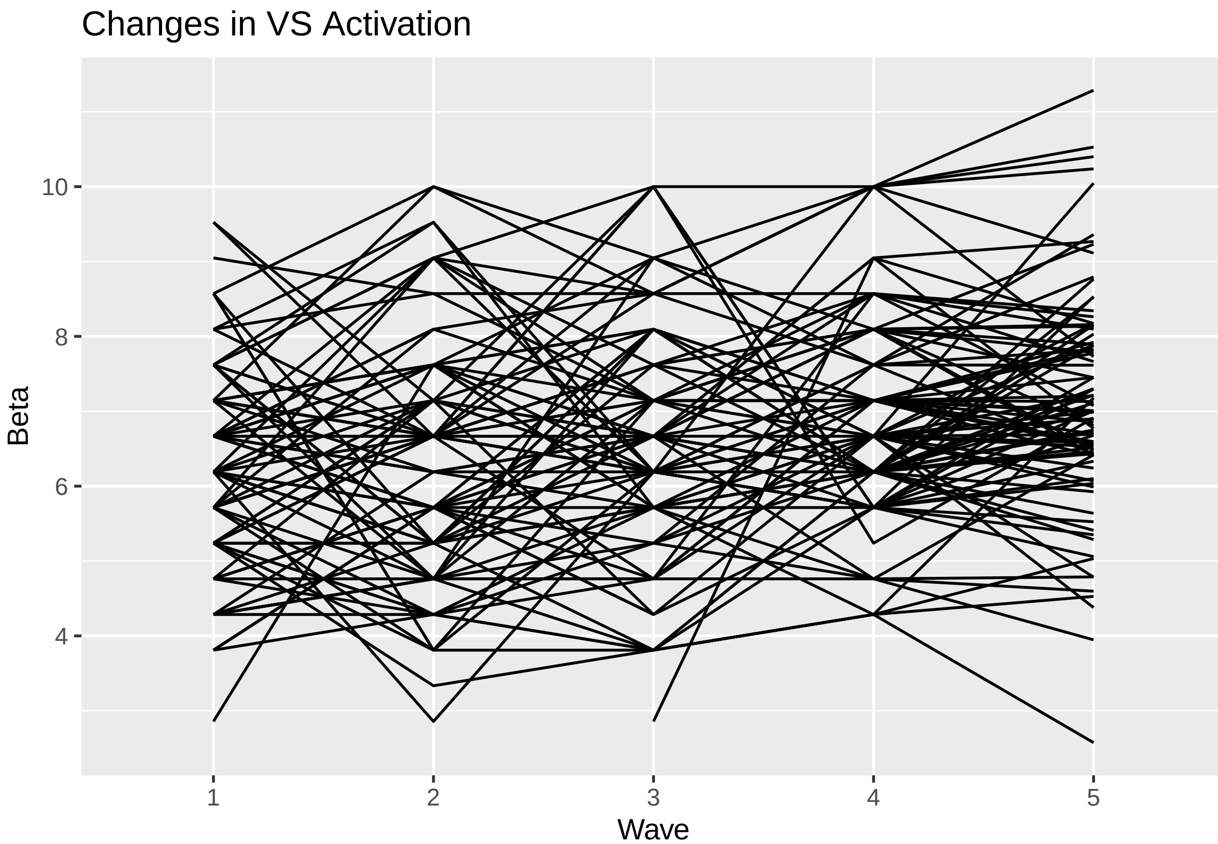
	vars	n	mean	sd	min	max	range	se
female	1	964	0.4460581	0.4973398	0.000000	1.000000	1.000000	0.0160182
advers	2	964	0.5477178	0.4979762	0.000000	1.000000	1.000000	0.0160387
VS.1	3	964	6.2393796	1.4674234	1.428571	10.000000	8.571429	0.0472625
VS.2	4	917	6.4355819	1.5863089	1.904762	10.000000	8.095238	0.0523845
VS.3	5	912	6.5544069	1.5376050	1.904762	10.000000	8.095238	0.0509152
VS.4	6	880	6.7408009	1.5402031	1.428571	10.000000	8.571429	0.0519202
VS.5	7	880	7.0536888	1.5951754	1.218616	12.257785	11.039169	0.0537734
EXT.1	8	923	2.3431141	0.8614213	0.000000	3.972222	3.972222	0.0283540
EXT.2	9	908	2.4371021	0.8133013	0.000000	3.972222	3.972222	0.0269903
EXT.3	10	884	2.4415848	0.8586684	0.000000	3.972222	3.972222	0.0288801
EXT.4	11	828	2.6071522	0.8143713	0.000000	3.972222	3.972222	0.0283014
EXT.5	12	828	2.7031994	1.2521188	-1.013396	6.267124	7.280519	0.0435141

```
  theme(legend.position='bottom')
```

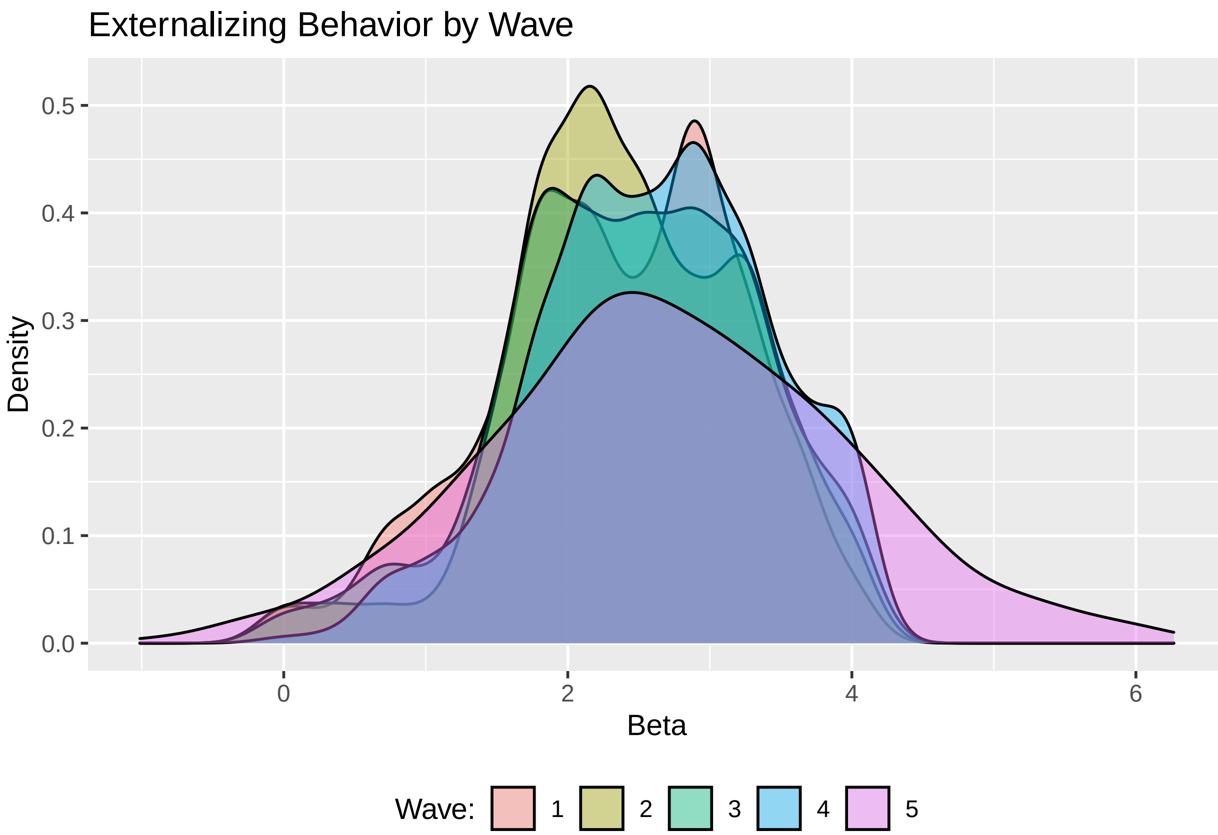



```
# VS Activation Change over Time: By ID
ggplot(ABCD %>%
         pivot_longer(cols=starts_with('VS'),
                      names_to='wave',
                      values_to='VS') %>%
         filter(id %in% sample(unique(ABCD$id), 100)),
       aes(x=wave, y=VS, group=id)) +
  geom_line() +
  labs(title = 'Changes in VS Activation',
       x='Wave',
```

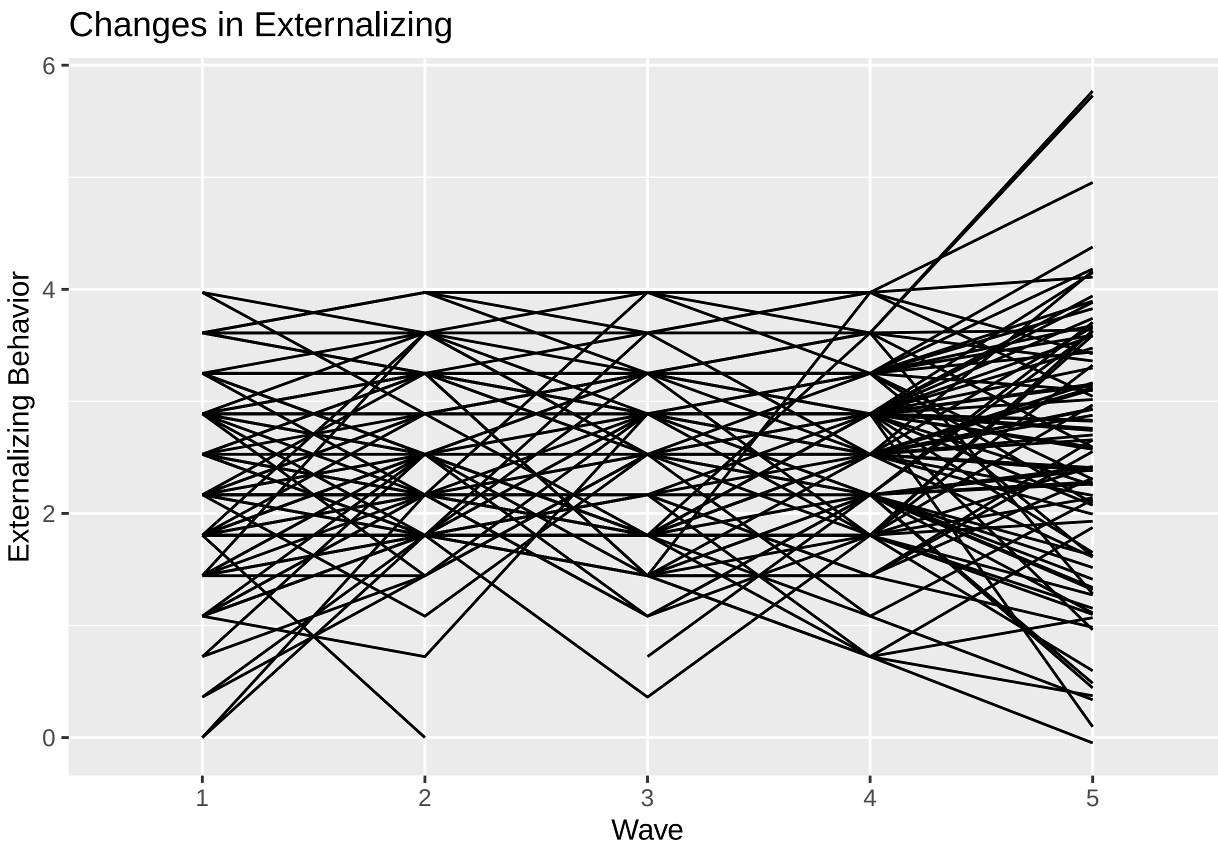
```
       y = 'Beta') +
  scale_x_discrete(labels = c('1','2','3','4','5'))
```




```
# Density by Wave: Externalizing Behavior
ggplot(ABCD %>%
         pivot_longer(cols=starts_with('EXT'),
                      names_to='wave',
                      values_to='EXT'),
       aes(x=EXT, group=wave, fill=wave)) +
  geom_density(alpha=.4) +
  labs(title = 'Externalizing Behavior by Wave',
       x='Beta',
       y = 'Density',
       fill='Wave:') +
  scale_fill_discrete(labels = c('1','2','3','4','5')) +
  theme(legend.position='bottom')
```




```
# Externalizing Behavior Change over Time: By ID
ggplot(ABCD %>%
         pivot_longer(cols=starts_with('EXT'),
                      names_to='wave',
                      values_to='EXT') %>%
         filter(id %in% sample(unique(ABCD$id), 100)),
       aes(x=wave, y=EXT, group=id)) +
  geom_line() +
  labs(title = 'Changes in Externalizing',
       x='Wave',
       y = 'Externalizing Behavior') +
  scale_x_discrete(labels = c('1','2','3','4','5'))
```



Some basics

Residualized change v difference scores

To easily compute these scores, I'll make the data long and compute the lag (t-1) value for each variable.

```
ABCD_l <- tidyr::pivot_longer(ABCD, names_to = 'key', values_to = 'value', tidyr::matches('^(VS|EXT)'))
  tidyr::extract(col = key, into = c('var', 'wave'), regex = '(\\w+)\\.(\\d+)') %>%
  tidyr::pivot_wider(names_from = 'var', values_from = 'value') %>%
  dplyr::group_by(id) %>%
  dplyr::mutate(across(c(wave, EXT, VS), lag, .names = '{.col}_lag'))
```

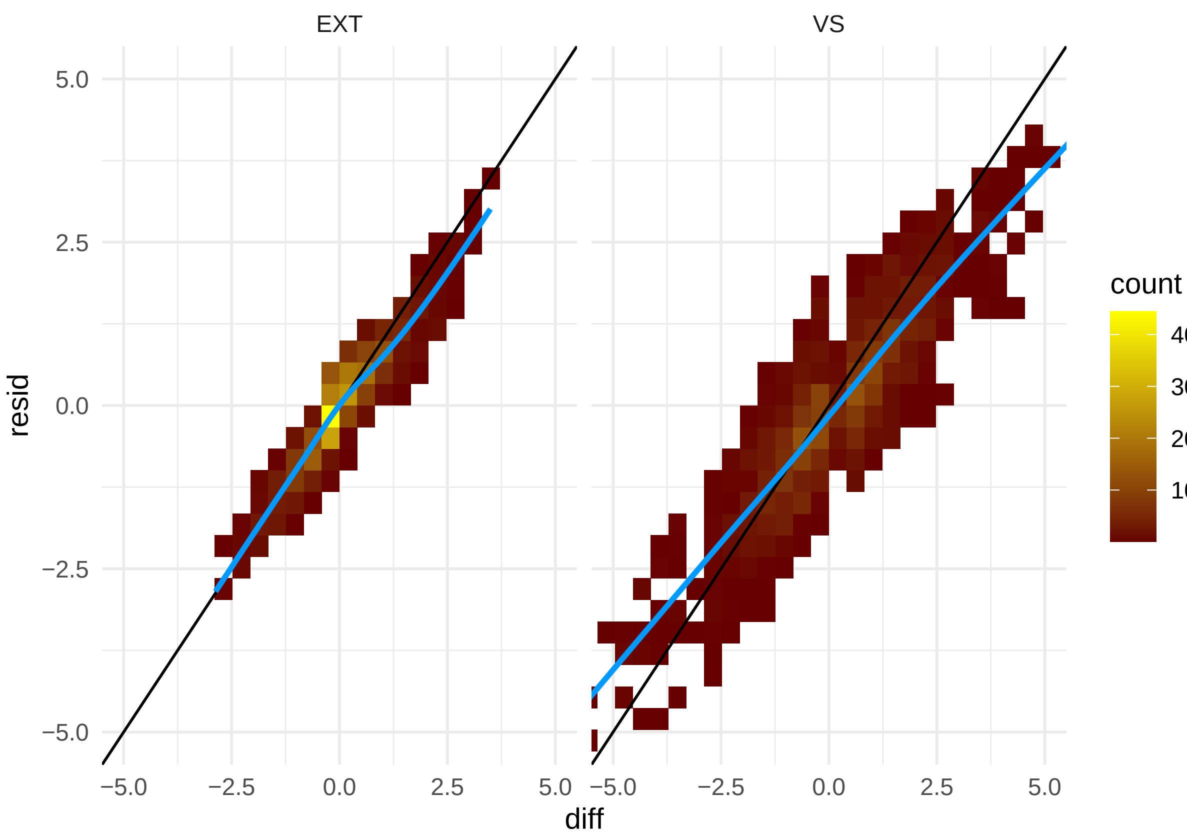
Now I'll get the residuals from a simple `lm` model, and also compute the raw difference score.

```
#residualized change
#get the residual for the regression for each wave on the one before it
ABCD_l_resid <- dplyr::group_by(filter(ABCD_l, !is.na(wave_lag)), wave, wave_lag) %>%
  mutate(VS_resid = resid(lm(VS ~ 1 + VS_lag, na.action = 'na.exclude')),
         EXT_resid = resid(lm(EXT ~ 1 + EXT_lag, na.action = 'na.exclude')))

#difference scores
#just take the difference between the measure at wave T and wave T-1
ABCD_l_resid_diff <- dplyr::group_by(ABCD_l_resid, wave, wave_lag, id) %>%
  mutate(VS_diff = VS - VS_lag,
         EXT_diff = EXT - EXT_lag)
```

What's the correlation of these?

```r
select(ungroup(ABCD_l_resid_diff), id, wave, matches('(VS|EXT)_(resid|diff)')) %>%
  pivot_longer(cols = c(-id, -wave)) %>%
  extract(name, c('var', 'stat'), '(\\w+)_(\\w+)') %>%
  pivot_wider(names_from = 'stat', values_from = 'value') %>%
  ggplot(aes(x = diff, y = resid)) +
  # geom_point(size = .5, alpha = .05) +
  geom_bin2d() +
  geom_abline(intercept = 0, slope = 1) +
  geom_line(stat = 'smooth', color = '#0099ff', size = 1) +
  facet_grid(~var) +
  scale_fill_gradient(low = '#660000', high = '#FFFF00') +
  theme_minimal() +
  coord_cartesian(x = c(-5, 5), y = c(-5, 5))
```

```
## Warning: Removed 843 rows containing non-finite values (stat_bin2d).
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## Warning: Removed 843 rows containing non-finite values (stat_smooth).
```



What does it seem like is going on here?

Think about how the variance differs for each type of estimate.

Equations, path diagrams, and … SYNTAX

Equations are the real deal (as real as it gets at this level of abstraction). Path diagrams and syntax are both ways of conveying the equations.

Let's start with the first diagram we saw:

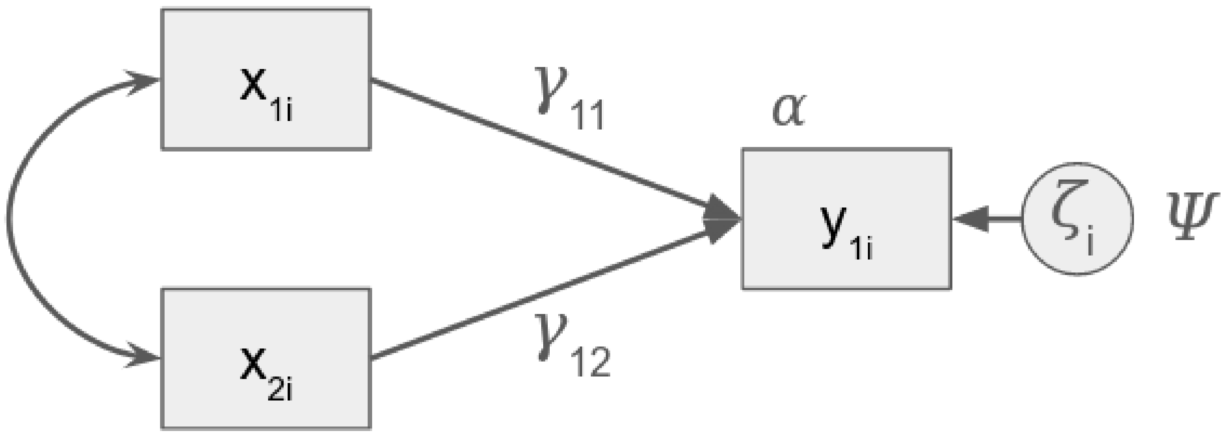


Figure 1: A simple SEM diagram with two X variables pointing at a Y variable

The `lavaan` package in R has *syntax* for all of the relationships here (plus more). If you recall, these are almost all just regressions. If you're already familiar with regression syntax in R, this will look familiar to you. One big difference from other common modeling packages is that the model is expressed as a string of text. We also need to learn the opperator for "covariance" to get the path between x_{1i} and x_{2i}. We also need an operator to specify the residual variance. Yay: the variance and covariance operators are the same. The intercept also gets it's own notation (it's not part of the regression for reasons).

- Regression: `VAR1 ~ VAR2 [+ VAR3 ...]`
- (Co)variance: `VAR1 ~~ VAR2 [+ VAR3 ...]`
- Mean/Intercept: `VAR1 ~ 1`

Note: Let's acknowledge that there is some weird stuff going on here. In this diagram y is regressed on the residual, but we only care about the residual's variance (the regression coefficient is actually set to 1). So you may see diagrams with arrows like this, or just with double headed arrows pointing to the same variable. To go down the rabbit hole a little further, you specify variances the same way whether they are variances of residuals (for variables that are on the DV side of a regression) *or* whether they are variances of a variable itself (if it's an IV). Moreover, sometimes they're not even written into the diagram! The diagram above assumes that both x_{1i} and x_{2i} have variances as well as covary with each other.

Also Note: Intercepts refer to means when they are part of a regression (conditional means), and means refer to means when they are unconditional. The syntax for both is the same

Also Also Note: Variance/covariance structures are often very consistent across models so `lavaan` adds many in by default. Same with intercepts/means. For now, I'm going to write out the full model, noting where lavaan usually has defaults.

```
simple_sem_model <- '
y ~ x1 + x2

#lavaan covariance defaults
#notice we do not allow covariance between
#residuals and the other variables.
x1 ~~ x1
x2 ~~ x1
x2 ~~ x2
y ~~ y
```

```
#lavaan intercept defaults
#Intercept
y ~ 1
#Means
x1 ~ 1
x2 ~ 1
'
```

We can count the number of parameters we think are implied by the model diagram and then count them in the syntax.

Let's actually fit this model using variables from the sample data.

```r
names(ABCD)
```

```
##  [1] "id"     "female" "advers" "VS.1"   "VS.2"   "VS.3"   "VS.4"   "VS.5"
##  [9] "EXT.1"  "EXT.2"  "EXT.3"  "EXT.4"  "EXT.5"
```

```r
simple_sem_model <- '
VS.2 ~ EXT.1 + EXT.2
EXT.2 ~~ EXT.1

#lavaan covariance defaults
#notice we do not allow covariance between
#residuals and the other variables.
EXT.1 ~~ EXT.1
EXT.2 ~~ EXT.1
VS.2 ~~ VS.2

#lavaan mean/intercept defaults
#Intercept
VS.2 ~ 1
#Means
EXT.1 ~ 1
EXT.2 ~ 1
'

simple_sem_fit <- lavaan::sem(simple_sem_model, data = ABCD)
```

```
## Warning in lav_partable_flat(FLAT, blocks = "group", meanstructure = meanstructure, : duplicated elem
##      EXT.1 ~~ EXT.2
```

```r
summary(simple_sem_fit)
```

```
## lavaan 0.6-7 ended normally after 26 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                          9
##
##                                                   Used       Total
##   Number of observations                           858         964
##
## Model Test User Model:
##
##   Test statistic                                 0.000
```

```
##   Degrees of freedom                                 0
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   VS.2 ~
##     EXT.1            -0.155    0.079   -1.946    0.052
##     EXT.2             0.435    0.082    5.336    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   EXT.1 ~~
##     EXT.2             0.393    0.026   14.982    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.2              5.739    0.184   31.237    0.000
##     EXT.1             2.394    0.028   85.206    0.000
##     EXT.2             2.459    0.027   89.839    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     EXT.1             0.678    0.033   20.712    0.000
##    .VS.2              2.369    0.114   20.712    0.000
##     EXT.2             0.643    0.031   20.712    0.000
```

Let's confirm what `lavaan` sets by default (see ?sem "Details" for more on the defaults).

```
more_simple_sem_model <- '
VS.2 ~ EXT.1 + EXT.2
EXT.2 ~~ EXT.1
'

more_simple_sem_fit <- lavaan::sem(more_simple_sem_model, data = ABCD)
summary(more_simple_sem_fit)
```

```
## lavaan 0.6-7 ended normally after 23 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                          6
##
##                                                   Used       Total
##   Number of observations                           858         964
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 0
##
```

```
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   VS.2 ~
##     EXT.1            -0.155    0.079   -1.946    0.052
##     EXT.2             0.435    0.082    5.336    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   EXT.1 ~~
##     EXT.2             0.393    0.026   14.982    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.2              2.369    0.114   20.712    0.000
##     EXT.1             0.678    0.033   20.712    0.000
##     EXT.2             0.643    0.031   20.712    0.000
```

Wait, what about the means?

