Plotting with Lavaan

Emorie Beck

10/19/2017

Contents

"https://github.com/emoriebeck/R-tutorials/tree/master/sem"

Plotting with Lavaan

```
library(lavaan)
library(psych)
library(semPlot)
library(plyr)
library(tidyverse)

data_path <- "https://github.com/longitudinal-data/1-descriptives-and-graphs-emoriebeck/raw/master/Condi
load(url(paste(data_path, "sample.RData", sep = "/")))

sample_dat_wide <- sample_dat %>%
  select(-year, -age, -CESD) %>%
  mutate(age0 = paste("A", age0, sep = ".")) %>%
  spread(key = age0, value = SensSeek) %>%
  full_join(
    sample_dat %>% select(-year, -age, -DemPweight, -SensSeek) %>%
      mutate(age0 = paste("D", age0, sep = ".")) %>%
      spread(key = age0, value = CESD)
  )
```

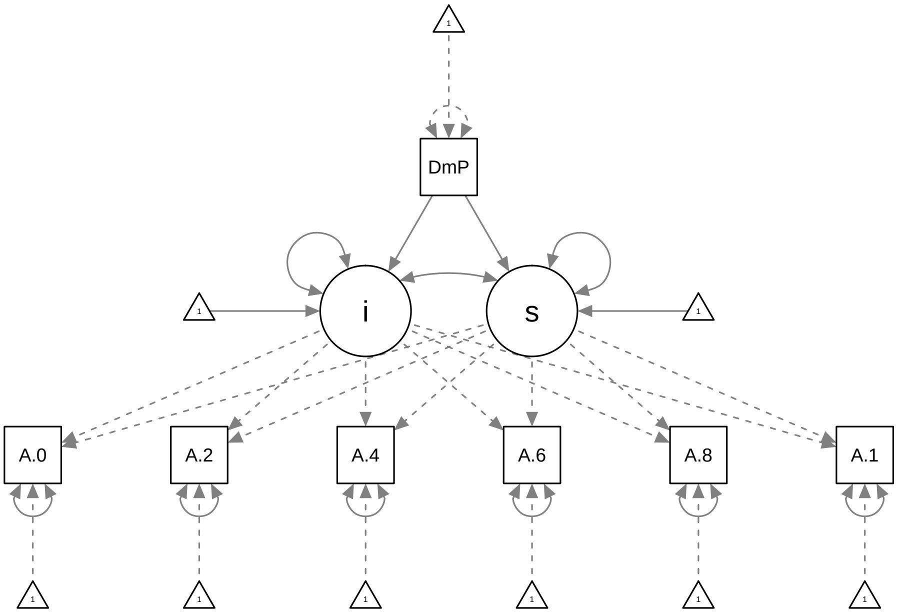
Time Invariant Predictors

Continuous

```
## linear growth model with a time-varying covariate
model.syntax <- '
  # intercept and slope with fixed coefficients
    i =~ 1*A.0 + 1*A.2 + 1*A.4 + 1*A.6 + 1*A.8 + 1*A.10
    s =~ 0*A.0 + 1*A.2 + 2*A.4 + 3*A.6 + 4*A.8 + 5*A.10

  # regressions
    i ~ DemPweight
    s ~ DemPweight
'

fit1 <- growth(model.syntax, data=sample_dat_wide, missing = "FIML")
summary(fit1)
```