Often, we don't even care about the means. Who interprets the intercept of a regression? The `sem` function is set by default to not estimate *any* means. But we can turn it on with `meanstructure = TRUE` in which case it will estimate all means and interecpts.

```r
more_simple_sem_fit <- lavaan::sem(more_simple_sem_model, data = ABCD, meanstructure = TRUE)
summary(more_simple_sem_fit)
```

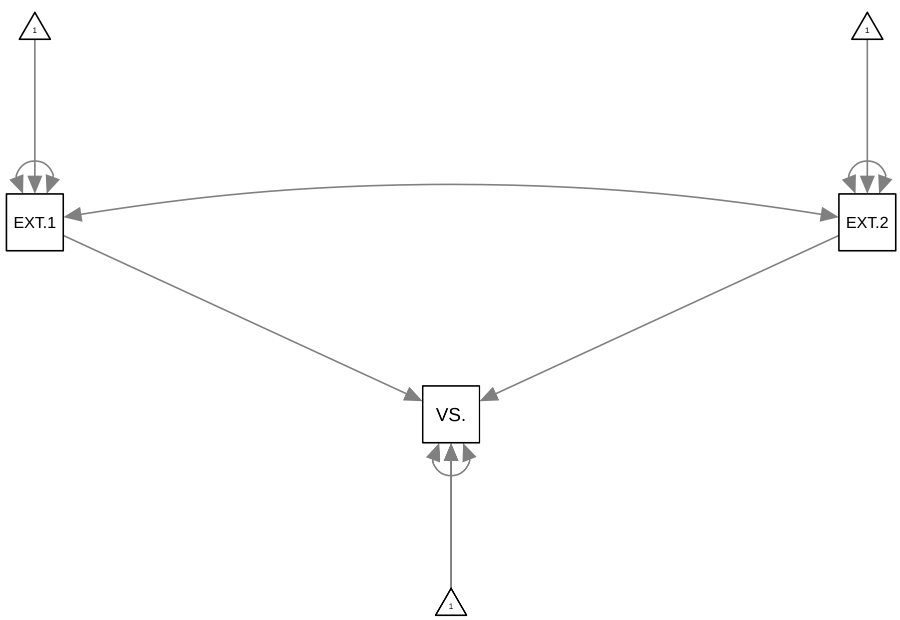
```
## lavaan 0.6-7 ended normally after 26 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                          9
##
##                                                   Used       Total
##   Number of observations                           858         964
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 0
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   VS.2 ~
##     EXT.1            -0.155    0.079   -1.946    0.052
```

```
##     EXT.2             0.435    0.082    5.336    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   EXT.1 ~~
##     EXT.2             0.393    0.026   14.982    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.2              5.739    0.184   31.237    0.000
##     EXT.1             2.394    0.028   85.206    0.000
##     EXT.2             2.459    0.027   89.839    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.2              2.369    0.114   20.712    0.000
##     EXT.1             0.678    0.033   20.712    0.000
##     EXT.2             0.643    0.031   20.712    0.000
```

This is the syntax we'll use to build up all the other models.

Oh, by the way, we can plot these too. More complex models don't work well but for simple ones we can:

```
semPlot::semPaths(more_simple_sem_fit)
```



The triangle with the `1` inside indicates that a mean or intercept is estimated (this will be important for growth models where we care about the mean of the latent slope).

Crazy take-away point: You can estimate the covariance structure separate from the mean structure!

Syntax for a slightly more complicated model

How to think through writing out these models based on diagrams? I like to start with the regressions.

We have 2 regressions here for y_{1i} and y_{2i}. Each has two arrows going into it. Let's find where they come from and write them out in lavaan syntax:

- `y1 ~ x1 + x2`

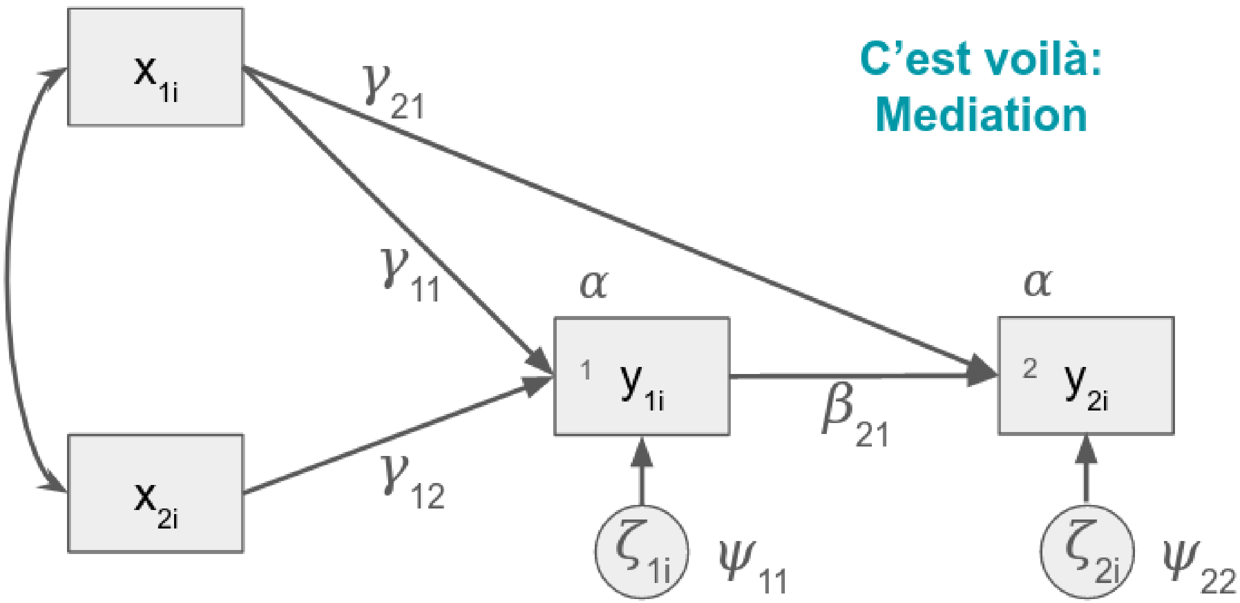


Figure 2: A more complicated SEM diagram with two X variables pointing at two Y variables

- `y2 ~ y1 + x1`

We now are left with one covariance:

- `x1 ~~ x2`

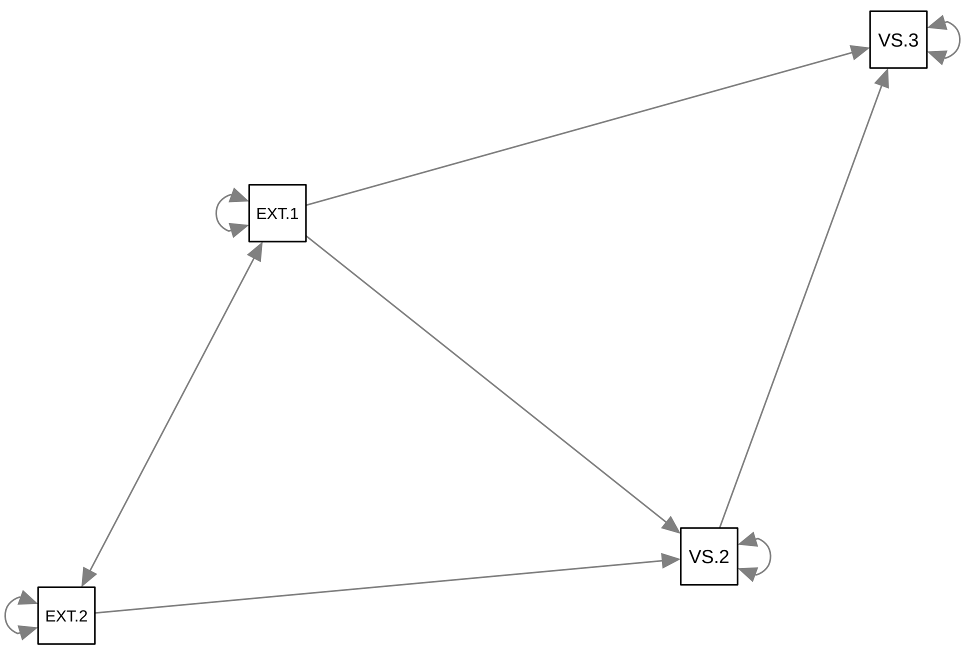
Using our variables from before (even though they might not make sense), we just add one regression equation:

```
slightly_complicated_sem_model <- '
VS.2 ~ EXT.1 + EXT.2
VS.3 ~ VS.2 + EXT.1
EXT.2 ~~ EXT.1
'

#leaving the mean structure out
slightly_complicated_sem_fit <- sem(slightly_complicated_sem_model, data = ABCD)
summary(slightly_complicated_sem_fit)
```

```
## lavaan 0.6-7 ended normally after 30 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                          9
##
##                                                   Used       Total
##   Number of observations                           820         964
##
## Model Test User Model:
##
##   Test statistic                                 0.096
##   Degrees of freedom                                 1
##   P-value (Chi-square)                           0.757
```

```
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   VS.2 ~
##     EXT.1            -0.163    0.079   -2.074    0.038
##     EXT.2             0.478    0.083    5.773    0.000
##   VS.3 ~
##     VS.2              0.513    0.029   17.505    0.000
##     EXT.1             0.080    0.054    1.470    0.141
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   EXT.1 ~~
##     EXT.2             0.389    0.027   14.679    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.2              2.246    0.111   20.248    0.000
##    .VS.3              1.647    0.081   20.248    0.000
##     EXT.1             0.685    0.034   20.248    0.000
##     EXT.2             0.621    0.031   20.248    0.000
```

```r
semPlot::semPaths(slightly_complicated_sem_fit, layout = 'spring')
```



Factors

This is where the magic starts to happen (though the above framework is awesome for doing regression too; very useful for mediation and other complex models).

Latent variables are nothing more than something invisible causing a bunch of things. In other words, it's a bunch of different y variables being regressed on something invisible.

```
some_factor_model <- '
y1 ~ ETA
y2 ~ ETA
y3 ~ ETA
y4 ~ ETA
'

some_factor_model <- '
EXT.1 ~ ETA
EXT.2 ~ ETA
EXT.3 ~ ETA
EXT.4 ~ ETA
'

some_factor_fit <- sem(some_factor_model, data = ABCD)
summary(some_factor_fit)
```

This doesn't run because ETA doesn't exist in the data. We need to use lavaan's syntax reserved for invisible variables:

- Factors: `FACTOR =~ VAR1 [+ VAR2 ...]`

```r
some_factor_model <- '
EXT_FAC =~ EXT.1 + EXT.2 + EXT.3 + EXT.4
'

some_factor_fit <- sem(some_factor_model, data = ABCD)
summary(some_factor_fit)
```

```
## lavaan 0.6-7 ended normally after 20 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                          8
##
##                                                   Used       Total
##   Number of observations                           761         964
##
## Model Test User Model:
##
##   Test statistic                                18.947
##   Degrees of freedom                                 2
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   EXT_FAC =~
##     EXT.1             1.000
```

```
##     EXT.2             1.001    0.053   19.055    0.000
##     EXT.3             1.009    0.057   17.659    0.000
##     EXT.4             1.127    0.057   19.704    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .EXT.1             0.309    0.020   15.468    0.000
##    .EXT.2             0.228    0.016   13.933    0.000
##    .EXT.3             0.344    0.022   15.825    0.000
##    .EXT.4             0.220    0.018   12.206    0.000
##     EXT_FAC           0.342    0.032   10.745    0.000
```

Now EXT_FAC is a variable like any other, except that we're inferring it from the observed variables (i.e., the indicators). Notice that `EXT_FAC` gets its own variance. We can include it in regressions, and we can also get its mean.

```
some_factor_model <- '
EXT_FAC =~ EXT.1 + EXT.2 + EXT.3 + EXT.4
VS.1 ~ EXT_FAC
'

some_factor_fit <- sem(some_factor_model, data = ABCD, meanstructure = TRUE)
summary(some_factor_fit)
```

```
## lavaan 0.6-7 ended normally after 29 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         15
##
##                                                   Used       Total
##   Number of observations                           761         964
##
## Model Test User Model:
##
##   Test statistic                                24.119
##   Degrees of freedom                                 5
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   EXT_FAC =~
##     EXT.1             1.000
##     EXT.2             1.003    0.053   19.042    0.000
##     EXT.3             1.009    0.057   17.617    0.000
##     EXT.4             1.132    0.057   19.721    0.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
```

```
##   VS.1 ~
##     EXT_FAC           0.312    0.093    3.366    0.001
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .EXT.1             2.448    0.029   83.680    0.000
##    .EXT.2             2.515    0.027   91.844    0.000
##    .EXT.3             2.496    0.030   82.740    0.000
##    .EXT.4             2.645    0.029   90.282    0.000
##    .VS.1              6.227    0.050  125.176    0.000
##     EXT_FAC           0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .EXT.1             0.311    0.020   15.524    0.000
##    .EXT.2             0.228    0.016   13.960    0.000
##    .EXT.3             0.346    0.022   15.872    0.000
##    .EXT.4             0.218    0.018   12.141    0.000
##    .VS.1              1.850    0.095   19.444    0.000
##     EXT_FAC           0.340    0.032   10.722    0.000
```

Is it weird that the mean of `EXT_FAC` is 0? It's actually the default in this kind of model to *set* it to 0. We'll see later when we run a latent growth model that it's allowed to be different from 0. However, this requires us to set other constraints.

"True score"

Using latent variables means we get more precision

```
ABCD_means <- ABCD %>%
  group_by(id) %>%
  mutate(EXT_MEAN = mean(c(EXT.1, EXT.2, EXT.3, EXT.4)),
         VS_MEAN = mean(c(VS.1, VS.2, VS.3, VS.4)))

factor_model <- '
EXT_FAC =~ EXT.1 + EXT.2 + EXT.3 + EXT.4
VS_FAC =~ VS.1 + VS.2 + VS.3 +VS.4
VS_FAC ~~ EXT_FAC
'

factor_fit <- sem(factor_model, data = ABCD)
summary(factor_fit, stan = TRUE)
```

```
## lavaan 0.6-7 ended normally after 32 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         17
##
##                                                   Used       Total
##   Number of observations                           732         964
##
## Model Test User Model:
##
##   Test statistic                                78.610
```

```
##   Degrees of freedom                                19
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   EXT_FAC =~
##     EXT.1             1.000                               0.579    0.719
##     EXT.2             1.006    0.054   18.677    0.000    0.583    0.780
##     EXT.3             1.028    0.059   17.378    0.000    0.596    0.716
##     EXT.4             1.135    0.059   19.171    0.000    0.657    0.811
##   VS_FAC =~
##     VS.1              1.000                               0.831    0.609
##     VS.2              1.202    0.092   13.068    0.000    0.999    0.669
##     VS.3              1.334    0.097   13.737    0.000    1.109    0.753
##     VS.4              1.200    0.092   13.002    0.000    0.998    0.663
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   EXT_FAC ~~
##     VS_FAC            0.110    0.023    4.695    0.000    0.228    0.228
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .EXT.1             0.313    0.020   15.333    0.000    0.313    0.483
##    .EXT.2             0.219    0.016   13.537    0.000    0.219    0.392
##    .EXT.3             0.338    0.022   15.410    0.000    0.338    0.488
##    .EXT.4             0.225    0.018   12.242    0.000    0.225    0.343
##    .VS.1              1.173    0.074   15.821    0.000    1.173    0.629
##    .VS.2              1.229    0.085   14.493    0.000    1.229    0.552
##    .VS.3              0.936    0.080   11.655    0.000    0.936    0.432
##    .VS.4              1.266    0.086   14.646    0.000    1.266    0.560
##     EXT_FAC           0.336    0.032   10.448    0.000    1.000    1.000
##     VS_FAC            0.691    0.086    7.993    0.000    1.000    1.000
```

```r
(scale_score_cor <- cor(ABCD_means[, c('EXT_MEAN', 'VS_MEAN')], use = 'pairwise.complete.obs'))
```

```
##           EXT_MEAN   VS_MEAN
## EXT_MEAN 1.0000000 0.1819191
## VS_MEAN  0.1819191 1.0000000
```

```r
std <- standardizedSolution(factor_fit)
```

The correlation for the latent variable model is 0.23 versus 0.18 for the zero-order correlation of the computed scale scores.

Important `lavaan` & `semPlot` stuff

Parameter specificaiton

We can “fix” parameters (e.g., factor loadings, covariances) by multiplying the variable by the value. To set a loading or covariance equal to 0, we can write `factor =~ 0*y1 + ...` or `EXT ~~ 0*VS`. *Anything that is not freely estimated in your model or by `lavaan` defaults is implicitly fixed to 0.* When reviewing your models you should make sure you’re okay with both the paths that do exist as well as with the paths that are left unspeficied. Sometimes fixing a path to 0 is a stronger theoretical statement than letting it be estimated freely.

Additionally you can add in equality constraints by multiplying by some label (e.g., `y ~ some_label*x1`, where `some_label` is an arbitrary string) and re-using the same label on some other parameter: `y ~ some_label*x1 + some_label*x2`; here both regression coefficients would be estimated exactly equal. This is a great way to salvage degrees of freedom if those constraints can be justified.

Missing data

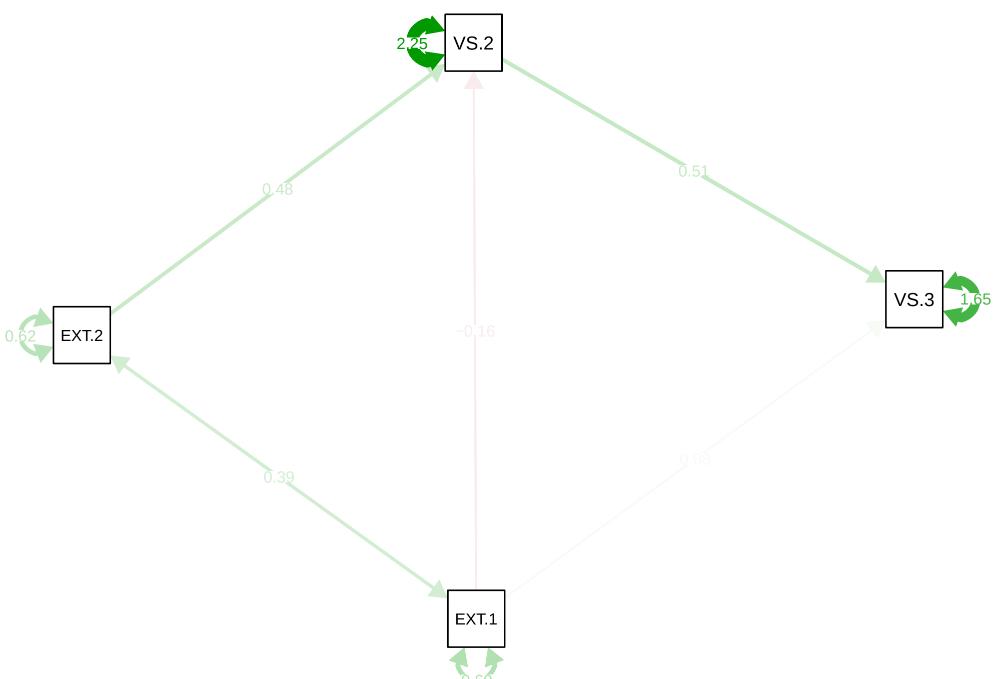
To specify how missing data are handled, we can use `missing = 'listwise'` to choose to delete individuals with missing data. The better more common approach is to use `missing = 'ML'` to use full information maximum likelihood (FIML, also known just as ML; this is distinct from restricted maximum likelihood [ReML]).

Never use listwise deletion when you can use ML. ML estimation protects you from bias in more situations. See this paper for more information.

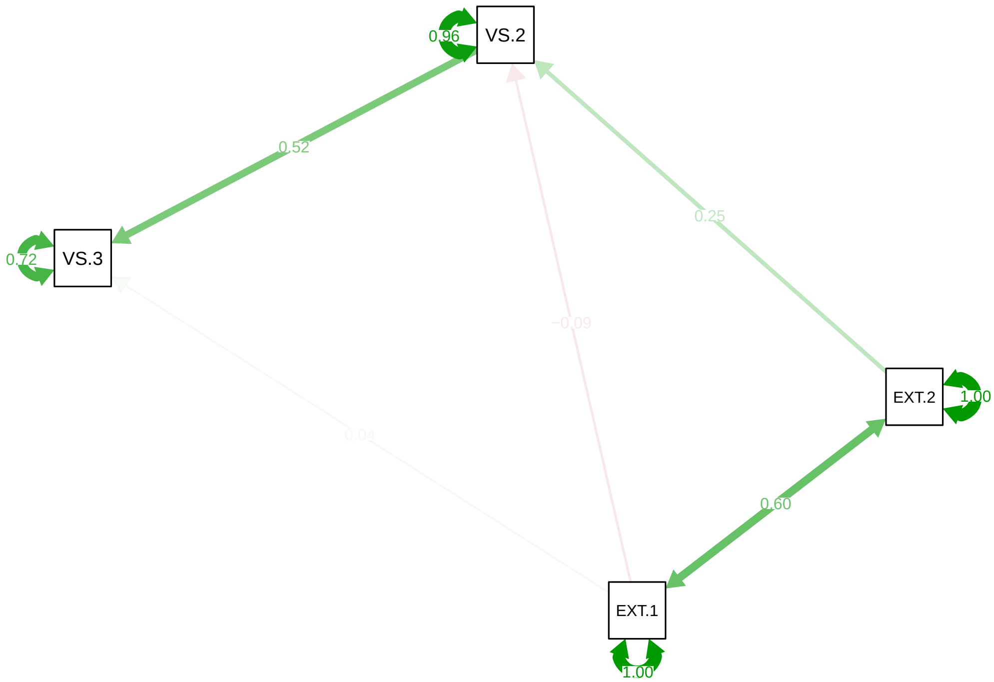
Useful info in plots

We can add useful info to these plots:

```r
semPlot::semPaths(slightly_complicated_sem_fit, what = 'est', layout = 'spring')
```




```r
semPlot::semPaths(slightly_complicated_sem_fit, what = 'std', layout = 'spring')
```



Standardized coefficients

We can also get standardized paths from `lavaan`.

```
summary(slightly_complicated_sem_fit, standardized = TRUE)
```

```
## lavaan 0.6-7 ended normally after 30 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                          9
##
##                                                   Used       Total
##   Number of observations                           820         964
##
## Model Test User Model:
##
##   Test statistic                                 0.096
##   Degrees of freedom                                 1
##   P-value (Chi-square)                           0.757
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   VS.2 ~
##     EXT.1            -0.163    0.079   -2.074    0.038   -0.163   -0.088
##     EXT.2             0.478    0.083    5.773    0.000    0.478    0.246
##   VS.3 ~
```

```
##     VS.2              0.513    0.029   17.505    0.000    0.513    0.521
##     EXT.1             0.080    0.054    1.470    0.141    0.080    0.044
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   EXT.1 ~~
##     EXT.2             0.389    0.027   14.679    0.000    0.389    0.597
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .VS.2              2.246    0.111   20.248    0.000    2.246    0.958
##    .VS.3              1.647    0.081   20.248    0.000    1.647    0.724
##     EXT.1             0.685    0.034   20.248    0.000    0.685    1.000
##     EXT.2             0.621    0.031   20.248    0.000    0.621    1.000
```

The Std.all column isi usually what you want to look at. Covariances in this column are equivalent to correlations (or partial correlations, if they are covariances between residuals). The Std.lv column shows values standardized only on the latent variables. We don't have any so they are equivalent to the Estimates.

Autoregressive-type Models

Autoregressive Cross-Lag Panel Models

Now we're starting to get into the longitudinalness of it all.

Remember the AR part of the model is just a series of regressions:

```
ARCL <- '
y4 ~ y3
y3 ~ y2
y2 ~ y1'
```

When we add in a second variable, we add in that variable's AR model,

```
ARCL <- '
#y
y4 ~ y3
y3 ~ y2
y2 ~ y1

#x
x4 ~ x3
x3 ~ x2
x2 ~ x1'
```

and we can also now add in the cross-lag part ($x_t = y_{t-1} + \ldots$),

```
ARCL <- '
#y
y4 ~ y3 + x3
y3 ~ y2 + x2
y2 ~ y1 + x1

#x
x4 ~ x3 + y3
x3 ~ x2 + y2
x2 ~ x1 + y1'
```

as well as the contemporaneous (residual) correlations.

```
ARCL <- '
#y
y4 ~ y3 + x3
y3 ~ y2 + x2
y2 ~ y1 + x1

#x
x4 ~ x3 + y3
x3 ~ x2 + y2
x2 ~ x1 + y1

#x ~~ y
x3 ~~ y3
x2 ~~ y2
x1 ~~ y1'
```

Let's look at this for the variables in the sample data:

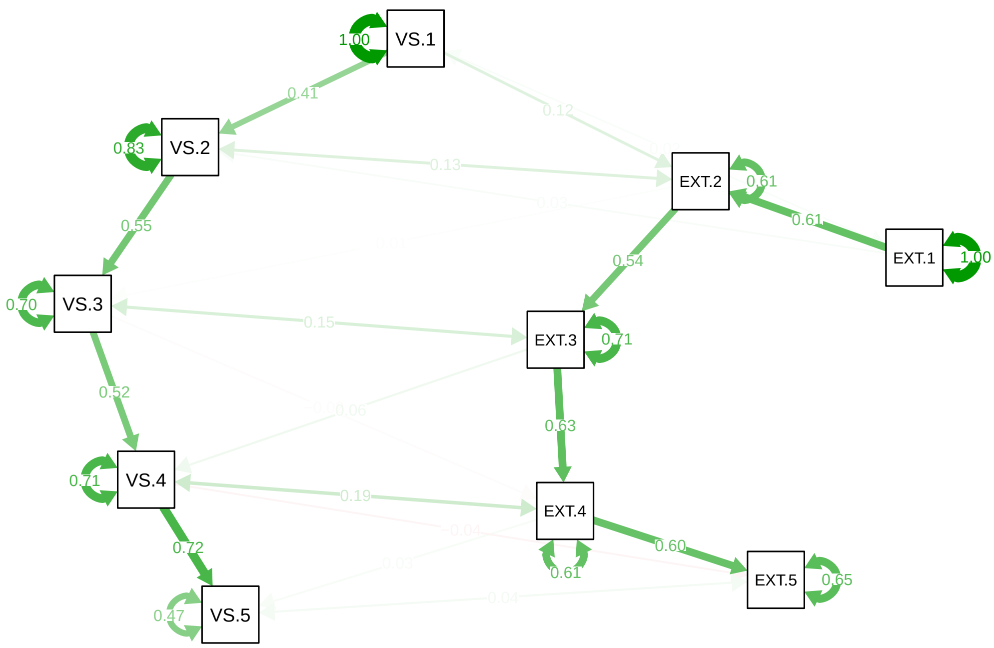
```
ARCL_model <- '
# Regressions for VS Activation
VS.2 ~ VS.1 + EXT.1
VS.3 ~ VS.2 + EXT.2
VS.4 ~ VS.3 + EXT.3
VS.5 ~ VS.4 + EXT.4

# Regressions for Externalizing Behavior
EXT.2 ~ EXT.1 + VS.1
EXT.3 ~ EXT.2 + VS.2
EXT.4 ~ EXT.3 + VS.3
EXT.5 ~ EXT.4 + VS.4

# Within-Time (Residual) Correlations
VS.1 ~~ EXT.1
VS.2 ~~ EXT.2
VS.3 ~~ EXT.3
VS.4 ~~ EXT.4
VS.5 ~~ EXT.5'

ARCL_fit <- sem(ARCL_model, data = ABCD, meanstructure = TRUE, missing = 'ML')
summary(ARCL_fit)
```

```
## lavaan 0.6-7 ended normally after 58 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         41
##
##   Number of observations                           964
##   Number of missing patterns                        47
##
## Model Test User Model:
##
##   Test statistic                               660.590
##   Degrees of freedom                                24
```

```
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   VS.2 ~
##     VS.1              0.444    0.033   13.596    0.000
##     EXT.1             0.053    0.057    0.916    0.360
##   VS.3 ~
##     VS.2              0.535    0.028   18.866    0.000
##     EXT.2             0.022    0.056    0.404    0.686
##   VS.4 ~
##     VS.3              0.524    0.030   17.494    0.000
##     EXT.3             0.102    0.053    1.912    0.056
##   VS.5 ~
##     VS.4              0.742    0.025   30.146    0.000
##     EXT.4             0.061    0.047    1.297    0.195
##   EXT.2 ~
##     EXT.1             0.578    0.026   22.435    0.000
##     VS.1              0.069    0.015    4.720    0.000
##   EXT.3 ~
##     EXT.2             0.571    0.033   17.523    0.000
##     VS.2              0.001    0.016    0.041    0.967
##   EXT.4 ~
##     EXT.3             0.593    0.027   21.773    0.000
##     VS.3             -0.009    0.015   -0.583    0.560
##   EXT.5 ~
##     EXT.4             0.910    0.044   20.538    0.000
##     VS.4             -0.031    0.024   -1.293    0.196
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   VS.1 ~~
##     EXT.1             0.029    0.041    0.702    0.483
##  .VS.2 ~~
##    .EXT.2             0.123    0.032    3.850    0.000
##  .VS.3 ~~
##    .EXT.3             0.145    0.033    4.333    0.000
##  .VS.4 ~~
##    .EXT.4             0.163    0.031    5.334    0.000
##  .VS.5 ~~
##    .EXT.5             0.050    0.039    1.274    0.203
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.2              3.545    0.247   14.354    0.000
##    .VS.3              3.015    0.212   14.198    0.000
##    .VS.4              3.072    0.222   13.835    0.000
##    .VS.5              1.890    0.188   10.047    0.000
```

```
##    .EXT.2             0.627    0.110    5.694    0.000
##    .EXT.3             1.029    0.122    8.421    0.000
##    .EXT.4             1.199    0.113   10.616    0.000
##    .EXT.5             0.539    0.181    2.984    0.003
##     VS.1              6.239    0.047  132.084    0.000
##     EXT.1             2.337    0.028   83.108    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.2              2.088    0.098   21.370    0.000
##    .VS.3              1.679    0.080   21.076    0.000
##    .VS.4              1.716    0.083   20.793    0.000
##    .VS.5              1.209    0.058   20.974    0.000
##    .EXT.2             0.411    0.020   21.074    0.000
##    .EXT.3             0.541    0.026   20.931    0.000
##    .EXT.4             0.416    0.021   20.113    0.000
##    .EXT.5             1.030    0.051   20.347    0.000
##     VS.1              2.151    0.098   21.955    0.000
##     EXT.1             0.740    0.034   21.558    0.000
```

```
semPaths(ARCL_fit, layout='spring', intercepts=FALSE, what = 'std')
```



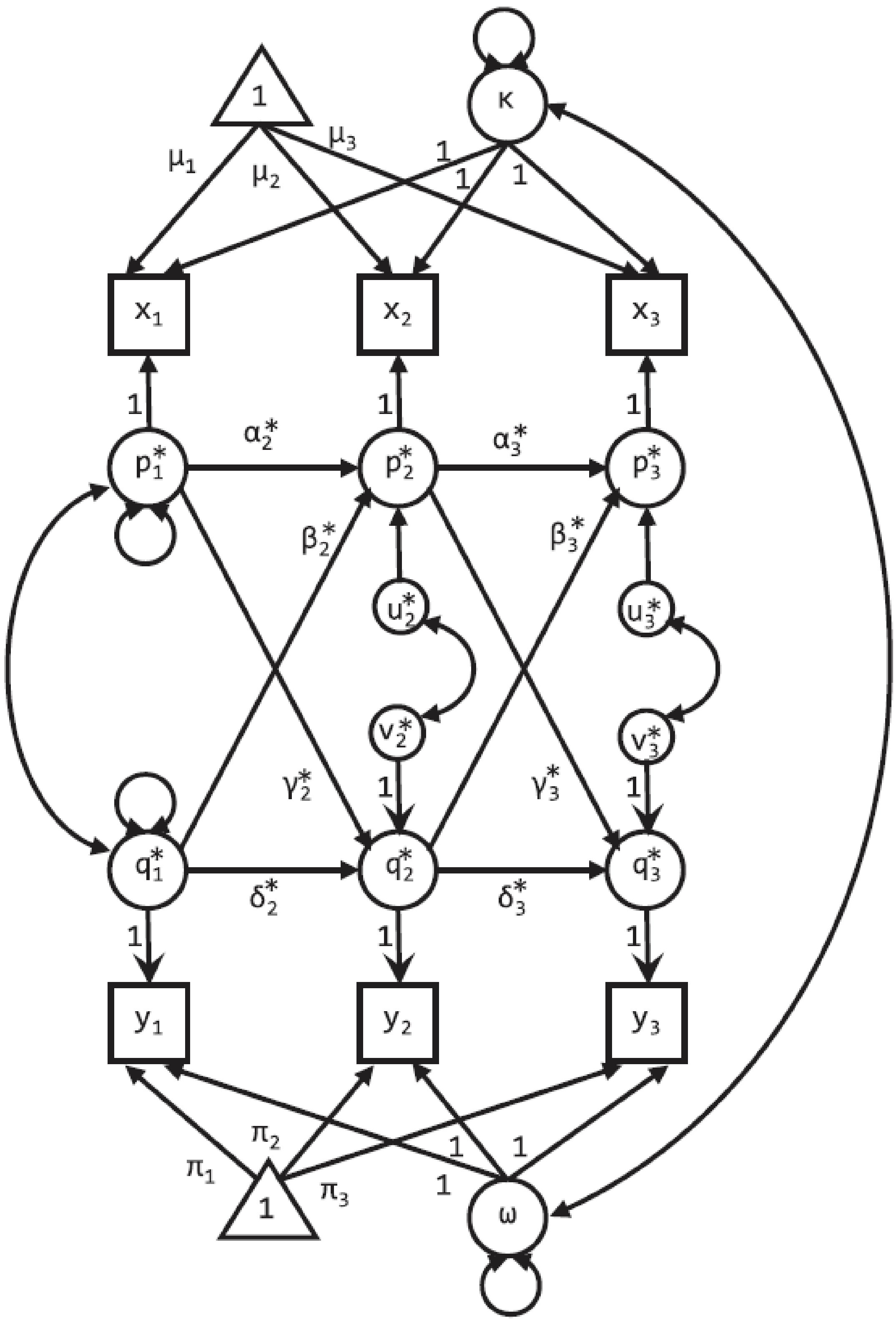
BUT REMEMBER NEVER USE THIS MODEL!

RI-ARCL (RI-CLPM)

We can use the riclpmr pacakge to generate code for this model using a list of variable names grouped by construct.

See also this blog post.

RI-CLPM with means



Notice a few things. First, we're estimating a mean for each variable. But we are also estimating a factor for each variable. The loadings are all set to 1. Why? In the factor model we saw before, the loadings were allowed to be free because we wanted the model to decide how much variance was due to a common factor versus error variance (this is the function of a measurement model). Here, we start with the assumption that the measurment model is well defined (in fact, we're assuming that the observations ($x_{1..3}$ and $y_{1..3}$) are all measured perfectly. What is this factor for then? It encodes the theoretical assumption that there is a stable trait for each variable x and y that causes the measurements. It's stable, so it causes them equally over time.

In other words, the latent factors κ and ω partial out stable, between-person (trait-like) differences from the observations. What's left behind are the residuals. We can conceptualize the residuals as the perturbations in the observations that are not explained by trait means. In other words, they are within-person changes unexplained by a stable mean score.

We can give "structure" to the residuals by turning them into factors ($p_{1..3}$ and $q_{1..3}$) which we can then use as variables in an ARCL model that describes the within-person change. Now you can see that each observation is determined by the between person factors (κ and ω), and the within-person factors ($p_{1..3}$ and $q_{1..3}$), with all of their loadings fixed to 1.

We still have residual variance left over in $u_{2,3}$ and $v_{2,3}$.

This becomes a lot to specify, and the lavaan defaults don't always play nice with these models. To help estimate these things I (John) wrote a little package to help.

```
#https://johnflournoy.science/riclpmr/
#devtools::install_github('jflournoy/riclpmr')
library(riclpmr)

variable_groups <- list(
  EXT = c('EXT.1','EXT.2', 'EXT.3', 'EXT.4', 'EXT.5'),
  VS =  c('VS.1','VS.2', 'VS.3', 'VS.4', 'VS.5'))
RIARCL_model <- riclpmr::riclpm_text(var_groups = variable_groups, constrain_over_waves = FALSE)

#check the model code
cat(RIARCL_model)
```

```
## ri_EXT =~ 1*EXT.1 + 1*EXT.2 + 1*EXT.3 + 1*EXT.4 + 1*EXT.5
## ri_VS =~ 1*VS.1 + 1*VS.2 + 1*VS.3 + 1*VS.4 + 1*VS.5
## ri_EXT ~~ ri_EXT
## ri_VS ~~ ri_VS
## ri_EXT ~~ ri_VS
## EXT.1 ~ EXT.1_mu*1
## EXT.2 ~ EXT.2_mu*1
## EXT.3 ~ EXT.3_mu*1
## EXT.4 ~ EXT.4_mu*1
## EXT.5 ~ EXT.5_mu*1
## VS.1 ~ VS.1_mu*1
## VS.2 ~ VS.2_mu*1
## VS.3 ~ VS.3_mu*1
## VS.4 ~ VS.4_mu*1
## VS.5 ~ VS.5_mu*1
## lat_EXT1 =~ 1*EXT.1
## lat_EXT2 =~ 1*EXT.2
## lat_EXT3 =~ 1*EXT.3
## lat_EXT4 =~ 1*EXT.4
## lat_EXT5 =~ 1*EXT.5
## lat_VS1 =~ 1*VS.1
## lat_VS2 =~ 1*VS.2
```

```
## lat_VS3 =~ 1*VS.3
## lat_VS4 =~ 1*VS.4
## lat_VS5 =~ 1*VS.5
## lat_EXT1 ~~ lat_VS1
## lat_EXT2 ~~ lat_VS2
## lat_EXT3 ~~ lat_VS3
## lat_EXT4 ~~ lat_VS4
## lat_EXT5 ~~ lat_VS5
## lat_EXT2 ~ lat_EXT1 + lat_VS1
## lat_EXT3 ~ lat_EXT2 + lat_VS2
## lat_EXT4 ~ lat_EXT3 + lat_VS3
## lat_EXT5 ~ lat_EXT4 + lat_VS4
## lat_EXT2 ~ lat_EXT5 + lat_VS5
## lat_VS2 ~ lat_EXT1 + lat_VS1
## lat_VS3 ~ lat_EXT2 + lat_VS2
## lat_VS4 ~ lat_EXT3 + lat_VS3
## lat_VS5 ~ lat_EXT4 + lat_VS4
## lat_VS2 ~ lat_EXT5 + lat_VS5
## lat_EXT1 ~~ lat_EXT1
## lat_EXT2 ~~ lat_EXT2
## lat_EXT3 ~~ lat_EXT3
## lat_EXT4 ~~ lat_EXT4
## lat_EXT5 ~~ lat_EXT5
## lat_VS1 ~~ lat_VS1
## lat_VS2 ~~ lat_VS2
## lat_VS3 ~~ lat_VS3
## lat_VS4 ~~ lat_VS4
## lat_VS5 ~~ lat_VS5
```

```r
RIARCL_fit <- riclpmr::lavriclpm(RIARCL_model, data = ABCD, missing = 'ML')
summary(RIARCL_fit)
```

```
## lavaan 0.6-7 ended normally after 65 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         48
##
##   Number of observations                           964
##   Number of missing patterns                        47
##
## Model Test User Model:
##
##   Test statistic                               131.872
##   Degrees of freedom                                17
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
```

```
##   ri_EXT =~
##     EXT.1             1.000
##     EXT.2             1.000
##     EXT.3             1.000
##     EXT.4             1.000
##     EXT.5             1.000
##   ri_VS =~
##     VS.1              1.000
##     VS.2              1.000
##     VS.3              1.000
##     VS.4              1.000
##     VS.5              1.000
##   lat_EXT1 =~
##     EXT.1             1.000
##   lat_EXT2 =~
##     EXT.2             1.000
##   lat_EXT3 =~
##     EXT.3             1.000
##   lat_EXT4 =~
##     EXT.4             1.000
##   lat_EXT5 =~
##     EXT.5             1.000
##   lat_VS1 =~
##     VS.1              1.000
##   lat_VS2 =~
##     VS.2              1.000
##   lat_VS3 =~
##     VS.3              1.000
##   lat_VS4 =~
##     VS.4              1.000
##   lat_VS5 =~
##     VS.5              1.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   lat_EXT2 ~
##     lat_EXT1          0.034    0.052    0.657    0.511
##     lat_VS1           0.034    0.024    1.372    0.170
##   lat_EXT3 ~
##     lat_EXT2         -0.179    0.088   -2.029    0.042
##     lat_VS2           0.005    0.027    0.182    0.856
##   lat_EXT4 ~
##     lat_EXT3          0.118    0.052    2.260    0.024
##     lat_VS3          -0.025    0.031   -0.787    0.431
##   lat_EXT5 ~
##     lat_EXT4          0.722    0.088    8.244    0.000
##     lat_VS4          -0.054    0.034   -1.607    0.108
##   lat_EXT2 ~
##     lat_EXT5          0.002    0.023    0.102    0.919
##     lat_VS5           0.050    0.030    1.664    0.096
##   lat_VS2 ~
##     lat_EXT1         -0.234    0.105   -2.223    0.026
##     lat_VS1           0.048    0.056    0.867    0.386
##   lat_VS3 ~
```

```
##     lat_EXT2          0.018    0.186    0.097    0.922
##     lat_VS2           0.199    0.049    4.085    0.000
##   lat_VS4 ~
##     lat_EXT3          0.149    0.099    1.501    0.133
##     lat_VS3           0.202    0.053    3.793    0.000
##   lat_VS5 ~
##     lat_EXT4         -0.133    0.114   -1.163    0.245
##     lat_VS4           0.541    0.037   14.585    0.000
##   lat_VS2 ~
##     lat_EXT5          0.085    0.048    1.746    0.081
##     lat_VS5           0.215    0.051    4.208    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   ri_EXT ~~
##     ri_VS             0.131    0.034    3.821    0.000
##   lat_EXT1 ~~
##     lat_VS1          -0.095    0.033   -2.863    0.004
##  .lat_EXT2 ~~
##    .lat_VS2           0.068    0.034    2.012    0.044
##  .lat_EXT3 ~~
##    .lat_VS3           0.128    0.047    2.748    0.006
##  .lat_EXT4 ~~
##    .lat_VS4           0.126    0.030    4.144    0.000
##  .lat_EXT5 ~~
##    .lat_VS5           0.004    0.038    0.100    0.921
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .EXT.1  (EXT.1)    2.337    0.029   81.726    0.000
##    .EXT.2  (EXT.2)    2.412    0.026   91.185    0.000
##    .EXT.3  (EXT.3)    2.405    0.029   82.313    0.000
##    .EXT.4  (EXT.4)    2.541    0.028   91.311    0.000
##    .EXT.5  (EXT.5)    2.640    0.043   61.631    0.000
##    .VS.1    (VS.1)    6.239    0.048  128.861    0.000
##    .VS.2    (VS.2)    6.442    0.052  123.567    0.000
##    .VS.3    (VS.3)    6.527    0.050  131.635    0.000
##    .VS.4    (VS.4)    6.730    0.052  128.295    0.000
##    .VS.5    (VS.5)    7.041    0.052  136.716    0.000
##     ri_EXT            0.000
##     ri_VS             0.000
##     l_EXT1            0.000
##    .l_EXT2            0.000
##    .l_EXT3            0.000
##    .l_EXT4            0.000
##    .l_EXT5            0.000
##     lt_VS1            0.000
##    .lt_VS2            0.000
##    .lt_VS3            0.000
##    .lt_VS4            0.000
##    .lt_VS5            0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
```

```
##     ri_EXT           0.417    0.024    17.066    0.000
##     ri_VS            0.930    0.076    12.246    0.000
##     lat_EXT1         0.351    0.025    14.193    0.000
##    .lat_EXT2         0.228    0.022    10.235    0.000
##    .lat_EXT3         0.352    0.033    10.609    0.000
##    .lat_EXT4         0.263    0.018    14.270    0.000
##    .lat_EXT5         1.010    0.050    20.086    0.000
##     lat_VS1          1.330    0.085    15.600    0.000
##    .lat_VS2          1.490    0.095    15.617    0.000
##    .lat_VS3          1.279    0.082    15.613    0.000
##    .lat_VS4          1.478    0.087    17.031    0.000
##    .lat_VS5          1.019    0.055    18.464    0.000
##    .EXT.1            0.000
##    .EXT.2            0.000
##    .EXT.3            0.000
##    .EXT.4            0.000
##    .EXT.5            0.000
##    .VS.1             0.000
##    .VS.2             0.000
##    .VS.3             0.000
##    .VS.4             0.000
##    .VS.5             0.000
```

Note: This model centers the between-person differences at 0, so above the means for the variables `ri_EXT` and `ri_VS` are both 0. We can use a slightly different model, called the LCM-SR (see below) to create a mean-growth model that is better at describing mean change.

Mean change using latent variables

Intercept and slope latent curve model

Set aside the auto-regressive cross-lag model for a minute, but keep in mind the latent intercept we saw in the RI-CLPM. As is the case in a multilevel growth model, we can describe a series of observations as being a function of an intercept and a slope (with each person getting their own values for these, as in the "random slopes random intercepts" model).

We start by specifying the intercept, very much like in the RI-CLPM:

```
Int =~ 1*y1 + 1*y2 + 1*y3 + 1*y4
```

Each person gets a value for Int that contributes equally to each of the observed `y`.

We continue by adding a slope:

```
int =~ 1*y1 + 1*y2 + 1*y3 + 1*y4
slp =~ 0*y1 + 1*y2 + 2*y3 + 3*y4
```

The slope latent variable increases its influence on the observed `y` more at each wave. The loadings (0,1,2,3) are essentially the `TIME` variable in a multilevel model. When the loading for slope is 0, the only contribution to the observed variables is the intercept. This is the timepoint for which the mean of the intercept factor is the expected mean of the data. In other words, time is "centered" at whatever timepoint you set the slope factor to 0.