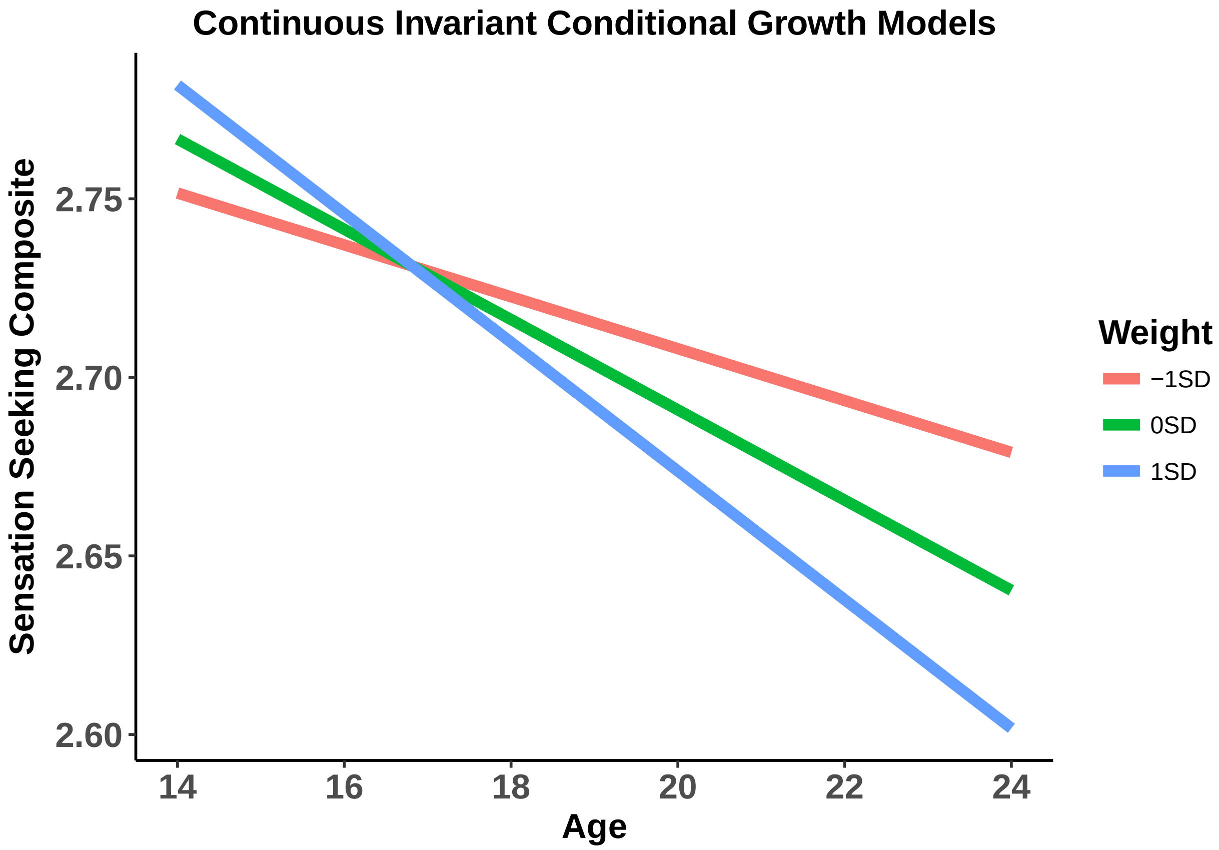
```
## lavaan (0.5-23.1097) converged normally after  38 iterations
##
##   Number of observations                           924
##
##   Number of missing patterns                        46
##
##   Estimator                                         ML
##   Minimum Function Test Statistic               45.354
##   Degrees of freedom                                20
##   P-value (Chi-square)                           0.001
##
## Parameter Estimates:
##
##   Information                                 Observed
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   i =~
##     A.0               1.000
##     A.2               1.000
##     A.4               1.000
##     A.6               1.000
##     A.8               1.000
##     A.10              1.000
##   s =~
##     A.0               0.000
##     A.2               1.000
##     A.4               2.000
##     A.6               3.000
##     A.8               4.000
##     A.10              5.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   i ~
##     DemPweight        0.049    0.066    0.745    0.456
##   s ~
##     DemPweight       -0.017    0.023   -0.740    0.459
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##  .i ~~
##    .s                -0.003    0.006   -0.553    0.581
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .A.0               0.000
##    .A.2              0.000
##    .A.4              0.000
##    .A.6              0.000
##    .A.8              0.000
##    .A.10             0.000
##    .i                2.735    0.047   57.593    0.000
##    .s               -0.001    0.017   -0.071    0.944
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .A.0              0.215    0.025    8.695    0.000
##    .A.2              0.201    0.017   12.109    0.000
##    .A.4              0.222    0.021   10.569    0.000
##    .A.6              0.138    0.019    7.452    0.000
##    .A.8              0.132    0.024    5.479    0.000
##    .A.10             0.131    0.030    4.386    0.000
##    .i                0.129    0.020    6.515    0.000
##    .s                0.005    0.002    2.395    0.017
```

```
semPaths(fit1)
```



```
get_fixef <- function(fit, target.var){
    df <- parameterestimates(fit) %>%
      filter(lhs != rhs & lhs %in% target.var & op %in% c("~", "~1")) %>%
      unite(term, lhs, op, rhs, sep = "")
    v <- as.vector(df$est)
    names(v) <- df$term
    return(v)
  }

# example for continuous
fixed.frame <- sample_dat %>%
  group_by(PROC_CID) %>%
  summarise(DemPweight = mean(DemPweight, na.rm = T)) %>%
  ungroup() %>%
```

```r
  summarise(mean = mean(DemPweight, na.rm = T),
            sd = sd(DemPweight, na.rm = T))

fixed.frame <-
  data.frame(
    expand.grid(
      # here, you add values for your time variable and predictors
      age0 = seq(0,10,2),
      DemPweight = c(fixed.frame$mean-fixed.frame$sd,
                     fixed.frame$mean,
                     fixed.frame$mean+fixed.frame$sd))) %>%
  # now take care of interactions and add an intercept
  mutate(`age0:DemPweight` = age0*DemPweight,
         Intercept = 1) %>%
  # reordering everything
  select(DemPweight, `age0:DemPweight`, Intercept, age0)

fixed.frame$y <- as.vector(as.matrix(fixed.frame) %*% get_fixef(fit1, c("i", "s")))

tbl_df(fixed.frame) %>%
  mutate(Weight = factor(DemPweight, levels = unique(DemPweight), labels = c("-1SD", "0SD", "1SD")),
         age = age0 + 14) %>%
  ggplot(aes(x = age, y = y, color = Weight)) +
    geom_line(size = 2) +
    labs(x = "Age", y = "Sensation Seeking Composite",
         title = "Continuous Invariant Conditional Growth Models") +
    theme_classic() +
    theme(axis.text = element_text(face = "bold", size = rel(1.2)),
          axis.title = element_text(face = "bold", size = rel(1.2)),
          legend.title = element_text(face = "bold", size = rel(1.2)),
          plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
```