Let's look at this using the data we have for externalizing:

```
# Linear Slope: Externalizing Behavior
LCM_VSlin_model = 'int =~ 1*EXT.1 + 1*EXT.2 + 1*EXT.3 + 1*EXT.4 + 1*EXT.5
             slp =~ 0*EXT.1 + 1*EXT.2 + 2*EXT.3 + 3*EXT.4 + 4*EXT.5'
```

```r
LCM_VSlin_fit = growth(LCM_VSlin_model, data=ABCD, missing = 'ML')
```

```
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING: some cases
##   649 678
```

```r
summary(LCM_VSlin_fit)
```

```
## lavaan 0.6-7 ended normally after 29 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         10
##
##                                                   Used       Total
##   Number of observations                           962         964
##   Number of missing patterns                        13
##
## Model Test User Model:
##
##   Test statistic                                23.630
##   Degrees of freedom                                10
##   P-value (Chi-square)                           0.009
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int =~
##     EXT.1             1.000
##     EXT.2             1.000
##     EXT.3             1.000
##     EXT.4             1.000
##     EXT.5             1.000
##   slp =~
##     EXT.1             0.000
##     EXT.2             1.000
##     EXT.3             2.000
##     EXT.4             3.000
##     EXT.5             4.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int ~~
##     slp              -0.016    0.009   -1.928    0.054
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .EXT.1             0.000
##    .EXT.2             0.000
##    .EXT.3             0.000
```

```
##    .EXT.4             0.000
##    .EXT.5             0.000
##     int               2.329    0.025   91.329    0.000
##     slp               0.068    0.008    8.170    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .EXT.1             0.288    0.024   12.083    0.000
##    .EXT.2             0.256    0.016   15.681    0.000
##    .EXT.3             0.370    0.021   17.566    0.000
##    .EXT.4             0.142    0.015    9.487    0.000
##    .EXT.5             0.881    0.051   17.439    0.000
##     int               0.430    0.031   13.754    0.000
##     slp               0.024    0.003    7.048    0.000
```

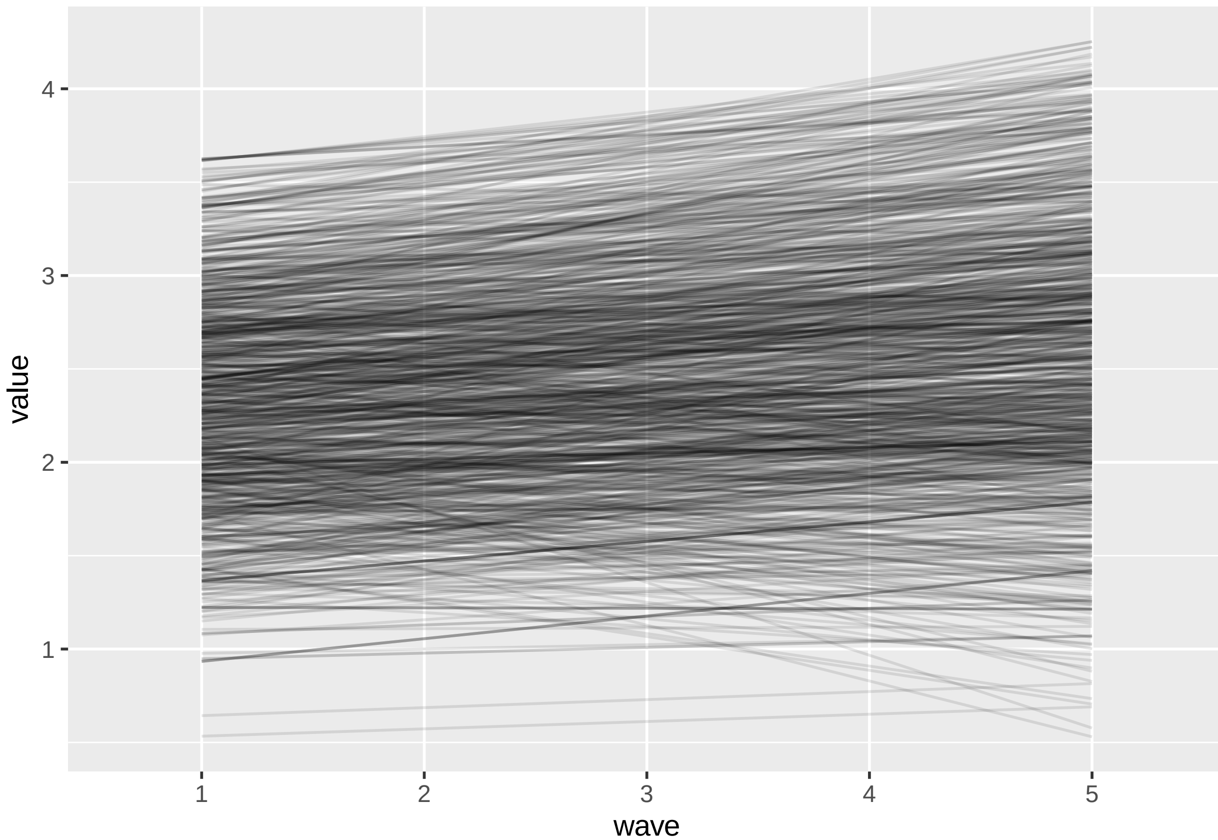
Notice we use the function `growth` now. This sets `lavaa` defaults to be appropriate for this (very common) model.

Plotting

We can plot each individual's trajectory.

```r
pred_data <- as.data.frame(lavPredict(LCM_VSlin_fit, type = 'yhat')) %>%
  mutate(id = 1:n()) %>%
  pivot_longer(cols = 1:5) %>%
  extract(name, c('var', 'wave'), '(EXT)\\.(\\d+)')

ggplot(pred_data) +
  geom_line(aes(x = wave, y = value, group = id), alpha = .1)
```

```
## Warning: Removed 10 row(s) containing missing values (geom_path).
```



Constrained latent growth models

The model above corresponds to a random intercept, random slope model. Let's play around with intercept-only models, and fixed intercepts and slopes.

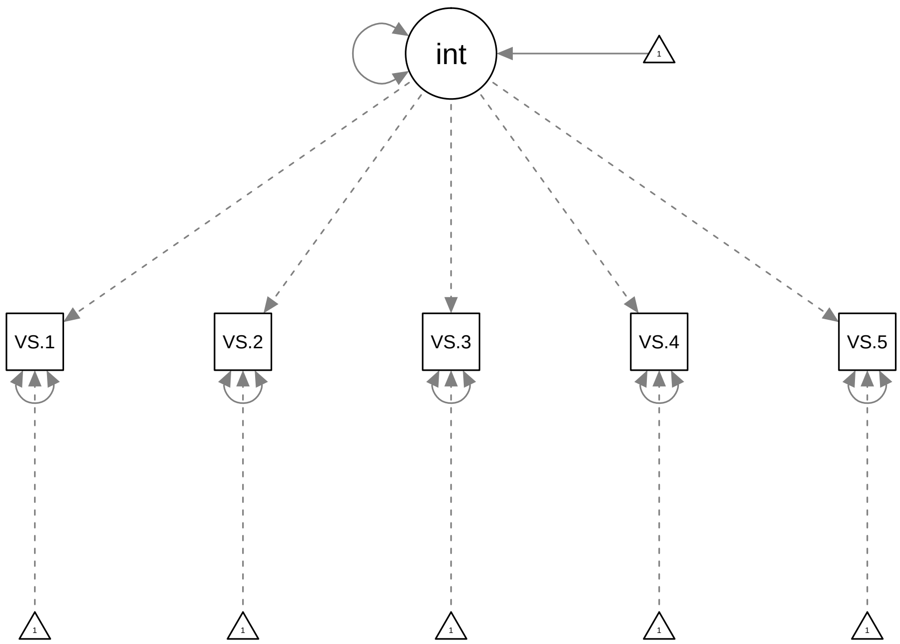
Intercept only:

```
# Intercept-Only Model: VS Activation
LCM_VSint = '
int =~ 1*VS.1 + 1*VS.2 + 1*VS.3 + 1*VS.4 + 1*VS.5'

LCM_VSint_fit = growth(LCM_VSint, data = ABCD, missing = 'ML')
summary(LCM_VSint_fit)
```

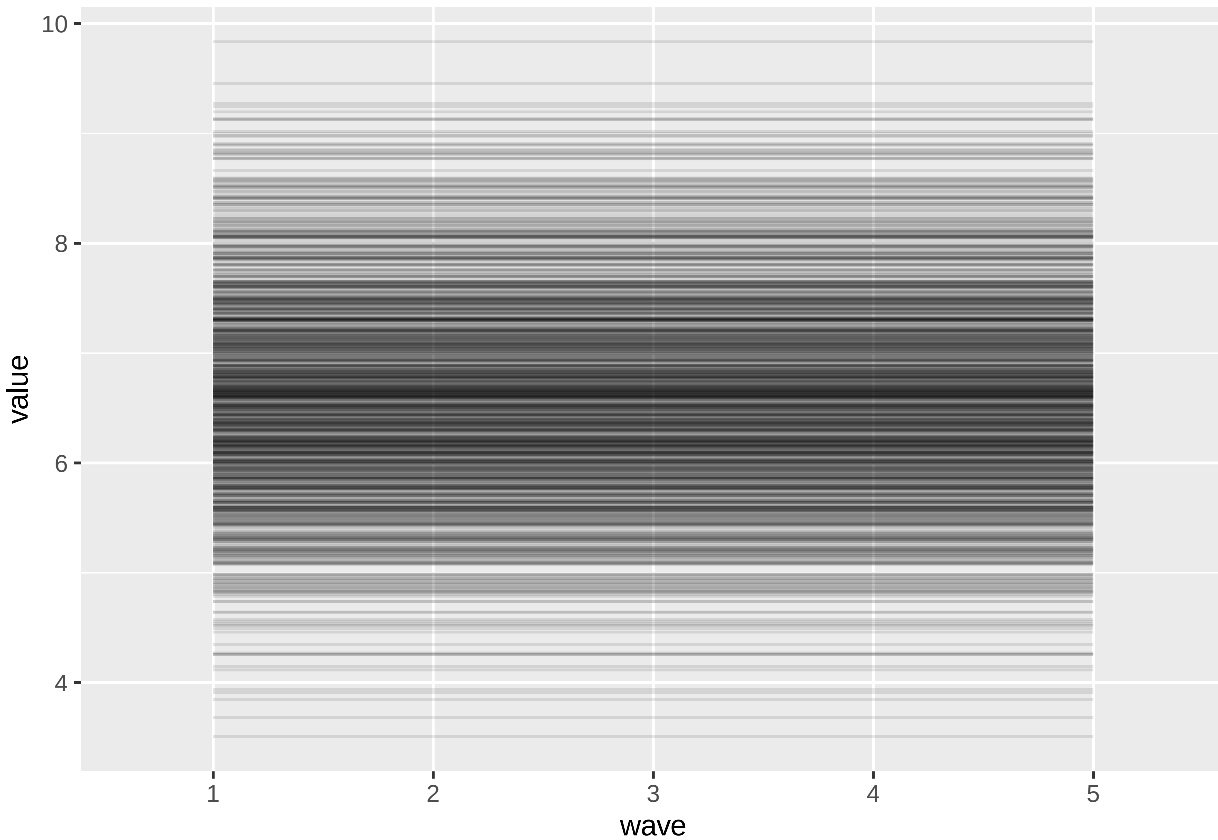
```
## lavaan 0.6-7 ended normally after 29 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                          7
##
##   Number of observations                           964
##   Number of missing patterns                         8
##
## Model Test User Model:
##
##   Test statistic                               506.230
##   Degrees of freedom                                13
##   P-value (Chi-square)                           0.000
```

```
##
## Parameter Estimates:
##
##   Standard errors                                 Standard
##   Information                                     Observed
##   Observed information based on                    Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int =~
##     VS.1              1.000
##     VS.2              1.000
##     VS.3              1.000
##     VS.4              1.000
##     VS.5              1.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              0.000
##    .VS.2              0.000
##    .VS.3              0.000
##    .VS.4              0.000
##    .VS.5              0.000
##     int               6.616    0.040  164.348    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              1.772    0.095   18.650    0.000
##    .VS.2              1.383    0.078   17.753    0.000
##    .VS.3              1.054    0.063   16.626    0.000
##    .VS.4              1.063    0.066   16.212    0.000
##    .VS.5              1.167    0.071   16.457    0.000
##     int               1.237    0.070   17.574    0.000
```

```
semPaths(LCM_VSint_fit, intercepts=TRUE)
```




```
pred_data <- as.data.frame(lavPredict(LCM_VSint_fit, type = 'yhat')) %>%
  mutate(id = 1:n()) %>%
  pivot_longer(cols = 1:5) %>%
  extract(name, c('var', 'wave'), '(VS)\\.(\\d+)')

ggplot(pred_data) +
  geom_line(aes(x = wave, y = value, group = id), alpha = .1)
```



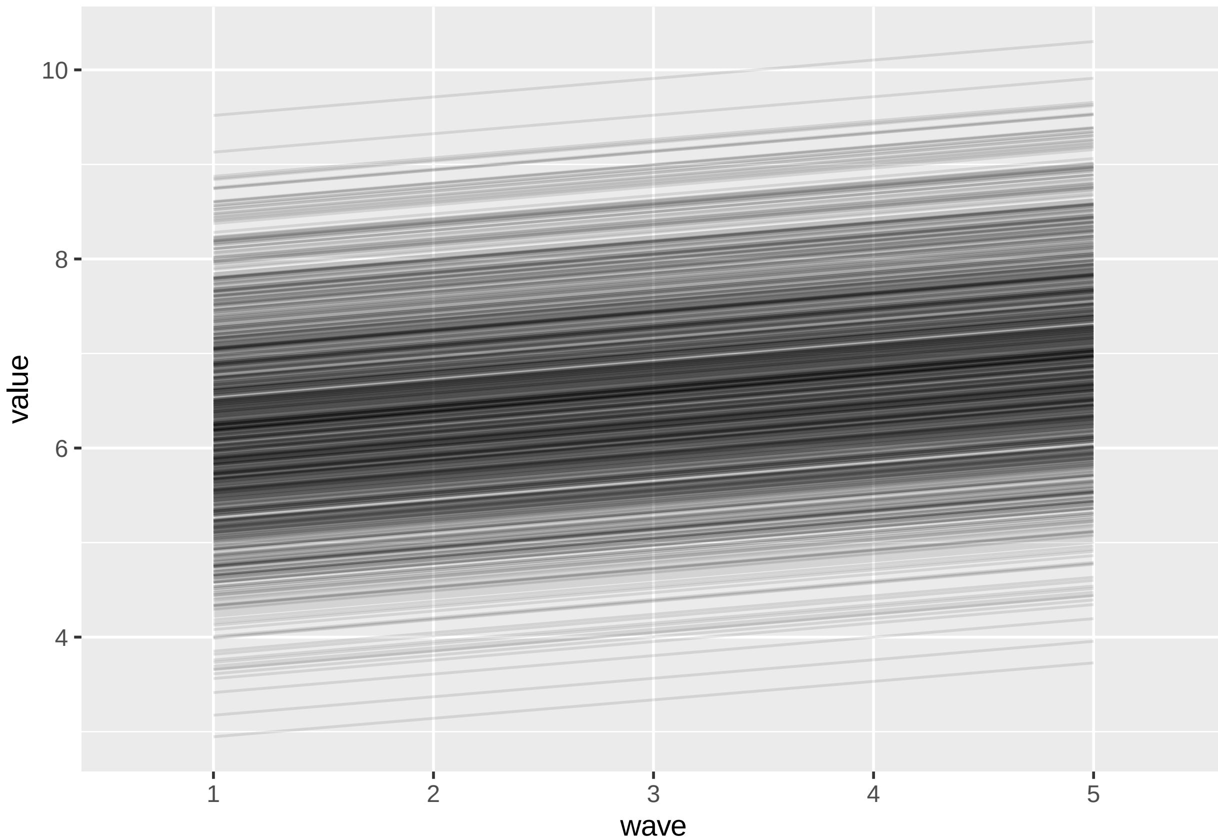
Slope variable, but slope is fixed:

```
# Linear Slope: VS Activation (fixed effect of time)
LCM_VSlin_fixed = '
int =~ 1*VS.1 + 1*VS.2 + 1*VS.3 + 1*VS.4 + 1*VS.5
slp =~ 0*VS.1 + 1*VS.2 + 2*VS.3 + 3*VS.4 + 4*VS.5
slp ~~ 0*slp
slp ~~ 0*int #notice we also have to constrain this!'

LCM_VSlin_fixed_fit = growth(LCM_VSlin_fixed, data=ABCD, missing='ML')
summary(LCM_VSlin_fixed_fit)
```

```
## lavaan 0.6-7 ended normally after 33 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                          8
##
##   Number of observations                           964
##   Number of missing patterns                         8
##
## Model Test User Model:
##
##   Test statistic                               247.515
##   Degrees of freedom                                12
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
```

```
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int =~
##     VS.1              1.000
##     VS.2              1.000
##     VS.3              1.000
##     VS.4              1.000
##     VS.5              1.000
##   slp =~
##     VS.1              0.000
##     VS.2              1.000
##     VS.3              2.000
##     VS.4              3.000
##     VS.5              4.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int ~~
##     slp               0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              0.000
##    .VS.2              0.000
##    .VS.3              0.000
##    .VS.4              0.000
##    .VS.5              0.000
##     int               6.204    0.047  131.243    0.000
##     slp               0.195    0.012   16.686    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     slp               0.000
##    .VS.1              1.626    0.087   18.633    0.000
##    .VS.2              1.364    0.076   17.956    0.000
##    .VS.3              1.059    0.063   16.916    0.000
##    .VS.4              1.026    0.062   16.467    0.000
##    .VS.5              0.917    0.058   15.867    0.000
##     int               1.262    0.071   17.827    0.000
```

```r
pred_data <- as.data.frame(lavPredict(LCM_VSlin_fixed_fit, type = 'yhat')) %>%
  mutate(id = 1:n()) %>%
  pivot_longer(cols = 1:5) %>%
  extract(name, c('var', 'wave'), '(VS)\\.(\\d+)')

ggplot(pred_data) +
  geom_line(aes(x = wave, y = value, group = id), alpha = .1)
```



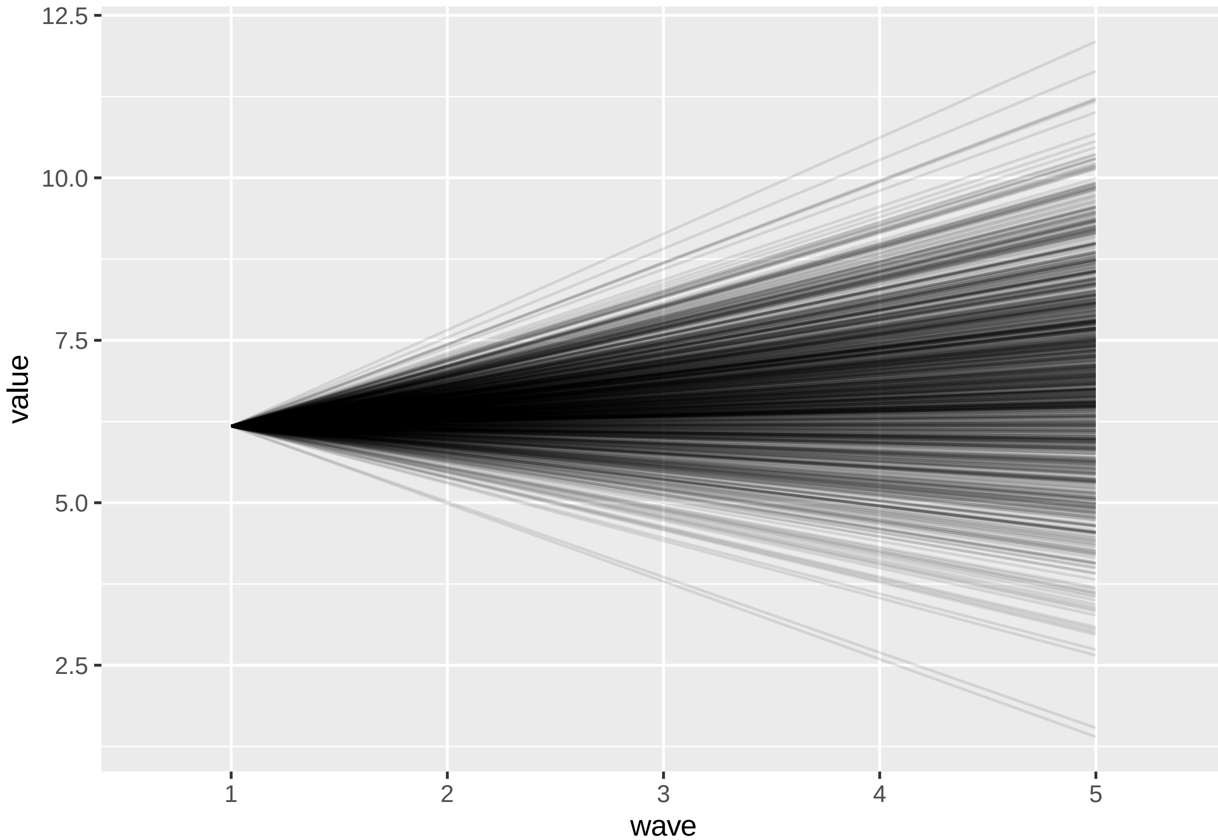
Fixed intercept, slope random:

```
# Linear Slope: VS Activation (fixed effect of intercept)
LCM_VSlin_fixedint = '
int =~ 1*VS.1 + 1*VS.2 + 1*VS.3 + 1*VS.4 + 1*VS.5
slp =~ 0*VS.1 + 1*VS.2 + 2*VS.3 + 3*VS.4 + 4*VS.5
int ~~ 0*int
int ~~ 0*slp'

LCM_VSlin_fixedint_fit = growth(LCM_VSlin_fixedint, data=ABCD, missing='ML')
summary(LCM_VSlin_fixedint_fit)
```

```
## lavaan 0.6-7 ended normally after 43 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                          8
##
##   Number of observations                           964
##   Number of missing patterns                         8
##
## Model Test User Model:
##
##   Test statistic                               467.927
##   Degrees of freedom                                12
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
```

```
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int =~
##     VS.1              1.000
##     VS.2              1.000
##     VS.3              1.000
##     VS.4              1.000
##     VS.5              1.000
##   slp =~
##     VS.1              0.000
##     VS.2              1.000
##     VS.3              2.000
##     VS.4              3.000
##     VS.5              4.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int ~~
##     slp               0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              0.000
##    .VS.2              0.000
##    .VS.3              0.000
##    .VS.4              0.000
##    .VS.5              0.000
##     int               6.179    0.034  184.223    0.000
##     slp               0.215    0.016   13.175    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     int               0.000
##    .VS.1              2.155    0.098   21.936    0.000
##    .VS.2              2.000    0.094   21.200    0.000
##    .VS.3              1.419    0.071   20.071    0.000
##    .VS.4              1.084    0.066   16.483    0.000
##    .VS.5              0.090    0.060    1.497    0.134
##     slp               0.161    0.008   19.169    0.000
```

```
pred_data <- as.data.frame(lavPredict(LCM_VSlin_fixedint_fit, type = 'yhat')) %>%
  mutate(id = 1:n()) %>%
  pivot_longer(cols = 1:5) %>%
  extract(name, c('var', 'wave'), '(VS)\\.(\\d+)')

ggplot(pred_data) +
  geom_line(aes(x = wave, y = value, group = id), alpha = .1)
```



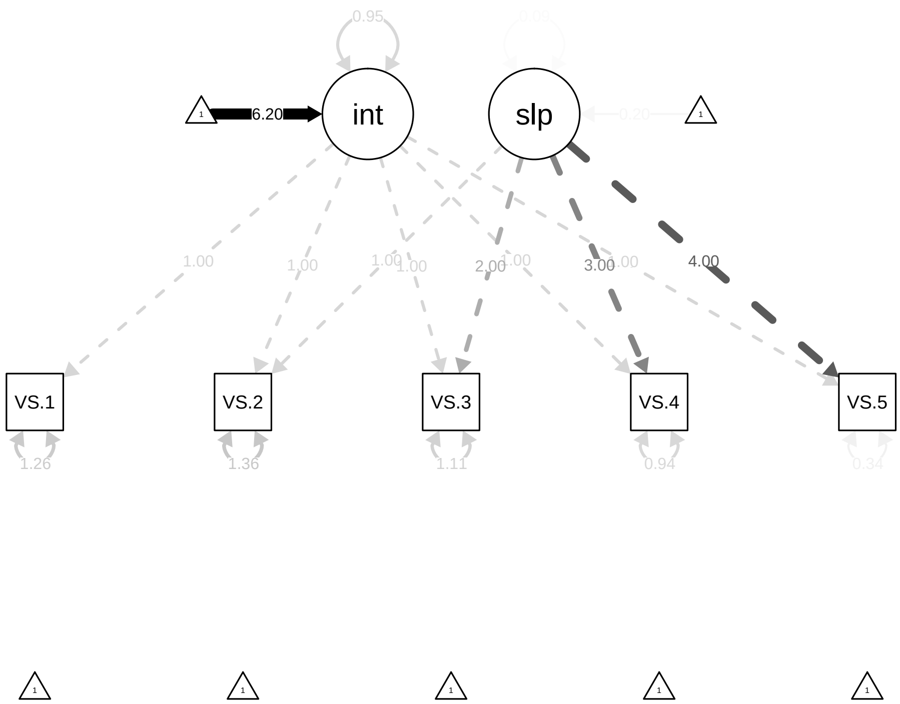
Random intercept and slope:

```
# Linear Slope: VS Activation (random effect of time)
LCM_VSlin = '
int =~ 1*VS.1 + 1*VS.2 + 1*VS.3 + 1*VS.4 + 1*VS.5
slp =~ 0*VS.1 + 1*VS.2 + 2*VS.3 + 3*VS.4 + 4*VS.5'

LCM_VSlin_fit = growth(LCM_VSlin, data=ABCD, missing='ML')
summary(LCM_VSlin_fit)
```

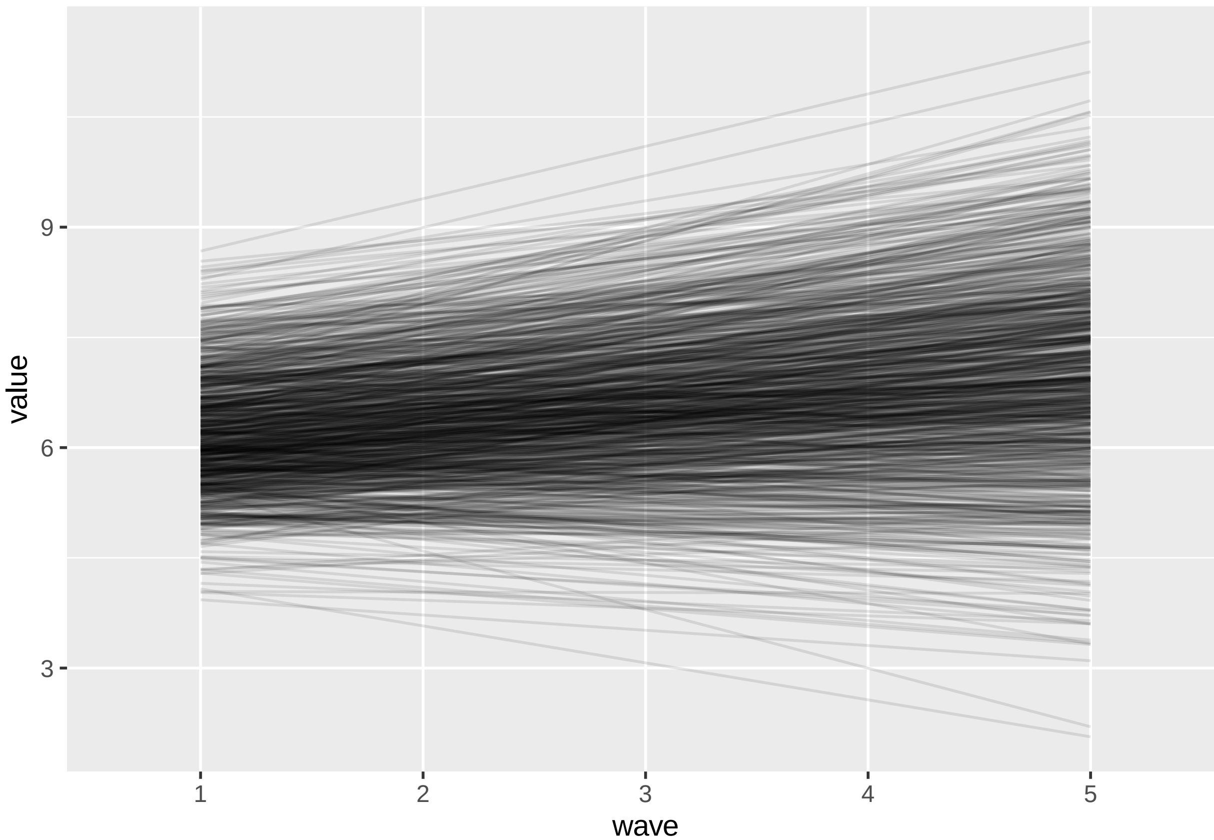
```
## lavaan 0.6-7 ended normally after 41 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         10
##
##   Number of observations                           964
##   Number of missing patterns                         8
##
## Model Test User Model:
##
##   Test statistic                                38.635
##   Degrees of freedom                                10
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
```

```
##   Information                                    Observed
##   Observed information based on                   Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int =~
##     VS.1              1.000
##     VS.2              1.000
##     VS.3              1.000
##     VS.4              1.000
##     VS.5              1.000
##   slp =~
##     VS.1              0.000
##     VS.2              1.000
##     VS.3              2.000
##     VS.4              3.000
##     VS.5              4.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int ~~
##     slp              -0.012    0.022   -0.538    0.591
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              0.000
##    .VS.2              0.000
##    .VS.3              0.000
##    .VS.4              0.000
##    .VS.5              0.000
##     int               6.198    0.042  148.400    0.000
##     slp               0.204    0.013   15.095    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              1.257    0.086   14.604    0.000
##    .VS.2              1.359    0.076   17.864    0.000
##    .VS.3              1.105    0.061   18.014    0.000
##    .VS.4              0.942    0.057   16.648    0.000
##    .VS.5              0.342    0.058    5.922    0.000
##     int               0.948    0.082   11.488    0.000
##     slp               0.087    0.010    9.025    0.000
```

```r
semPaths(LCM_VSlin_fit, intercepts=TRUE, edge.color='black', what = 'est')
```




```
pred_data <- as.data.frame(lavPredict(LCM_VSlin_fit, type = 'yhat')) %>%
  mutate(id = 1:n()) %>%
  pivot_longer(cols = 1:5) %>%
  extract(name, c('var', 'wave'), '(VS)\\.(\\d+)')

ggplot(pred_data) +
  geom_line(aes(x = wave, y = value, group = id), alpha = .1)
```



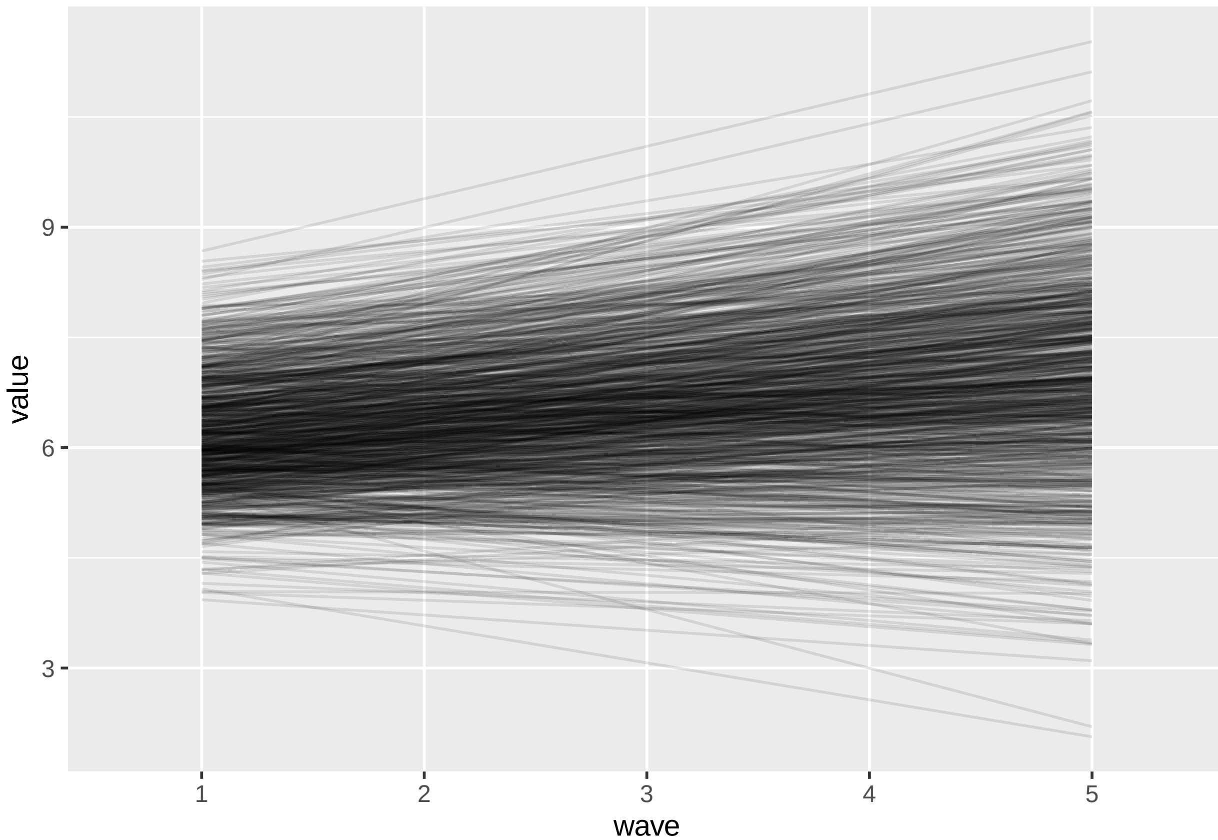
Alternative time coding so that intercept is the mean level at the last wave:

```
# Linear Slope: VS Activation (alternative time coding)
LCM_VSlin2 = '
int =~ 1*VS.1 + 1*VS.2 + 1*VS.3 + 1*VS.4 + 1*VS.5
slp =~ -4*VS.1 + -3*VS.2 + -2*VS.3 + -1*VS.4 + 0*VS.5'

LCM_VSlin2_fit = growth(LCM_VSlin2, data=ABCD, missing='ML')
summary(LCM_VSlin2_fit)
```

```
## lavaan 0.6-7 ended normally after 41 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         10
##
##   Number of observations                           964
##   Number of missing patterns                         8
##
## Model Test User Model:
##
##   Test statistic                                38.635
##   Degrees of freedom                                10
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
```

```
##   Information                                      Observed
##   Observed information based on                     Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int =~
##     VS.1              1.000
##     VS.2              1.000
##     VS.3              1.000
##     VS.4              1.000
##     VS.5              1.000
##   slp =~
##     VS.1             -4.000
##     VS.2             -3.000
##     VS.3             -2.000
##     VS.4             -1.000
##     VS.5              0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int ~~
##     slp               0.335    0.030   11.140    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              0.000
##    .VS.2              0.000
##    .VS.3              0.000
##    .VS.4              0.000
##    .VS.5              0.000
##     int               7.013    0.053  133.242    0.000
##     slp               0.204    0.013   15.095    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              1.257    0.086   14.604    0.000
##    .VS.2              1.359    0.076   17.864    0.000
##    .VS.3              1.105    0.061   18.014    0.000
##    .VS.4              0.942    0.057   16.648    0.000
##    .VS.5              0.342    0.058    5.922    0.000
##     int               2.240    0.129   17.385    0.000
##     slp               0.087    0.010    9.025    0.000
```

```r
pred_data <- as.data.frame(lavPredict(LCM_VSlin2_fit, type = 'yhat')) %>%
  mutate(id = 1:n()) %>%
  pivot_longer(cols = 1:5) %>%
  extract(name, c('var', 'wave'), '(VS)\\.(\\d+)')

ggplot(pred_data) +
  geom_line(aes(x = wave, y = value, group = id), alpha = .1)
```



Advanced latent growth models

Unconditional Quadratic Latent Growth Curve Model

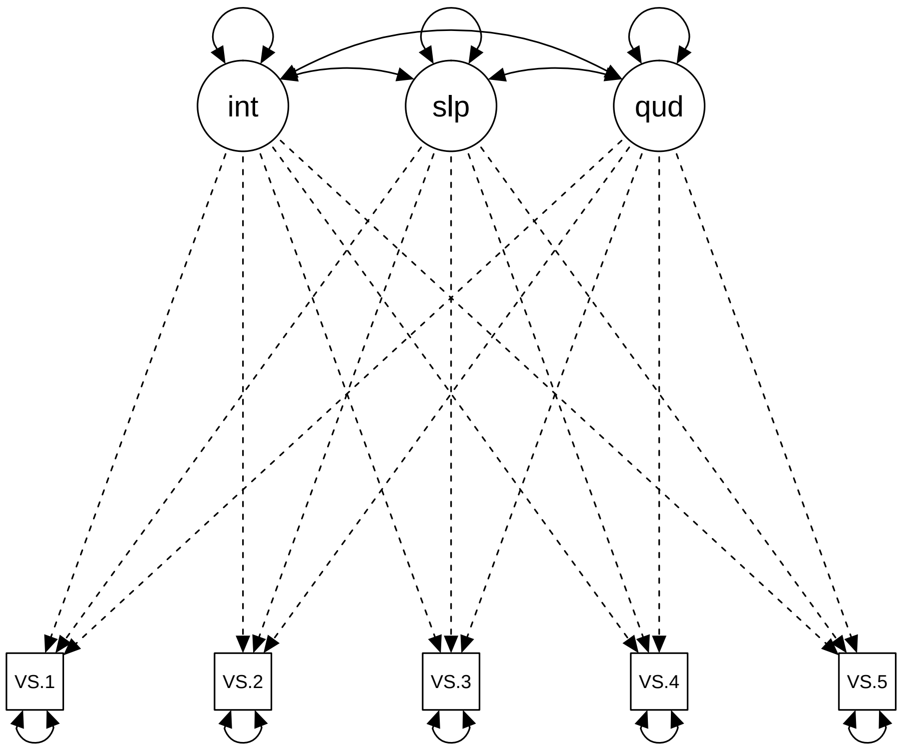
```
# Quadratic Effect: VS Activation
LCM.VSqud = 'int =~ 1*VS.1 + 1*VS.2 + 1*VS.3 + 1*VS.4 + 1*VS.5
             slp =~ 0*VS.1 + 1*VS.2 + 2*VS.3 + 3*VS.4 + 4*VS.5
             qud =~ 0*VS.1 + 1*VS.2 + 4*VS.3 + 9*VS.4 + 16*VS.5'

LCM.VSqud.fit = growth(LCM.VSqud, data=ABCD, missing='ML')
summary(LCM.VSqud.fit)
```

```
## lavaan 0.6-7 ended normally after 64 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         14
##
##   Number of observations                           964
##   Number of missing patterns                         8
##
## Model Test User Model:
##
##   Test statistic                                26.539
##   Degrees of freedom                                 6
##   P-value (Chi-square)                           0.000
```

```
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int =~
##     VS.1              1.000
##     VS.2              1.000
##     VS.3              1.000
##     VS.4              1.000
##     VS.5              1.000
##   slp =~
##     VS.1              0.000
##     VS.2              1.000
##     VS.3              2.000
##     VS.4              3.000
##     VS.5              4.000
##   qud =~
##     VS.1              0.000
##     VS.2              1.000
##     VS.3              4.000
##     VS.4              9.000
##     VS.5             16.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int ~~
##     slp               0.023    0.118    0.193    0.847
##     qud              -0.005    0.022   -0.218    0.827
##   slp ~~
##     qud               0.000    0.023    0.003    0.998
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              0.000
##    .VS.2              0.000
##    .VS.3              0.000
##    .VS.4              0.000
##    .VS.5              0.000
##     int               6.266    0.046  135.200    0.000
##     slp               0.081    0.039    2.097    0.036
##     qud               0.028    0.008    3.389    0.001
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              1.305    0.153    8.526    0.000
##    .VS.2              1.345    0.080   16.896    0.000
##    .VS.3              1.084    0.070   15.523    0.000
##    .VS.4              0.966    0.066   14.564    0.000
##    .VS.5              0.243    0.123    1.974    0.048
```

```
##     int                0.912    0.147    6.188    0.000
##     slp                0.063    0.113    0.558    0.577
##     qud                0.002    0.005    0.292    0.770
```

```
semPaths(LCM.VSqud.fit, intercepts=FALSE, edge.color='black')
```




```
# Quadratic Effect: Externalizing Behavior
LCM.EXTqud = 'int =~ 1*EXT.1 + 1*EXT.2 + 1*EXT.3 + 1*EXT.4 + 1*EXT.5
              slp =~ 0*EXT.1 + 1*EXT.2 + 2*EXT.3 + 3*EXT.4 + 4*EXT.5
              qud =~ 0*EXT.1 + 1*EXT.2 + 4*EXT.3 + 9*EXT.4 + 16*EXT.5'

LCM.EXTqud.fit = growth(LCM.EXTqud, data=ABCD, missing='ML')
```

```
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING: some cases
##   649 678
```

```
## Warning in lav_object_post_check(object): lavaan WARNING: some estimated lv
## variances are negative
```

```
summary(LCM.EXTqud.fit)
```

```
## lavaan 0.6-7 ended normally after 51 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         14
##
##                                                   Used       Total
##   Number of observations                           962         964
##   Number of missing patterns                        13
##
## Model Test User Model:
```

```
##
##   Test statistic                                12.606
##   Degrees of freedom                                 6
##   P-value (Chi-square)                           0.050
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int =~
##     EXT.1             1.000
##     EXT.2             1.000
##     EXT.3             1.000
##     EXT.4             1.000
##     EXT.5             1.000
##   slp =~
##     EXT.1             0.000
##     EXT.2             1.000
##     EXT.3             2.000
##     EXT.4             3.000
##     EXT.5             4.000
##   qud =~
##     EXT.1             0.000
##     EXT.2             1.000
##     EXT.3             4.000
##     EXT.4             9.000
##     EXT.5            16.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int ~~
##     slp              -0.041    0.039   -1.050    0.294
##     qud               0.007    0.009    0.852    0.394
##   slp ~~
##     qud               0.005    0.009    0.595    0.552
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .EXT.1             0.000
##    .EXT.2             0.000
##    .EXT.3             0.000
##    .EXT.4             0.000
##    .EXT.5             0.000
##     int               2.347    0.028   85.281    0.000
##     slp               0.026    0.023    1.162    0.245
##     qud               0.012    0.006    1.959    0.050
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .EXT.1             0.286    0.041    6.914    0.000
```

```
##    .EXT.2             0.264    0.018   14.639    0.000
##    .EXT.3             0.374    0.022   17.100    0.000
##    .EXT.4             0.143    0.016    9.148    0.000
##    .EXT.5             0.887    0.066   13.472    0.000
##     int               0.454    0.047    9.648    0.000
##     slp               0.001    0.039    0.036    0.971
##     qud              -0.001    0.002   -0.533    0.594
```

```
LCM.EXTqud2 = 'int =~ 1*EXT.1 + 1*EXT.2 + 1*EXT.3 + 1*EXT.4 + 1*EXT.5
               slp =~ 0*EXT.1 + 1*EXT.2 + 2*EXT.3 + 3*EXT.4 + 4*EXT.5
               qud =~ 0*EXT.1 + 1*EXT.2 + 4*EXT.3 + 9*EXT.4 + 16*EXT.5

               qud ~~ 0*qud
               qud ~~ 0*int + 0*slp'

LCM.EXTqud2.fit = growth(LCM.EXTqud2, data=ABCD, missing='ML')
```

```
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING: some cases
##   649 678
```

```
summary(LCM.EXTqud2.fit)
```

```
## lavaan 0.6-7 ended normally after 36 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         11
##
##                                                   Used       Total
##   Number of observations                           962         964
##   Number of missing patterns                        13
##
## Model Test User Model:
##
##   Test statistic                                19.153
##   Degrees of freedom                                 9
##   P-value (Chi-square)                           0.024
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int =~
##     EXT.1             1.000
##     EXT.2             1.000
##     EXT.3             1.000
##     EXT.4             1.000
##     EXT.5             1.000
##   slp =~
##     EXT.1             0.000
##     EXT.2             1.000
```

```
##     EXT.3             2.000
##     EXT.4             3.000
##     EXT.5             4.000
##   qud =~
##     EXT.1             0.000
##     EXT.2             1.000
##     EXT.3             4.000
##     EXT.4             9.000
##     EXT.5            16.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int ~~
##     qud               0.000
##   slp ~~
##     qud               0.000
##   int ~~
##     slp              -0.017    0.009   -2.003    0.045
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .EXT.1             0.000
##    .EXT.2             0.000
##    .EXT.3             0.000
##    .EXT.4             0.000
##    .EXT.5             0.000
##     int               2.348    0.027   86.745    0.000
##     slp               0.023    0.023    0.998    0.318
##     qud               0.013    0.006    2.118    0.034
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     qud               0.000
##    .EXT.1             0.286    0.024   11.998    0.000
##    .EXT.2             0.257    0.016   15.689    0.000
##    .EXT.3             0.367    0.021   17.514    0.000
##    .EXT.4             0.142    0.015    9.509    0.000
##    .EXT.5             0.879    0.050   17.428    0.000
##     int               0.432    0.031   13.777    0.000
##     slp               0.024    0.003    7.069    0.000
```

Unconditional Piecewise Latent Growth Curve Model

```r
# Piecewise Linear Effect: VS Activation
LCM.VSpw = 'int =~ 1*VS.1 + 1*VS.2 + 1*VS.3 + 1*VS.4 + 1*VS.5
            pw1 =~ 0*VS.1 + 1*VS.2 + 2*VS.3 + 3*VS.4 + 3*VS.5
            pw2 =~ 0*VS.1 + 0*VS.2 + 0*VS.3 + 1*VS.4 + 2*VS.5'

LCM.VSpw.fit = growth(LCM.VSpw, data=ABCD, missing='ML')
```

```
## Warning in lav_object_post_check(object): lavaan WARNING: some estimated lv
## variances are negative
```

```r
summary(LCM.VSpw.fit)
```

```
## lavaan 0.6-7 ended normally after 52 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         14
##
##   Number of observations                           964
##   Number of missing patterns                         8
##
## Model Test User Model:
##
##   Test statistic                                38.154
##   Degrees of freedom                                 6
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int =~
##     VS.1              1.000
##     VS.2              1.000
##     VS.3              1.000
##     VS.4              1.000
##     VS.5              1.000
##   pw1 =~
##     VS.1              0.000
##     VS.2              1.000
##     VS.3              2.000
##     VS.4              3.000
##     VS.5              3.000
##   pw2 =~
##     VS.1              0.000
##     VS.2              0.000
##     VS.3              0.000
##     VS.4              1.000
##     VS.5              2.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int ~~
##     pw1               0.015    0.050    0.291    0.771
##     pw2              -0.025    0.044   -0.561    0.575
##   pw1 ~~
##     pw2               0.089    0.029    3.028    0.002
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
```

```
##    .VS.1              0.000
##    .VS.2              0.000
##    .VS.3              0.000
##    .VS.4              0.000
##    .VS.5              0.000
##     int               6.288    0.045  139.147    0.000
##     pw1               0.098    0.022    4.484    0.000
##     pw2               0.223    0.026    8.617    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              1.324    0.114   11.568    0.000
##    .VS.2              1.359    0.078   17.480    0.000
##    .VS.3              1.164    0.080   14.557    0.000
##    .VS.4              0.883    0.070   12.701    0.000
##    .VS.5              0.530    0.118    4.486    0.000
##     int               0.941    0.110    8.584    0.000
##     pw1               0.030    0.030    0.996    0.319
##     pw2              -0.061    0.065   -0.937    0.349
```

Unconditional Latent Basis Growth Curve Model

```
# Latent Basis Effect: VS Activation
LCM.VSfl = 'int =~ 1*VS.1 + 1*VS.2 + 1*VS.3 + 1*VS.4 + 1*VS.5
            fl =~ 0*VS.1 + 1*VS.2 + VS.3 + VS.4 + VS.5'

LCM.VSfl.fit = growth(LCM.VSfl, data=ABCD, missing='ML')
summary(LCM.VSfl.fit)
```

```
## lavaan 0.6-7 ended normally after 88 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         13
##
##   Number of observations                           964
##   Number of missing patterns                         8
##
## Model Test User Model:
##
##   Test statistic                                18.435
##   Degrees of freedom                                 7
##   P-value (Chi-square)                           0.010
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int =~
##     VS.1              1.000
```

```
##     VS.2              1.000
##     VS.3              1.000
##     VS.4              1.000
##     VS.5              1.000
##   fl =~
##     VS.1              0.000
##     VS.2              1.000
##     VS.3              1.514    0.210    7.194    0.000
##     VS.4              2.416    0.341    7.078    0.000
##     VS.5              3.948    0.617    6.398    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int ~~
##     fl               -0.007    0.022   -0.322    0.747
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              0.000
##    .VS.2              0.000
##    .VS.3              0.000
##    .VS.4              0.000
##    .VS.5              0.000
##     int               6.232    0.045  137.239    0.000
##     fl                0.206    0.039    5.336    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              1.277    0.084   15.185    0.000
##    .VS.2              1.375    0.077   17.926    0.000
##    .VS.3              1.111    0.062   17.955    0.000
##    .VS.4              1.015    0.061   16.514    0.000
##    .VS.5              0.102    0.101    1.001    0.317
##     int               0.942    0.076   12.462    0.000
##     fl                0.105    0.035    2.973    0.003
```