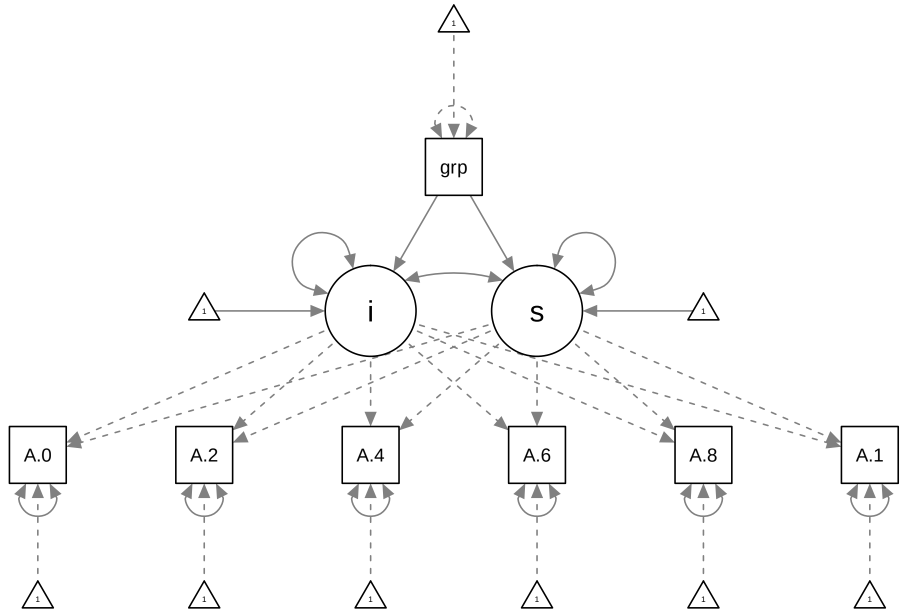
Categorical

```
## linear growth model with a time-varying covariate
model.syntax <- '
  # intercept and slope with fixed coefficients
    i =~ 1*A.0 + 1*A.2 + 1*A.4 + 1*A.6 + 1*A.8 + 1*A.10
    s =~ 0*A.0 + 1*A.2 + 2*A.4 + 3*A.6 + 4*A.8 + 5*A.10

  # regressions
    i ~ groups
    s ~ groups
'

fit2 <- growth(model.syntax, data=sample_dat_wide %>% filter(groups != "CommServ"), missing = "FIML")
summary(fit2)
```