Conditional Latent Growth Curve Model: TICs

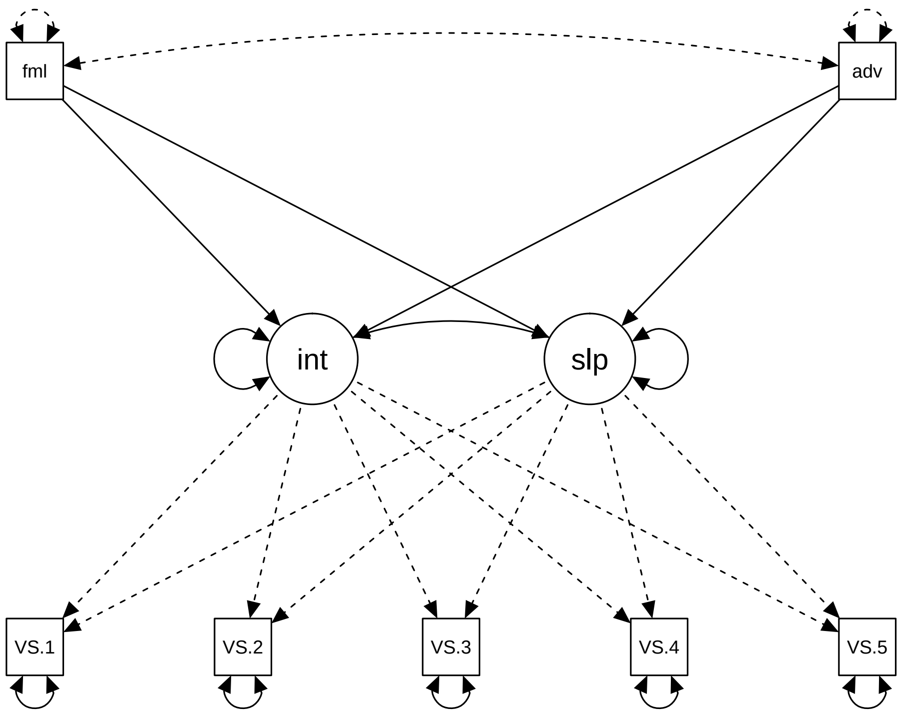
```
# Conditional TIC Model
LCM.TIC = 'int =~ 1*VS.1 + 1*VS.2 + 1*VS.3 + 1*VS.4 + 1*VS.5
           slp =~ 0*VS.1 + 1*VS.2 + 2*VS.3 + 3*VS.4 + 4*VS.5

           int ~ female + advers
           slp ~ female + advers'

LCM.TIC.fit = growth(LCM.TIC, data=ABCD, missing='ML')
summary(LCM.TIC.fit)
```

```
## lavaan 0.6-7 ended normally after 46 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         14
##
```

```
##   Number of observations                           964
##   Number of missing patterns                         8
##
## Model Test User Model:
##
##   Test statistic                                66.459
##   Degrees of freedom                                16
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int =~
##     VS.1              1.000
##     VS.2              1.000
##     VS.3              1.000
##     VS.4              1.000
##     VS.5              1.000
##   slp =~
##     VS.1              0.000
##     VS.2              1.000
##     VS.3              2.000
##     VS.4              3.000
##     VS.5              4.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int ~
##     female            0.015    0.086    0.176    0.860
##     advers            0.034    0.086    0.399    0.690
##   slp ~
##     female            0.050    0.028    1.799    0.072
##     advers            0.015    0.027    0.535    0.593
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##  .int ~~
##    .slp              -0.011    0.022   -0.513    0.608
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              0.000
##    .VS.2              0.000
##    .VS.3              0.000
##    .VS.4              0.000
##    .VS.5              0.000
##    .int               6.172    0.078   79.118    0.000
##    .slp               0.174    0.025    6.954    0.000
##
```

```
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1             1.258    0.086   14.581    0.000
##    .VS.2             1.361    0.076   17.847    0.000
##    .VS.3             1.107    0.061   18.000    0.000
##    .VS.4             0.940    0.057   16.612    0.000
##    .VS.5             0.344    0.058    5.950    0.000
##    .int              0.946    0.082   11.470    0.000
##    .slp              0.086    0.010    8.951    0.000
```

```r
semPaths(LCM.TIC.fit, intercepts=FALSE, edge.color='black')
```




```r
# Conditional TIC Model with Explicit Exogenous Covariance
LCM.TIC = 'int =~ 1*VS.1 + 1*VS.2 + 1*VS.3 + 1*VS.4 + 1*VS.5
           slp =~ 0*VS.1 + 1*VS.2 + 2*VS.3 + 3*VS.4 + 4*VS.5

           int ~ female + advers
           slp ~ female + advers

           female ~~ advers'

LCM.TIC.fit = growth(LCM.TIC, data=ABCD, missing='ML')
summary(LCM.TIC.fit)
```

```
## lavaan 0.6-7 ended normally after 45 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         17
##
##   Number of observations                           964
##   Number of missing patterns                         8
```

```
##
## Model Test User Model:
##
##   Test statistic                              1273.142
##   Degrees of freedom                                18
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int =~
##     VS.1              1.000
##     VS.2              1.000
##     VS.3              1.000
##     VS.4              1.000
##     VS.5              1.000
##   slp =~
##     VS.1              0.000
##     VS.2              1.000
##     VS.3              2.000
##     VS.4              3.000
##     VS.5              4.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int ~
##     female            0.015    0.086    0.176    0.860
##     advers            0.034    0.086    0.399    0.690
##   slp ~
##     female            0.050    0.028    1.799    0.072
##     advers            0.015    0.027    0.535    0.593
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   female ~~
##     advers            0.196    0.017   11.448    0.000
##  .int ~~
##    .slp              -0.011    0.022   -0.513    0.608
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              0.000
##    .VS.2              0.000
##    .VS.3              0.000
##    .VS.4              0.000
##    .VS.5              0.000
##     female            0.000
##     advers            0.000
##    .int               6.172    0.078   79.118    0.000
```

```
##    .slp              0.174    0.025    6.954    0.000
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1             1.258    0.086   14.581    0.000
##    .VS.2             1.361    0.076   17.847    0.000
##    .VS.3             1.107    0.061   18.000    0.000
##    .VS.4             0.940    0.057   16.612    0.000
##    .VS.5             0.344    0.058    5.950    0.000
##     female           0.446    0.020   21.954    0.000
##     advers           0.548    0.025   21.954    0.000
##    .int              0.946    0.082   11.470    0.000
##    .slp              0.086    0.010    8.951    0.000
```

Conditional Latent Growth Curve Model: TVCs

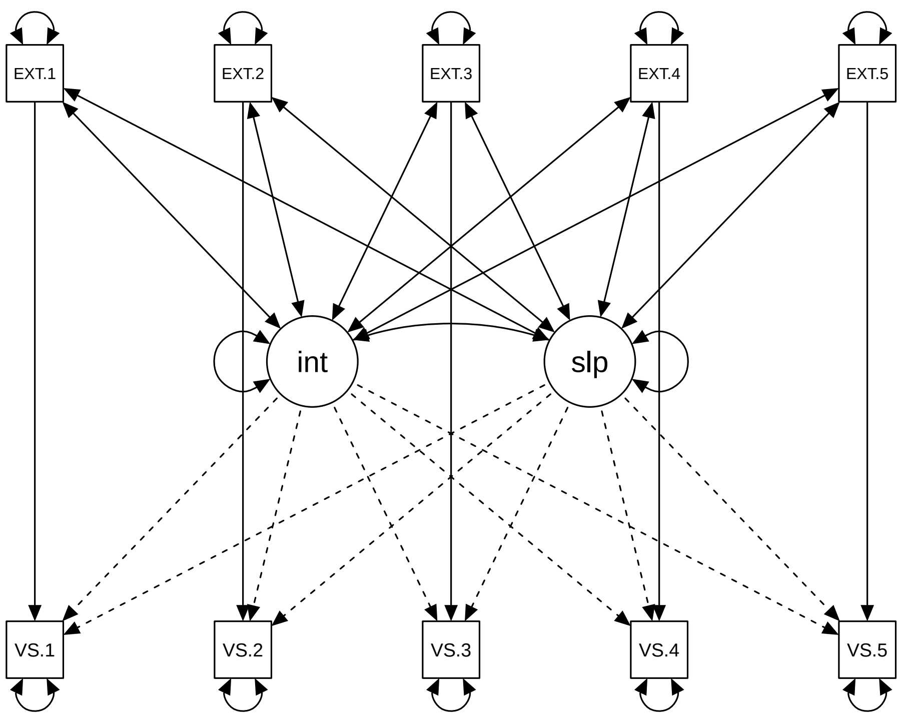
```r
# Conditional TVC Model
LCM.TVC1 = 'int =~ 1*VS.1 + 1*VS.2 + 1*VS.3 + 1*VS.4 + 1*VS.5
            slp =~ 0*VS.1 + 1*VS.2 + 2*VS.3 + 3*VS.4 + 4*VS.5

            VS.1 ~ EXT.1
            VS.2 ~ EXT.2
            VS.3 ~ EXT.3
            VS.4 ~ EXT.4
            VS.5 ~ EXT.5

            int ~~ EXT.1 + EXT.2 + EXT.3 + EXT.4 + EXT.5
            slp ~~ EXT.1 + EXT.2 + EXT.3 + EXT.4 + EXT.5'

LCM.TVC1.fit = growth(LCM.TVC1, data=ABCD, missing='ML')
summary(LCM.TVC1.fit)
```

```
## lavaan 0.6-7 ended normally after 162 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         30
##
##   Number of observations                           964
##   Number of missing patterns                        47
##
## Model Test User Model:
##
##   Test statistic                             11176.284
##   Degrees of freedom                                35
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
```

```
##                   Estimate  Std.Err  z-value  P(>|z|)
##   int =~
##     VS.1             1.000
##     VS.2             1.000
##     VS.3             1.000
##     VS.4             1.000
##     VS.5             1.000
##   slp =~
##     VS.1             0.000
##     VS.2             1.000
##     VS.3             2.000
##     VS.4             3.000
##     VS.5             4.000
##
## Regressions:
##                   Estimate  Std.Err  z-value  P(>|z|)
##   VS.1 ~
##     EXT.1            0.274    0.056    4.869    0.000
##   VS.2 ~
##     EXT.2            0.240    0.041    5.897    0.000
##   VS.3 ~
##     EXT.3            0.167    0.032    5.249    0.000
##   VS.4 ~
##     EXT.4            0.147    0.030    4.857    0.000
##   VS.5 ~
##     EXT.5            0.150    0.037    4.051    0.000
##
## Covariances:
##                   Estimate  Std.Err  z-value  P(>|z|)
##   int ~~
##     EXT.1           -2.051    0.421   -4.873    0.000
##     EXT.2            1.097    0.384    2.856    0.004
##     EXT.3           -0.200    0.374   -0.535    0.593
##     EXT.4           -1.209    0.425   -2.843    0.004
##     EXT.5            1.605    0.351    4.572    0.000
##   slp ~~
##     EXT.1            0.738    0.135    5.465    0.000
##     EXT.2            0.134    0.146    0.919    0.358
##     EXT.3            0.167    0.113    1.473    0.141
##     EXT.4            0.278    0.137    2.021    0.043
##     EXT.5           -0.702    0.143   -4.901    0.000
##   int ~~
##     slp             -0.400    0.100   -3.994    0.000
##
## Intercepts:
##                   Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1             0.000
##    .VS.2             0.000
##    .VS.3             0.000
##    .VS.4             0.000
##    .VS.5             0.000
##     EXT.1            0.000
##     EXT.2            0.000
##     EXT.3            0.000
```

```
##     EXT.4              0.000
##     EXT.5              0.000
##     int                5.980    0.133   45.107    0.000
##     slp               -0.008    0.044   -0.179    0.858
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              1.208    0.085   14.238    0.000
##    .VS.2              1.320    0.074   17.838    0.000
##    .VS.3              1.070    0.060   17.694    0.000
##    .VS.4              0.910    0.055   16.429    0.000
##    .VS.5              0.321    0.056    5.739    0.000
##     EXT.1             6.227    0.289   21.514    0.000
##     EXT.2             6.596    0.309   21.342    0.000
##     EXT.3             6.735    0.322   20.909    0.000
##     EXT.4             7.474    0.368   20.318    0.000
##     EXT.5             8.939    0.442   20.236    0.000
##     int               2.200    0.348    6.324    0.000
##     slp               0.240    0.038    6.240    0.000
```

```
semPaths(LCM.TVC1.fit, intercepts=FALSE, edge.color='black')
```



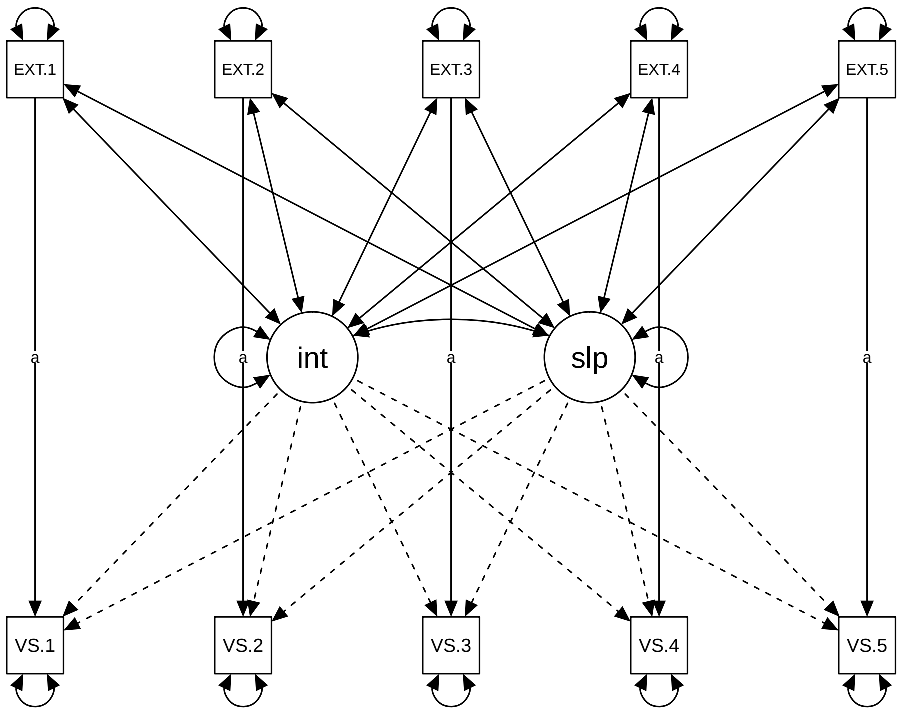

```
# Conditional TVC Model with Equality Constraints
LCM.TVC2 = 'int =~ 1*VS.1 + 1*VS.2 + 1*VS.3 + 1*VS.4 + 1*VS.5
            slp =~ 0*VS.1 + 1*VS.2 + 2*VS.3 + 3*VS.4 + 4*VS.5

            VS.1 ~ a*EXT.1
            VS.2 ~ a*EXT.2
            VS.3 ~ a*EXT.3
            VS.4 ~ a*EXT.4
            VS.5 ~ a*EXT.5
```

```r
           int ~~ EXT.1 + EXT.2 + EXT.3 + EXT.4 + EXT.5
           slp ~~ EXT.1 + EXT.2 + EXT.3 + EXT.4 + EXT.5'

LCM.TVC2.fit = growth(LCM.TVC2, data=ABCD, missing='ML')
summary(LCM.TVC2.fit)
```

```
## lavaan 0.6-7 ended normally after 144 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         30
##   Number of equality constraints                     4
##
##   Number of observations                           964
##   Number of missing patterns                        47
##
## Model Test User Model:
##
##   Test statistic                             11190.367
##   Degrees of freedom                                39
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int =~
##     VS.1              1.000
##     VS.2              1.000
##     VS.3              1.000
##     VS.4              1.000
##     VS.5              1.000
##   slp =~
##     VS.1              0.000
##     VS.2              1.000
##     VS.3              2.000
##     VS.4              3.000
##     VS.5              4.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   VS.1 ~
##     EXT.1      (a)    0.181    0.026    6.882    0.000
##   VS.2 ~
##     EXT.2      (a)    0.181    0.026    6.882    0.000
##   VS.3 ~
##     EXT.3      (a)    0.181    0.026    6.882    0.000
##   VS.4 ~
##     EXT.4      (a)    0.181    0.026    6.882    0.000
##   VS.5 ~
```

```
##     EXT.5      (a)    0.181    0.026    6.882    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int ~~
##     EXT.1            -1.700    0.369   -4.603    0.000
##     EXT.2             1.205    0.379    3.179    0.001
##     EXT.3            -0.244    0.373   -0.654    0.513
##     EXT.4            -1.186    0.429   -2.768    0.006
##     EXT.5             1.662    0.349    4.758    0.000
##   slp ~~
##     EXT.1             0.628    0.118    5.306    0.000
##     EXT.2             0.111    0.144    0.771    0.441
##     EXT.3             0.175    0.112    1.559    0.119
##     EXT.4             0.267    0.136    1.966    0.049
##     EXT.5            -0.776    0.127   -6.096    0.000
##   int ~~
##     slp              -0.347    0.089   -3.896    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              0.000
##    .VS.2              0.000
##    .VS.3              0.000
##    .VS.4              0.000
##    .VS.5              0.000
##     EXT.1             0.000
##     EXT.2             0.000
##     EXT.3             0.000
##     EXT.4             0.000
##     EXT.5             0.000
##     int               5.984    0.132   45.205    0.000
##     slp              -0.008    0.044   -0.190    0.849
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              1.206    0.085   14.227    0.000
##    .VS.2              1.326    0.074   17.875    0.000
##    .VS.3              1.076    0.061   17.683    0.000
##    .VS.4              0.909    0.056   16.368    0.000
##    .VS.5              0.333    0.056    5.898    0.000
##     EXT.1             6.230    0.290   21.500    0.000
##     EXT.2             6.596    0.309   21.336    0.000
##     EXT.3             6.734    0.322   20.922    0.000
##     EXT.4             7.460    0.366   20.356    0.000
##     EXT.5             8.938    0.441   20.247    0.000
##     int               2.043    0.312    6.543    0.000
##     slp               0.226    0.036    6.209    0.000
```

```r
semPaths(LCM.TVC2.fit, intercepts=FALSE, edge.color='black')
```



Multivariate Latent Curve Model

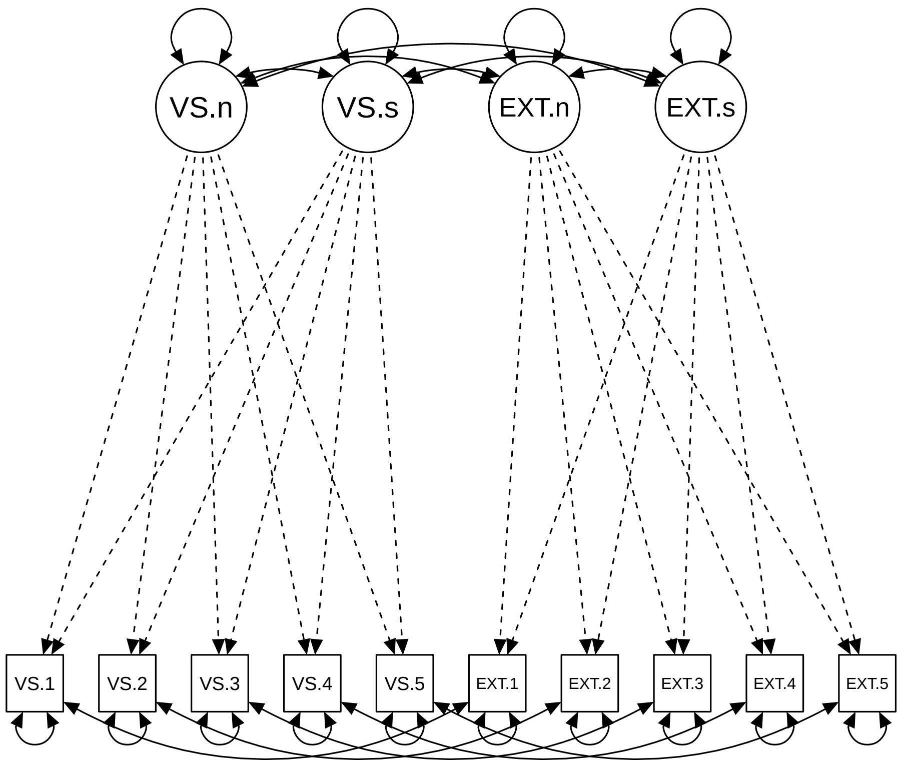
```
mLCM =      'VS.int =~ 1*VS.1 + 1*VS.2 + 1*VS.3 + 1*VS.4 + 1*VS.5
             VS.slp =~ 0*VS.1 + 1*VS.2 + 2*VS.3 + 3*VS.4 + 4*VS.5

             EXT.int =~ 1*EXT.1 + 1*EXT.2 + 1*EXT.3 + 1*EXT.4 + 1*EXT.5
             EXT.slp =~ 0*EXT.1 + 1*EXT.2 + 2*EXT.3 + 3*EXT.4 + 4*EXT.5

             VS.1 ~~ EXT.1
             VS.2 ~~ EXT.2
             VS.3 ~~ EXT.3
             VS.4 ~~ EXT.4
             VS.5 ~~ EXT.5

             VS.int ~~ VS.slp + EXT.int + EXT.slp
             VS.slp ~~ EXT.int + EXT.slp
             EXT.int ~~ EXT.slp'
mLCM.fit = growth(mLCM, data=ABCD, missing='ML')
summary(mLCM.fit)
```

```
## lavaan 0.6-7 ended normally after 63 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         29
##
##   Number of observations                           964
##   Number of missing patterns                        47
##
## Model Test User Model:
##
```

```
##   Test statistic                                97.489
##   Degrees of freedom                                36
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   VS.int =~
##     VS.1              1.000
##     VS.2              1.000
##     VS.3              1.000
##     VS.4              1.000
##     VS.5              1.000
##   VS.slp =~
##     VS.1              0.000
##     VS.2              1.000
##     VS.3              2.000
##     VS.4              3.000
##     VS.5              4.000
##   EXT.int =~
##     EXT.1             1.000
##     EXT.2             1.000
##     EXT.3             1.000
##     EXT.4             1.000
##     EXT.5             1.000
##   EXT.slp =~
##     EXT.1             0.000
##     EXT.2             1.000
##     EXT.3             2.000
##     EXT.4             3.000
##     EXT.5             4.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##  .VS.1 ~~
##    .EXT.1            -0.007    0.032   -0.216    0.829
##  .VS.2 ~~
##    .EXT.2             0.084    0.025    3.344    0.001
##  .VS.3 ~~
##    .EXT.3             0.123    0.027    4.571    0.000
##  .VS.4 ~~
##    .EXT.4             0.092    0.020    4.559    0.000
##  .VS.5 ~~
##    .EXT.5            -0.029    0.036   -0.806    0.420
##   VS.int ~~
##     VS.slp           -0.008    0.021   -0.394    0.694
##     EXT.int           0.019    0.036    0.542    0.588
##     EXT.slp           0.040    0.012    3.388    0.001
##   VS.slp ~~
```

```
##     EXT.int           0.059    0.012    5.061    0.000
##     EXT.slp          -0.010    0.004   -2.770    0.006
##   EXT.int ~~
##     EXT.slp          -0.016    0.008   -1.850    0.064
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              0.000
##    .VS.2              0.000
##    .VS.3              0.000
##    .VS.4              0.000
##    .VS.5              0.000
##    .EXT.1             0.000
##    .EXT.2             0.000
##    .EXT.3             0.000
##    .EXT.4             0.000
##    .EXT.5             0.000
##     VS.int            6.203    0.042  148.937    0.000
##     VS.slp            0.201    0.013   14.916    0.000
##     EXT.int           2.327    0.025   91.340    0.000
##     EXT.slp           0.071    0.008    8.608    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              1.274    0.087   14.716    0.000
##    .VS.2              1.353    0.076   17.885    0.000
##    .VS.3              1.112    0.062   18.060    0.000
##    .VS.4              0.957    0.057   16.796    0.000
##    .VS.5              0.316    0.057    5.597    0.000
##    .EXT.1             0.285    0.023   12.235    0.000
##    .EXT.2             0.256    0.016   15.725    0.000
##    .EXT.3             0.376    0.021   17.624    0.000
##    .EXT.4             0.141    0.015    9.496    0.000
##    .EXT.5             0.876    0.050   17.502    0.000
##     VS.int            0.937    0.082   11.434    0.000
##     VS.slp            0.087    0.010    9.102    0.000
##     EXT.int           0.431    0.031   13.814    0.000
##     EXT.slp           0.024    0.003    7.155    0.000
```

```
semPaths(mLCM.fit, intercepts=FALSE, edge.color='black')
```



Multivariate Latent Curve Model with Structured Residuals

```
mLCMSR =   '# Define the Latent Factors
            VS.int =~ 1*VS.1 + 1*VS.2 + 1*VS.3 + 1*VS.4 + 1*VS.5
            VS.slp =~ 0*VS.1 + 1*VS.2 + 2*VS.3 + 3*VS.4 + 4*VS.5

            EXT.int =~ 1*EXT.1 + 1*EXT.2 + 1*EXT.3 + 1*EXT.4 + 1*EXT.5
            EXT.slp =~ 0*EXT.1 + 1*EXT.2 + 2*EXT.3 + 3*EXT.4 + 4*EXT.5

            # Factor Covariances
            VS.int ~~ VS.slp + EXT.int + EXT.slp
            VS.slp ~~ EXT.int + EXT.slp
            EXT.int ~~ EXT.slp

            # Define Phantom Variables
            VS.1 ~~ 0*VS.1; srVS.1 =~ 1*VS.1; srVS.1 ~ 0; srVS.1 ~~ srVS.1
            VS.2 ~~ 0*VS.2; srVS.2 =~ 1*VS.2; srVS.2 ~ 0; srVS.2 ~~ srVS.2
            VS.3 ~~ 0*VS.3; srVS.3 =~ 1*VS.3; srVS.3 ~ 0; srVS.3 ~~ srVS.3
            VS.4 ~~ 0*VS.4; srVS.4 =~ 1*VS.4; srVS.4 ~ 0; srVS.4 ~~ srVS.4
            VS.5 ~~ 0*VS.5; srVS.5 =~ 1*VS.5; srVS.5 ~ 0; srVS.5 ~~ srVS.5

            EXT.1 ~~ 0*EXT.1; srEXT.1 =~ 1*EXT.1; srEXT.1 ~ 0; srEXT.1 ~~ srEXT.1
            EXT.2 ~~ 0*EXT.2; srEXT.2 =~ 1*EXT.2; srEXT.2 ~ 0; srEXT.2 ~~ srEXT.2
            EXT.3 ~~ 0*EXT.3; srEXT.3 =~ 1*EXT.3; srEXT.3 ~ 0; srEXT.3 ~~ srEXT.3
            EXT.4 ~~ 0*EXT.4; srEXT.4 =~ 1*EXT.4; srEXT.4 ~ 0; srEXT.4 ~~ srEXT.4
            EXT.5 ~~ 0*EXT.5; srEXT.5 =~ 1*EXT.5; srEXT.5 ~ 0; srEXT.5 ~~ srEXT.5

            # Structured Residuals Regressions + Covariances
            srVS.2 ~ a*srVS.1 + b*srEXT.1
```

```r
            srVS.3 ~ a*srVS.2 + b*srEXT.2
            srVS.4 ~ a*srVS.3 + b*srEXT.3
            srVS.5 ~ a*srVS.4 + b*srEXT.4

            srEXT.2 ~ c*srEXT.1 + d*srVS.1
            srEXT.3 ~ c*srEXT.2 + d*srVS.2
            srEXT.4 ~ c*srEXT.3 + d*srVS.3
            srEXT.5 ~ c*srEXT.4 + d*srVS.4

            srVS.1 ~~ srEXT.1
            srVS.2 ~~ srEXT.2
            srVS.3 ~~ srEXT.3
            srVS.4 ~~ srEXT.4
            srVS.5 ~~ srEXT.5

            # Uncouple 1st SRs from Growth Factors
            VS.int ~~ 0*srVS.1 + 0*srEXT.1
            VS.slp ~~ 0*srVS.1 + 0*srEXT.1
            EXT.int ~~ 0*srVS.1 + 0*srEXT.1
            EXT.slp ~~ 0*srVS.1 + 0*srEXT.1
'
mLCMSR.fit = growth(mLCMSR, data=ABCD, missing='ML')
summary(mLCMSR.fit)
```

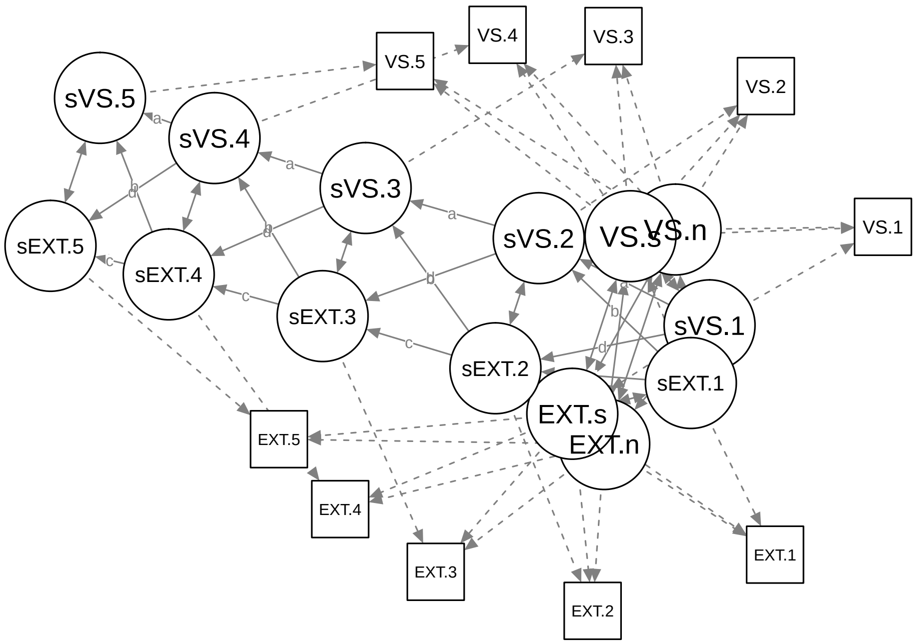
```
## lavaan 0.6-7 ended normally after 84 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         45
##   Number of equality constraints                    12
##
##   Number of observations                           964
##   Number of missing patterns                        47
##
## Model Test User Model:
##
##   Test statistic                                87.795
##   Degrees of freedom                                32
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   VS.int =~
##     VS.1              1.000
##     VS.2              1.000
##     VS.3              1.000
##     VS.4              1.000
##     VS.5              1.000
```

```
##   VS.slp =~
##     VS.1              0.000
##     VS.2              1.000
##     VS.3              2.000
##     VS.4              3.000
##     VS.5              4.000
##   EXT.int =~
##     EXT.1             1.000
##     EXT.2             1.000
##     EXT.3             1.000
##     EXT.4             1.000
##     EXT.5             1.000
##   EXT.slp =~
##     EXT.1             0.000
##     EXT.2             1.000
##     EXT.3             2.000
##     EXT.4             3.000
##     EXT.5             4.000
##   srVS.1 =~
##     VS.1              1.000
##   srVS.2 =~
##     VS.2              1.000
##   srVS.3 =~
##     VS.3              1.000
##   srVS.4 =~
##     VS.4              1.000
##   srVS.5 =~
##     VS.5              1.000
##   srEXT.1 =~
##     EXT.1             1.000
##   srEXT.2 =~
##     EXT.2             1.000
##   srEXT.3 =~
##     EXT.3             1.000
##   srEXT.4 =~
##     EXT.4             1.000
##   srEXT.5 =~
##     EXT.5             1.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   srVS.2 ~
##     srVS.1     (a)    0.032    0.032    1.012    0.312
##     srEXT.1    (b)    0.109    0.054    2.000    0.045
##   srVS.3 ~
##     srVS.2     (a)    0.032    0.032    1.012    0.312
##     srEXT.2    (b)    0.109    0.054    2.000    0.045
##   srVS.4 ~
##     srVS.3     (a)    0.032    0.032    1.012    0.312
##     srEXT.3    (b)    0.109    0.054    2.000    0.045
##   srVS.5 ~
##     srVS.4     (a)    0.032    0.032    1.012    0.312
##     srEXT.4    (b)    0.109    0.054    2.000    0.045
##   srEXT.2 ~
```

```
##     srEXT.1    (c)   -0.058    0.034   -1.694    0.090
##     srVS.1     (d)    0.019    0.013    1.431    0.152
##   srEXT.3 ~
##     srEXT.2    (c)   -0.058    0.034   -1.694    0.090
##     srVS.2     (d)    0.019    0.013    1.431    0.152
##   srEXT.4 ~
##     srEXT.3    (c)   -0.058    0.034   -1.694    0.090
##     srVS.3     (d)    0.019    0.013    1.431    0.152
##   srEXT.5 ~
##     srEXT.4    (c)   -0.058    0.034   -1.694    0.090
##     srVS.4     (d)    0.019    0.013    1.431    0.152
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   VS.int ~~
##     VS.slp            0.008    0.026    0.297    0.767
##     EXT.int          -0.007    0.039   -0.174    0.862
##     EXT.slp           0.048    0.013    3.781    0.000
##   VS.slp ~~
##     EXT.int           0.066    0.012    5.390    0.000
##     EXT.slp          -0.014    0.004   -3.360    0.001
##   EXT.int ~~
##     EXT.slp          -0.022    0.009   -2.505    0.012
##   srVS.1 ~~
##     srEXT.1           0.011    0.034    0.322    0.748
##  .srVS.2 ~~
##    .srEXT.2           0.108    0.028    3.892    0.000
##  .srVS.3 ~~
##    .srEXT.3           0.135    0.027    4.927    0.000
##  .srVS.4 ~~
##    .srEXT.4           0.117    0.023    5.045    0.000
##  .srVS.5 ~~
##    .srEXT.5          -0.020    0.036   -0.564    0.573
##   VS.int ~~
##     srVS.1            0.000
##     srEXT.1           0.000
##   VS.slp ~~
##     srVS.1            0.000
##     srEXT.1           0.000
##   EXT.int ~~
##     srVS.1            0.000
##     srEXT.1           0.000
##   EXT.slp ~~
##     srVS.1            0.000
##     srEXT.1           0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     srVS.1            0.000
##    .srVS.2            0.000
##    .srVS.3            0.000
##    .srVS.4            0.000
##    .srVS.5            0.000
##     srEXT.1           0.000
```

```
##    .srEXT.2          0.000
##    .srEXT.3          0.000
##    .srEXT.4          0.000
##    .srEXT.5          0.000
##    .VS.1             0.000
##    .VS.2             0.000
##    .VS.3             0.000
##    .VS.4             0.000
##    .VS.5             0.000
##    .EXT.1            0.000
##    .EXT.2            0.000
##    .EXT.3            0.000
##    .EXT.4            0.000
##    .EXT.5            0.000
##     VS.int           6.202    0.042  149.172    0.000
##     VS.slp           0.201    0.013   14.962    0.000
##     EXT.int          2.328    0.025   91.319    0.000
##     EXT.slp          0.071    0.008    8.526    0.000
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1             0.000
##     srVS.1           1.329    0.099   13.382    0.000
##    .VS.2             0.000
##    .srVS.2           1.388    0.085   16.329    0.000
##    .VS.3             0.000
##    .srVS.3           1.119    0.062   17.969    0.000
##    .VS.4             0.000
##    .srVS.4           1.003    0.071   14.030    0.000
##    .VS.5             0.000
##    .srVS.5           0.346    0.061    5.623    0.000
##    .EXT.1            0.000
##     srEXT.1          0.266    0.023   11.479    0.000
##    .EXT.2            0.000
##    .srEXT.2          0.245    0.017   14.256    0.000
##    .EXT.3            0.000
##    .srEXT.3          0.357    0.023   15.463    0.000
##    .EXT.4            0.000
##    .srEXT.4          0.129    0.017    7.801    0.000
##    .EXT.5            0.000
##    .srEXT.5          0.874    0.050   17.467    0.000
##     VS.int           0.878    0.097    9.030    0.000
##     VS.slp           0.081    0.011    7.278    0.000
##     EXT.int          0.452    0.032   13.958    0.000
##     EXT.slp          0.028    0.004    7.312    0.000
```

```r
semPaths(mLCMSR.fit, layout='spring', intercepts = F, residuals = F)
```



Two-Timepoint Latent Change Score

```
# FIML 2TP LCS Model
LCS1 = '# Set Regression Path to 1
        VS.2 ~ 1*VS.1

        # Define Change Latent Variable
        dVS.21 =~ 1*VS.2
        dVS.21 ~ 1

        # Estimate Intercept and Variance of V.1
        VS.1 ~ 1
        VS.1 ~~ VS.1

        # Constraint Intercept and Variance of V.2 to 0
        VS.2 ~ 0
        VS.2 ~~ 0*VS.2
'
LCS1.fit = sem(LCS1, data=ABCD, missing='ML')
summary(LCS1.fit)
```

```
## lavaan 0.6-7 ended normally after 15 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                          4
##
##   Number of observations                           964
##   Number of missing patterns                         2
##
## Model Test User Model:
##
##   Test statistic                               240.497
```

```
##   Degrees of freedom                                 1
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   dVS.21 =~
##     VS.2              1.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   VS.2 ~
##     VS.1              1.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     dVS.21            0.188    0.054    3.460    0.001
##     VS.1              6.239    0.047  132.084    0.000
##    .VS.2              0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     VS.1              2.151    0.098   21.954    0.000
##    .VS.2              0.000
##     dVS.21            2.708    0.126   21.413    0.000
```

```r
# Complete Case 2TP LCS Model
LCS1.fit2 = sem(LCS1, data=ABCD, missing='listwise')
summary(LCS1.fit2)
```

```
## lavaan 0.6-7 ended normally after 13 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                          4
##
##                                                   Used       Total
##   Number of observations                           917         964
##
## Model Test User Model:
##
##   Test statistic                               240.497
##   Degrees of freedom                                 1
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
```

```
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   dVS.21 =~
##     VS.2              1.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   VS.2 ~
##     VS.1              1.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     dVS.21            0.188    0.054    3.460    0.001
##     VS.1              6.248    0.048  130.781    0.000
##    .VS.2              0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     VS.1              2.093    0.098   21.413    0.000
##    .VS.2              0.000
##     dVS.21            2.708    0.126   21.413    0.000
```

```
# Paired Samples T-Test
t.test(ABCD$VS.1, ABCD$VS.2, paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  ABCD$VS.1 and ABCD$VS.2
## t = -3.4577, df = 916, p-value = 0.00057
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.29468244 -0.08128474
## sample estimates:
## mean of the differences
##              -0.1879836
```

```
describe(ABCD[,c('VS.1','VS.2')])
```

```
##      vars   n mean   sd median trimmed  mad  min max range skew kurtosis   se
## VS.1    1 964 6.24 1.47   6.19    6.22 1.41 1.43  10  8.57 0.09     0.09 0.05
## VS.2    2 917 6.44 1.59   6.67    6.41 1.41 1.90  10  8.10 0.10    -0.24 0.05
```

```
6.248 + .188
```

```
## [1] 6.436
```

```
# Proportional Change LCS
LCS2 = '# Set Regression Path to 1
        VS.2 ~ 1*VS.1

        # Define Change Latent Variable
        dVS.21 =~ 1*VS.2
        dVS.21 ~ 1

        # Regress Change on Initial Status
```

```
        dVS.21 ~ VS.1

        # Estimate Intercept and Variance of V.1
        VS.1 ~ 1
        VS.1 ~~ VS.1

        # Constraint Intercept and Variance of V.2 to 0
        VS.2 ~ 0
        VS.2 ~~ 0*VS.2
'
LCS2.fit = sem(LCS2, data=ABCD, missing='ML')
summary(LCS2.fit)
```

```
## lavaan 0.6-7 ended normally after 29 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                          5
##
##   Number of observations                           964
##   Number of missing patterns                         2
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 0
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   dVS.21 =~
##     VS.2              1.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   VS.2 ~
##     VS.1              1.000
##   dVS.21 ~
##     VS.1             -0.546    0.033  -16.583    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .dVS.21            3.601    0.211   17.045    0.000
##     VS.1              6.239    0.047  132.084    0.000
##    .VS.2              0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     VS.1              2.151    0.098   21.954    0.000
```

```
##    .VS.2              0.000
##    .dVS.21            2.083    0.097   21.413    0.000
```

Latent Change Score Trajectory Model

```
# LCS Trajectory Model
LCSt = '# Define Phantom Variables
        pVS.1 =~ 1*VS.1; VS.1 ~ 0; VS.1 ~~ VS.1; pVS.1 ~~ 0*pVS.1
        pVS.2 =~ 1*VS.2; VS.2 ~ 0; VS.2 ~~ VS.2; pVS.2 ~~ 0*pVS.2
        pVS.3 =~ 1*VS.3; VS.3 ~ 0; VS.3 ~~ VS.3; pVS.3 ~~ 0*pVS.3
        pVS.4 =~ 1*VS.4; VS.4 ~ 0; VS.4 ~~ VS.4; pVS.4 ~~ 0*pVS.4
        pVS.5 =~ 1*VS.5; VS.5 ~ 0; VS.5 ~~ VS.5; pVS.5 ~~ 0*pVS.5

        # Regressions Between Adjacent Observations
        pVS.2 ~ 1*pVS.1
        pVS.3 ~ 1*pVS.2
        pVS.4 ~ 1*pVS.3
        pVS.5 ~ 1*pVS.4

        # Define Change Latent Variables
        dVS.21 =~ 1*pVS.2; dVS.21 ~~ 0*dVS.21
        dVS.32 =~ 1*pVS.3; dVS.32 ~~ 0*dVS.32
        dVS.43 =~ 1*pVS.4; dVS.43 ~~ 0*dVS.43
        dVS.54 =~ 1*pVS.5; dVS.54 ~~ 0*dVS.54

        # Define Intercept and Slope
        int =~ 1*pVS.1
        slp =~ 1*dVS.21 + 1*dVS.32 + 1*dVS.43 + 1*dVS.54

        int ~ 1
        slp ~ 1
        int ~~ int + slp
        slp ~~ slp
'
LCSt.fit = sem(LCSt, data=ABCD, missing='ML')
summary(LCSt.fit)
```

```
## lavaan 0.6-7 ended normally after 41 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         10
##
##   Number of observations                           964
##   Number of missing patterns                         8
##
## Model Test User Model:
##
##   Test statistic                                38.635
##   Degrees of freedom                                10
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
```

```
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   pVS.1 =~
##     VS.1              1.000
##   pVS.2 =~
##     VS.2              1.000
##   pVS.3 =~
##     VS.3              1.000
##   pVS.4 =~
##     VS.4              1.000
##   pVS.5 =~
##     VS.5              1.000
##   dVS.21 =~
##     pVS.2             1.000
##   dVS.32 =~
##     pVS.3             1.000
##   dVS.43 =~
##     pVS.4             1.000
##   dVS.54 =~
##     pVS.5             1.000
##   int =~
##     pVS.1             1.000
##   slp =~
##     dVS.21            1.000
##     dVS.32            1.000
##     dVS.43            1.000
##     dVS.54            1.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   pVS.2 ~
##     pVS.1             1.000
##   pVS.3 ~
##     pVS.2             1.000
##   pVS.4 ~
##     pVS.3             1.000
##   pVS.5 ~
##     pVS.4             1.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int ~~
##     slp              -0.012    0.022   -0.538    0.591
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              0.000
##    .VS.2              0.000
##    .VS.3              0.000
##    .VS.4              0.000
```

```
##    .VS.5              0.000
##     int               6.198    0.042  148.400    0.000
##     slp               0.204    0.013   15.095    0.000
##    .pVS.1             0.000
##    .pVS.2             0.000
##    .pVS.3             0.000
##    .pVS.4             0.000
##    .pVS.5             0.000
##    .dVS.21            0.000
##    .dVS.32            0.000
##    .dVS.43            0.000
##    .dVS.54            0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              1.257    0.086   14.604    0.000
##    .pVS.1             0.000
##    .VS.2              1.359    0.076   17.864    0.000
##    .pVS.2             0.000
##    .VS.3              1.105    0.061   18.014    0.000
##    .pVS.3             0.000
##    .VS.4              0.942    0.057   16.648    0.000
##    .pVS.4             0.000
##    .VS.5              0.342    0.058    5.922    0.000
##    .pVS.5             0.000
##    .dVS.21            0.000
##    .dVS.32            0.000
##    .dVS.43            0.000
##    .dVS.54            0.000
##     int               0.948    0.082   11.488    0.000
##     slp               0.087    0.010    9.025    0.000
```

```
# Identical LCM Model
summary(LCM_VSlin_fit)
```

```
## lavaan 0.6-7 ended normally after 41 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         10
##
##   Number of observations                           964
##   Number of missing patterns                         8
##
## Model Test User Model:
##
##   Test statistic                                38.635
##   Degrees of freedom                                10
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
```

```
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int =~
##     VS.1              1.000
##     VS.2              1.000
##     VS.3              1.000
##     VS.4              1.000
##     VS.5              1.000
##   slp =~
##     VS.1              0.000
##     VS.2              1.000
##     VS.3              2.000
##     VS.4              3.000
##     VS.5              4.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int ~~
##     slp              -0.012    0.022   -0.538    0.591
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              0.000
##    .VS.2              0.000
##    .VS.3              0.000
##    .VS.4              0.000
##    .VS.5              0.000
##     int               6.198    0.042  148.400    0.000
##     slp               0.204    0.013   15.095    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              1.257    0.086   14.604    0.000
##    .VS.2              1.359    0.076   17.864    0.000
##    .VS.3              1.105    0.061   18.014    0.000
##    .VS.4              0.942    0.057   16.648    0.000
##    .VS.5              0.342    0.058    5.922    0.000
##     int               0.948    0.082   11.488    0.000
##     slp               0.087    0.010    9.025    0.000
```

```
# Proportional Change LCS Trajectory Model
LCSpt = '# Define Phantom Variables
        pVS.1 =~ 1*VS.1; VS.1 ~ 0; VS.1 ~~ VS.1; pVS.1 ~~ 0*pVS.1
        pVS.2 =~ 1*VS.2; VS.2 ~ 0; VS.2 ~~ VS.2; pVS.2 ~~ 0*pVS.2
        pVS.3 =~ 1*VS.3; VS.3 ~ 0; VS.3 ~~ VS.3; pVS.3 ~~ 0*pVS.3
        pVS.4 =~ 1*VS.4; VS.4 ~ 0; VS.4 ~~ VS.4; pVS.4 ~~ 0*pVS.4
        pVS.5 =~ 1*VS.5; VS.5 ~ 0; VS.5 ~~ VS.5; pVS.5 ~~ 0*pVS.5

        # Regressions Between Adjacent Observations
        pVS.2 ~ 1*pVS.1
        pVS.3 ~ 1*pVS.2
        pVS.4 ~ 1*pVS.3
        pVS.5 ~ 1*pVS.4
```

```r
        # Define Change Latent Variables
        dVS.21 =~ 1*pVS.2; dVS.21 ~~ 0*dVS.21
        dVS.32 =~ 1*pVS.3; dVS.32 ~~ 0*dVS.32
        dVS.43 =~ 1*pVS.4; dVS.43 ~~ 0*dVS.43
        dVS.54 =~ 1*pVS.5; dVS.54 ~~ 0*dVS.54

        # Define Proportional Regressions
        dVS.21 ~ beta*pVS.1
        dVS.32 ~ beta*pVS.2
        dVS.43 ~ beta*pVS.3
        dVS.54 ~ beta*pVS.4

        # Define Intercept and Slope
        int =~ 1*pVS.1
        slp =~ 1*dVS.21 + 1*dVS.32 + 1*dVS.43 + 1*dVS.54

        int ~ 1
        slp ~ 1
        int ~~ int + slp
        slp ~~ slp
'
LCSpt.fit = sem(LCSpt, data=ABCD, missing='ML')
summary(LCSpt.fit)
```

```
## lavaan 0.6-7 ended normally after 72 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         14
##   Number of equality constraints                     3
##
##   Number of observations                           964
##   Number of missing patterns                         8
##
## Model Test User Model:
##
##   Test statistic                                26.912
##   Degrees of freedom                                 9
##   P-value (Chi-square)                           0.001
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   pVS.1 =~
##     VS.1              1.000
##   pVS.2 =~
##     VS.2              1.000
##   pVS.3 =~
```

```
##     VS.3              1.000
##   pVS.4 =~
##     VS.4              1.000
##   pVS.5 =~
##     VS.5              1.000
##   dVS.21 =~
##     pVS.2             1.000
##   dVS.32 =~
##     pVS.3             1.000
##   dVS.43 =~
##     pVS.4             1.000
##   dVS.54 =~
##     pVS.5             1.000
##   int =~
##     pVS.1             1.000
##   slp =~
##     dVS.21            1.000
##     dVS.32            1.000
##     dVS.43            1.000
##     dVS.54            1.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   pVS.2 ~
##     pVS.1             1.000
##   pVS.3 ~
##     pVS.2             1.000
##   pVS.4 ~
##     pVS.3             1.000
##   pVS.5 ~
##     pVS.4             1.000
##   dVS.21 ~
##     pVS.1   (beta)    0.298    0.105    2.852    0.004
##   dVS.32 ~
##     pVS.2   (beta)    0.298    0.105    2.852    0.004
##   dVS.43 ~
##     pVS.3   (beta)    0.298    0.105    2.852    0.004
##   dVS.54 ~
##     pVS.4   (beta)    0.298    0.105    2.852    0.004
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   int ~~
##     slp              -0.278    0.102   -2.716    0.007
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              0.000
##    .VS.2              0.000
##    .VS.3              0.000
##    .VS.4              0.000
##    .VS.5              0.000
##     int               6.256    0.045  139.992    0.000
##     slp              -1.739    0.682   -2.551    0.011
```

```
##    .pVS.1             0.000
##    .pVS.2             0.000
##    .pVS.3             0.000
##    .pVS.4             0.000
##    .pVS.5             0.000
##    .dVS.21            0.000
##    .dVS.32            0.000
##    .dVS.43            0.000
##    .dVS.54            0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              1.316    0.084   15.693    0.000
##    .pVS.1             0.000
##    .VS.2              1.348    0.077   17.558    0.000
##    .pVS.2             0.000
##    .VS.3              1.104    0.061   18.056    0.000
##    .pVS.3             0.000
##    .VS.4              0.999    0.062   16.185    0.000
##    .pVS.4             0.000
##    .VS.5              0.163    0.091    1.788    0.074
##    .pVS.5             0.000
##    .dVS.21            0.000
##    .dVS.32            0.000
##    .dVS.43            0.000
##    .dVS.54            0.000
##     int               0.949    0.074   12.796    0.000
##     slp               0.119    0.049    2.452    0.014
```

Bivariate Latent Change Score Model

```
# Bivariate Within-Construct Proportional LCS Trajectory Model
bLCSt = '# Define Phantom Variables
         pVS.1 =~ 1*VS.1; VS.1 ~ 0; VS.1 ~~ VS.1; pVS.1 ~~ 0*pVS.1
         pVS.2 =~ 1*VS.2; VS.2 ~ 0; VS.2 ~~ VS.2; pVS.2 ~~ 0*pVS.2
         pVS.3 =~ 1*VS.3; VS.3 ~ 0; VS.3 ~~ VS.3; pVS.3 ~~ 0*pVS.3
         pVS.4 =~ 1*VS.4; VS.4 ~ 0; VS.4 ~~ VS.4; pVS.4 ~~ 0*pVS.4
         pVS.5 =~ 1*VS.5; VS.5 ~ 0; VS.5 ~~ VS.5; pVS.5 ~~ 0*pVS.5

         pEXT.1 =~ 1*EXT.1; EXT.1 ~ 0; EXT.1 ~~ EXT.1; pEXT.1 ~~ 0*pEXT.1
         pEXT.2 =~ 1*EXT.2; EXT.2 ~ 0; EXT.2 ~~ EXT.2; pEXT.2 ~~ 0*pEXT.2
         pEXT.3 =~ 1*EXT.3; EXT.3 ~ 0; EXT.3 ~~ EXT.3; pEXT.3 ~~ 0*pEXT.3
         pEXT.4 =~ 1*EXT.4; EXT.4 ~ 0; EXT.4 ~~ EXT.4; pEXT.4 ~~ 0*pEXT.4
         pEXT.5 =~ 1*EXT.5; EXT.5 ~ 0; EXT.5 ~~ EXT.5; pEXT.5 ~~ 0*pEXT.5

         # Residual Cross-Construct Covariances
         VS.1 ~~ EXT.1
         VS.2 ~~ EXT.2
         VS.3 ~~ EXT.3
         VS.4 ~~ EXT.4
         VS.5 ~~ EXT.5

         # Regressions Between Adjacent Observations
         pVS.2 ~ 1*pVS.1
```

```
        pVS.3 ~ 1*pVS.2
        pVS.4 ~ 1*pVS.3
        pVS.5 ~ 1*pVS.4

        pEXT.2 ~ 1*pEXT.1
        pEXT.3 ~ 1*pEXT.2
        pEXT.4 ~ 1*pEXT.3
        pEXT.5 ~ 1*pEXT.4

        # Define Change Latent Variables
        dVS.21 =~ 1*pVS.2; dVS.21 ~~ 0*dVS.21
        dVS.32 =~ 1*pVS.3; dVS.32 ~~ 0*dVS.32
        dVS.43 =~ 1*pVS.4; dVS.43 ~~ 0*dVS.43
        dVS.54 =~ 1*pVS.5; dVS.54 ~~ 0*dVS.54

        dEXT.21 =~ 1*pEXT.2; dEXT.21 ~~ 0*dEXT.21
        dEXT.32 =~ 1*pEXT.3; dEXT.32 ~~ 0*dEXT.32
        dEXT.43 =~ 1*pEXT.4; dEXT.43 ~~ 0*dEXT.43
        dEXT.54 =~ 1*pEXT.5; dEXT.54 ~~ 0*dEXT.54

        # Define Within-Construct Proportional Regressions
        dVS.21 ~ beta.V*pVS.1
        dVS.32 ~ beta.V*pVS.2
        dVS.43 ~ beta.V*pVS.3
        dVS.54 ~ beta.V*pVS.4

        dEXT.21 ~ beta.E*pEXT.1
        dEXT.32 ~ beta.E*pEXT.2
        dEXT.43 ~ beta.E*pEXT.3
        dEXT.54 ~ beta.E*pEXT.4

        # Define Intercept and Slope
        int.V =~ 1*pVS.1
        slp.V =~ 1*dVS.21 + 1*dVS.32 + 1*dVS.43 + 1*dVS.54

        int.V ~ 1
        slp.V ~ 1
        int.V ~~ int.V + slp.V + int.E + slp.E
        slp.V ~~ slp.V + int.E + slp.E

        int.E =~ 1*pEXT.1
        slp.E =~ 1*dEXT.21 + 1*dEXT.32 + 1*dEXT.43 + 1*dEXT.54

        int.E ~ 1
        slp.E ~ 1
        int.E ~~ int.E + slp.E
        slp.E ~~ slp.E
'
bLCSt.fit = sem(bLCSt, data=ABCD, missing='ML')
summary(bLCSt.fit)
```

```
## lavaan 0.6-7 ended normally after 121 iterations
##
##   Estimator                                         ML
```

```
##   Optimization method                           NLMINB
##   Number of free parameters                         37
##   Number of equality constraints                     6
##
##   Number of observations                           964
##   Number of missing patterns                        47
##
## Model Test User Model:
##
##   Test statistic                                74.560
##   Degrees of freedom                                34
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   pVS.1 =~
##     VS.1              1.000
##   pVS.2 =~
##     VS.2              1.000
##   pVS.3 =~
##     VS.3              1.000
##   pVS.4 =~
##     VS.4              1.000
##   pVS.5 =~
##     VS.5              1.000
##   pEXT.1 =~
##     EXT.1             1.000
##   pEXT.2 =~
##     EXT.2             1.000
##   pEXT.3 =~
##     EXT.3             1.000
##   pEXT.4 =~
##     EXT.4             1.000
##   pEXT.5 =~
##     EXT.5             1.000
##   dVS.21 =~
##     pVS.2             1.000
##   dVS.32 =~
##     pVS.3             1.000
##   dVS.43 =~
##     pVS.4             1.000
##   dVS.54 =~
##     pVS.5             1.000
##   dEXT.21 =~
##     pEXT.2            1.000
##   dEXT.32 =~
##     pEXT.3            1.000
##   dEXT.43 =~
```

```
##     pEXT.4            1.000
##   dEXT.54 =~
##     pEXT.5            1.000
##   int.V =~
##     pVS.1             1.000
##   slp.V =~
##     dVS.21            1.000
##     dVS.32            1.000
##     dVS.43            1.000
##     dVS.54            1.000
##   int.E =~
##     pEXT.1            1.000
##   slp.E =~
##     dEXT.21           1.000
##     dEXT.32           1.000
##     dEXT.43           1.000
##     dEXT.54           1.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   pVS.2 ~
##     pVS.1             1.000
##   pVS.3 ~
##     pVS.2             1.000
##   pVS.4 ~
##     pVS.3             1.000
##   pVS.5 ~
##     pVS.4             1.000
##   pEXT.2 ~
##     pEXT.1            1.000
##   pEXT.3 ~
##     pEXT.2            1.000
##   pEXT.4 ~
##     pEXT.3            1.000
##   pEXT.5 ~
##     pEXT.4            1.000
##   dVS.21 ~
##     pVS.1   (bt.V)    0.348    0.108    3.211    0.001
##   dVS.32 ~
##     pVS.2   (bt.V)    0.348    0.108    3.211    0.001
##   dVS.43 ~
##     pVS.3   (bt.V)    0.348    0.108    3.211    0.001
##   dVS.54 ~
##     pVS.4   (bt.V)    0.348    0.108    3.211    0.001
##   dEXT.21 ~
##     pEXT.1  (bt.E)    0.365    0.158    2.319    0.020
##   dEXT.32 ~
##     pEXT.2  (bt.E)    0.365    0.158    2.319    0.020
##   dEXT.43 ~
##     pEXT.3  (bt.E)    0.365    0.158    2.319    0.020
##   dEXT.54 ~
##     pEXT.4  (bt.E)    0.365    0.158    2.319    0.020
##
## Covariances:
```

```
##                   Estimate  Std.Err  z-value  P(>|z|)
##  .VS.1 ~~
##    .EXT.1           -0.024    0.030   -0.797    0.425
##  .VS.2 ~~
##    .EXT.2            0.091    0.026    3.563    0.000
##  .VS.3 ~~
##    .EXT.3            0.120    0.027    4.505    0.000
##  .VS.4 ~~
##    .EXT.4            0.094    0.021    4.547    0.000
##  .VS.5 ~~
##    .EXT.5           -0.021    0.039   -0.532    0.595
##   int.V ~~
##     slp.V           -0.319    0.107   -2.977    0.003
##     int.E            0.044    0.032    1.372    0.170
##     slp.E            0.005    0.021    0.252    0.801
##   slp.V ~~
##     int.E            0.019    0.019    1.005    0.315
##     slp.E           -0.018    0.008   -2.303    0.021
##   int.E ~~
##     slp.E           -0.156    0.063   -2.488    0.013
##
## Intercepts:
##                   Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1             0.000
##    .VS.2             0.000
##    .VS.3             0.000
##    .VS.4             0.000
##    .VS.5             0.000
##    .EXT.1            0.000
##    .EXT.2            0.000
##    .EXT.3            0.000
##    .EXT.4            0.000
##    .EXT.5            0.000
##     int.V            6.267    0.044  141.086    0.000
##     slp.V           -2.063    0.705   -2.925    0.003
##     int.E            2.342    0.026   91.672    0.000
##     slp.E           -0.810    0.380   -2.131    0.033
##    .pVS.1            0.000
##    .pVS.2            0.000
##    .pVS.3            0.000
##    .pVS.4            0.000
##    .pVS.5            0.000
##    .pEXT.1           0.000
##    .pEXT.2           0.000
##    .pEXT.3           0.000
##    .pEXT.4           0.000
##    .pEXT.5           0.000
##    .dVS.21           0.000
##    .dVS.32           0.000
##    .dVS.43           0.000
##    .dVS.54           0.000
##    .dEXT.21          0.000
##    .dEXT.32          0.000
##    .dEXT.43          0.000
```

```
##    .dEXT.54           0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              1.338    0.084   15.967    0.000
##    .pVS.1             0.000
##    .VS.2              1.339    0.076   17.546    0.000
##    .pVS.2             0.000
##    .VS.3              1.110    0.061   18.093    0.000
##    .pVS.3             0.000
##    .VS.4              1.018    0.062   16.406    0.000
##    .pVS.4             0.000
##    .VS.5              0.118    0.091    1.293    0.196
##    .pVS.5             0.000
##    .EXT.1             0.307    0.023   13.433    0.000
##    .pEXT.1            0.000
##    .EXT.2             0.254    0.016   15.388    0.000
##    .pEXT.2            0.000
##    .EXT.3             0.372    0.021   17.657    0.000
##    .pEXT.3            0.000
##    .EXT.4             0.151    0.015    9.718    0.000
##    .pEXT.4            0.000
##    .EXT.5             0.819    0.056   14.520    0.000
##    .pEXT.5            0.000
##    .dVS.21            0.000
##    .dVS.32            0.000
##    .dVS.43            0.000
##    .dVS.54            0.000
##    .dEXT.21           0.000
##    .dEXT.32           0.000
##    .dEXT.43           0.000
##    .dEXT.54           0.000
##     int.V             0.942    0.073   12.904    0.000
##     slp.V             0.141    0.061    2.308    0.021
##     int.E             0.415    0.028   14.643    0.000
##     slp.E             0.068    0.043    1.565    0.118
```

```
# Bivariate Dual Change LCS Trajectory Model
bLCSpt = '# Define Phantom Variables
          pVS.1 =~ 1*VS.1; VS.1 ~ 0; VS.1 ~~ VS.1; pVS.1 ~~ 0*pVS.1
          pVS.2 =~ 1*VS.2; VS.2 ~ 0; VS.2 ~~ VS.2; pVS.2 ~~ 0*pVS.2
          pVS.3 =~ 1*VS.3; VS.3 ~ 0; VS.3 ~~ VS.3; pVS.3 ~~ 0*pVS.3
          pVS.4 =~ 1*VS.4; VS.4 ~ 0; VS.4 ~~ VS.4; pVS.4 ~~ 0*pVS.4
          pVS.5 =~ 1*VS.5; VS.5 ~ 0; VS.5 ~~ VS.5; pVS.5 ~~ 0*pVS.5

          pEXT.1 =~ 1*EXT.1; EXT.1 ~ 0; EXT.1 ~~ EXT.1; pEXT.1 ~~ 0*pEXT.1
          pEXT.2 =~ 1*EXT.2; EXT.2 ~ 0; EXT.2 ~~ EXT.2; pEXT.2 ~~ 0*pEXT.2
          pEXT.3 =~ 1*EXT.3; EXT.3 ~ 0; EXT.3 ~~ EXT.3; pEXT.3 ~~ 0*pEXT.3
          pEXT.4 =~ 1*EXT.4; EXT.4 ~ 0; EXT.4 ~~ EXT.4; pEXT.4 ~~ 0*pEXT.4
          pEXT.5 =~ 1*EXT.5; EXT.5 ~ 0; EXT.5 ~~ EXT.5; pEXT.5 ~~ 0*pEXT.5

          # Residual Cross-Construct Covariances
          VS.1 ~~ EXT.1
          VS.2 ~~ EXT.2
```

```
VS.3 ~~ EXT.3
VS.4 ~~ EXT.4
VS.5 ~~ EXT.5

# Regressions Between Adjacent Observations
pVS.2 ~ 1*pVS.1
pVS.3 ~ 1*pVS.2
pVS.4 ~ 1*pVS.3
pVS.5 ~ 1*pVS.4

pEXT.2 ~ 1*pEXT.1
pEXT.3 ~ 1*pEXT.2
pEXT.4 ~ 1*pEXT.3
pEXT.5 ~ 1*pEXT.4

# Define Change Latent Variables
dVS.21 =~ 1*pVS.2; dVS.21 ~~ 0*dVS.21
dVS.32 =~ 1*pVS.3; dVS.32 ~~ 0*dVS.32
dVS.43 =~ 1*pVS.4; dVS.43 ~~ 0*dVS.43
dVS.54 =~ 1*pVS.5; dVS.54 ~~ 0*dVS.54

dEXT.21 =~ 1*pEXT.2; dEXT.21 ~~ 0*dEXT.21
dEXT.32 =~ 1*pEXT.3; dEXT.32 ~~ 0*dEXT.32
dEXT.43 =~ 1*pEXT.4; dEXT.43 ~~ 0*dEXT.43
dEXT.54 =~ 1*pEXT.5; dEXT.54 ~~ 0*dEXT.54

# Define Within- and Between Construct Proportional Regressions
dVS.21 ~ beta.V*pVS.1 + beta.VE*pEXT.1
dVS.32 ~ beta.V*pVS.2 + beta.VE*pEXT.2
dVS.43 ~ beta.V*pVS.3 + beta.VE*pEXT.3
dVS.54 ~ beta.V*pVS.4 + beta.VE*pEXT.4

dEXT.21 ~ beta.E*pEXT.1 + beta.EV*pVS.1
dEXT.32 ~ beta.E*pEXT.2 + beta.EV*pVS.2
dEXT.43 ~ beta.E*pEXT.3 + beta.EV*pVS.3
dEXT.54 ~ beta.E*pEXT.4 + beta.EV*pVS.4

# Define Intercept and Slope
int.V =~ 1*pVS.1
slp.V =~ 1*dVS.21 + 1*dVS.32 + 1*dVS.43 + 1*dVS.54

int.V ~ 1
slp.V ~ 1
int.V ~~ int.V + slp.V + int.E + slp.E
slp.V ~~ slp.V + int.E + slp.E

int.E =~ 1*pEXT.1
slp.E =~ 1*dEXT.21 + 1*dEXT.32 + 1*dEXT.43 + 1*dEXT.54

int.E ~ 1
slp.E ~ 1
int.E ~~ int.E + slp.E
slp.E ~~ slp.E
```

```r
'
bLCSpt.fit = sem(bLCSpt, data=ABCD, missing='ML')
summary(bLCSpt.fit)
```

```
## lavaan 0.6-7 ended normally after 177 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         45
##   Number of equality constraints                    12
##
##   Number of observations                           964
##   Number of missing patterns                        47
##
## Model Test User Model:
##
##   Test statistic                                74.006
##   Degrees of freedom                                32
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   pVS.1 =~
##     VS.1              1.000
##   pVS.2 =~
##     VS.2              1.000
##   pVS.3 =~
##     VS.3              1.000
##   pVS.4 =~
##     VS.4              1.000
##   pVS.5 =~
##     VS.5              1.000
##   pEXT.1 =~
##     EXT.1             1.000
##   pEXT.2 =~
##     EXT.2             1.000
##   pEXT.3 =~
##     EXT.3             1.000
##   pEXT.4 =~
##     EXT.4             1.000
##   pEXT.5 =~
##     EXT.5             1.000
##   dVS.21 =~
##     pVS.2             1.000
##   dVS.32 =~
##     pVS.3             1.000
##   dVS.43 =~
##     pVS.4             1.000
```

```
##   dVS.54 =~
##     pVS.5             1.000
##   dEXT.21 =~
##     pEXT.2            1.000
##   dEXT.32 =~
##     pEXT.3            1.000
##   dEXT.43 =~
##     pEXT.4            1.000
##   dEXT.54 =~
##     pEXT.5            1.000
##   int.V =~
##     pVS.1             1.000
##   slp.V =~
##     dVS.21            1.000
##     dVS.32            1.000
##     dVS.43            1.000
##     dVS.54            1.000
##   int.E =~
##     pEXT.1            1.000
##   slp.E =~
##     dEXT.21           1.000
##     dEXT.32           1.000
##     dEXT.43           1.000
##     dEXT.54           1.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   pVS.2 ~
##     pVS.1             1.000
##   pVS.3 ~
##     pVS.2             1.000
##   pVS.4 ~
##     pVS.3             1.000
##   pVS.5 ~
##     pVS.4             1.000
##   pEXT.2 ~
##     pEXT.1            1.000
##   pEXT.3 ~
##     pEXT.2            1.000
##   pEXT.4 ~
##     pEXT.3            1.000
##   pEXT.5 ~
##     pEXT.4            1.000
##   dVS.21 ~
##     pVS.1   (bt.V)    0.322    0.120    2.682    0.007
##     pEXT.1  (b.VE)    0.118    0.253    0.465    0.642
##   dVS.32 ~
##     pVS.2   (bt.V)    0.322    0.120    2.682    0.007
##     pEXT.2  (b.VE)    0.118    0.253    0.465    0.642
##   dVS.43 ~
##     pVS.3   (bt.V)    0.322    0.120    2.682    0.007
##     pEXT.3  (b.VE)    0.118    0.253    0.465    0.642
##   dVS.54 ~
##     pVS.4   (bt.V)    0.322    0.120    2.682    0.007
```

```
##     pEXT.4  (b.VE)    0.118    0.253    0.465    0.642
##   dEXT.21 ~
##     pEXT.1  (bt.E)    0.324    0.166    1.955    0.051
##     pVS.1   (b.EV)    0.047    0.067    0.708    0.479
##   dEXT.32 ~
##     pEXT.2  (bt.E)    0.324    0.166    1.955    0.051
##     pVS.2   (b.EV)    0.047    0.067    0.708    0.479
##   dEXT.43 ~
##     pEXT.3  (bt.E)    0.324    0.166    1.955    0.051
##     pVS.3   (b.EV)    0.047    0.067    0.708    0.479
##   dEXT.54 ~
##     pEXT.4  (bt.E)    0.324    0.166    1.955    0.051
##     pVS.4   (b.EV)    0.047    0.067    0.708    0.479
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##  .VS.1 ~~
##    .EXT.1            -0.016    0.033   -0.479    0.632
##  .VS.2 ~~
##    .EXT.2             0.089    0.026    3.495    0.000
##  .VS.3 ~~
##    .EXT.3             0.119    0.027    4.465    0.000
##  .VS.4 ~~
##    .EXT.4             0.099    0.023    4.353    0.000
##  .VS.5 ~~
##    .EXT.5            -0.045    0.049   -0.911    0.362
##   int.V ~~
##     slp.V            -0.309    0.108   -2.866    0.004
##     int.E             0.041    0.033    1.228    0.219
##     slp.E            -0.034    0.061   -0.555    0.579
##   slp.V ~~
##     int.E            -0.024    0.098   -0.247    0.805
##     slp.E             0.008    0.037    0.206    0.837
##   int.E ~~
##     slp.E            -0.145    0.063   -2.290    0.022
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              0.000
##    .VS.2              0.000
##    .VS.3              0.000
##    .VS.4              0.000
##    .VS.5              0.000
##    .EXT.1             0.000
##    .EXT.2             0.000
##    .EXT.3             0.000
##    .EXT.4             0.000
##    .EXT.5             0.000
##     int.V             6.270    0.045  139.286    0.000
##     slp.V            -2.184    0.743   -2.941    0.003
##     int.E             2.345    0.026   89.978    0.000
##     slp.E            -1.018    0.475   -2.145    0.032
##    .pVS.1             0.000
##    .pVS.2             0.000
```

```
##    .pVS.3             0.000
##    .pVS.4             0.000
##    .pVS.5             0.000
##    .pEXT.1            0.000
##    .pEXT.2            0.000
##    .pEXT.3            0.000
##    .pEXT.4            0.000
##    .pEXT.5            0.000
##    .dVS.21            0.000
##    .dVS.32            0.000
##    .dVS.43            0.000
##    .dVS.54            0.000
##    .dEXT.21           0.000
##    .dEXT.32           0.000
##    .dEXT.43           0.000
##    .dEXT.54           0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .VS.1              1.327    0.087   15.280    0.000
##    .pVS.1             0.000
##    .VS.2              1.341    0.076   17.545    0.000
##    .pVS.2             0.000
##    .VS.3              1.110    0.061   18.118    0.000
##    .pVS.3             0.000
##    .VS.4              1.014    0.063   16.025    0.000
##    .pVS.4             0.000
##    .VS.5              0.134    0.098    1.377    0.169
##    .pVS.5             0.000
##    .EXT.1             0.304    0.024   12.712    0.000
##    .pEXT.1            0.000
##    .EXT.2             0.254    0.017   15.399    0.000
##    .pEXT.2            0.000
##    .EXT.3             0.372    0.021   17.670    0.000
##    .pEXT.3            0.000
##    .EXT.4             0.149    0.016    9.527    0.000
##    .pEXT.4            0.000
##    .EXT.5             0.829    0.057   14.604    0.000
##    .pEXT.5            0.000
##    .dVS.21            0.000
##    .dVS.32            0.000
##    .dVS.43            0.000
##    .dVS.54            0.000
##    .dEXT.21           0.000
##    .dEXT.32           0.000
##    .dEXT.43           0.000
##    .dEXT.54           0.000
##     int.V             0.951    0.076   12.514    0.000
##     slp.V             0.135    0.058    2.339    0.019
##     int.E             0.418    0.030   14.024    0.000
##     slp.E             0.061    0.039    1.545    0.122
```