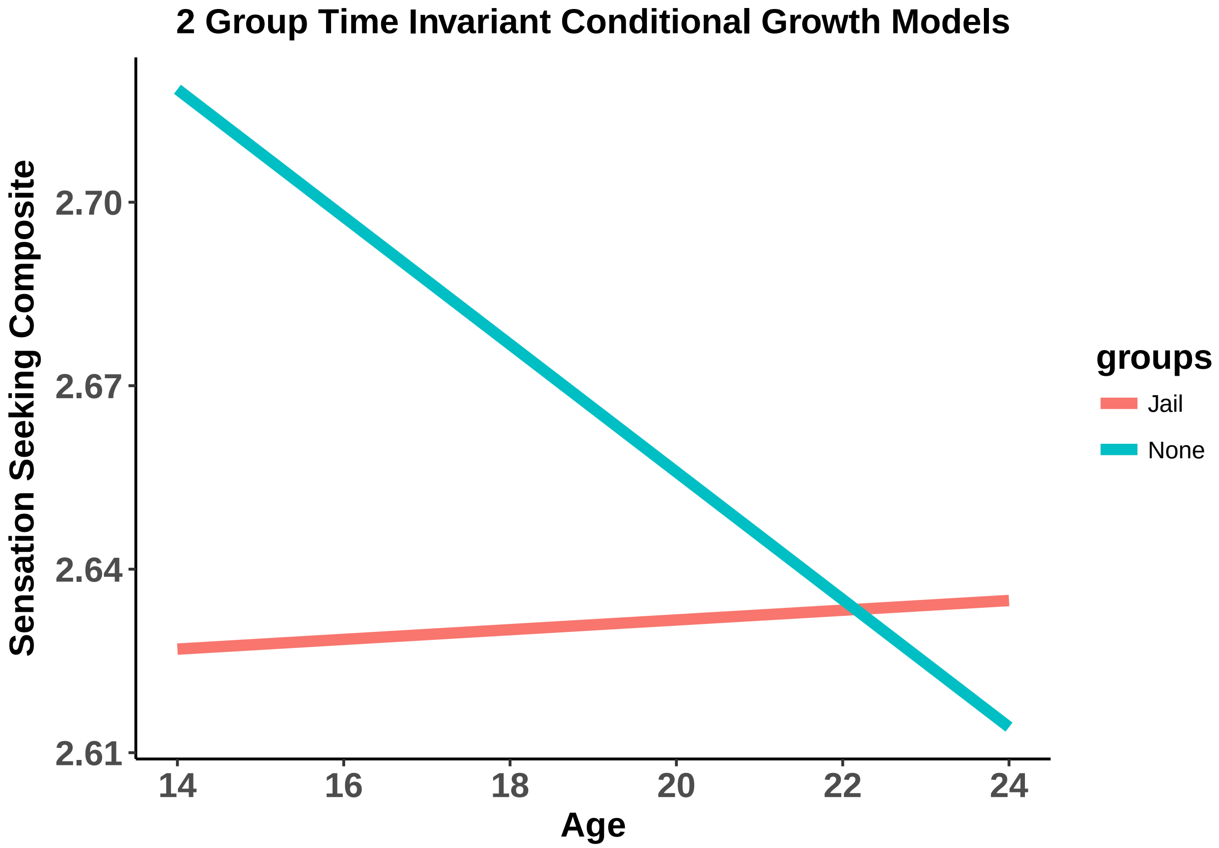
```
## lavaan (0.5-23.1097) converged normally after  55 iterations
##
##   Number of observations                           689
##
##   Number of missing patterns                        46
##
##   Estimator                                         ML
##   Minimum Function Test Statistic               38.403
##   Degrees of freedom                                20
##   P-value (Chi-square)                           0.008
```

```
##
## Parameter Estimates:
##
##   Information                                 Observed
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   i =~
##     A.0               1.000
##     A.2               1.000
##     A.4               1.000
##     A.6               1.000
##     A.8               1.000
##     A.10              1.000
##   s =~
##     A.0               0.000
##     A.2               1.000
##     A.4               2.000
##     A.6               3.000
##     A.8               4.000
##     A.10              5.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   i ~
##     groups            0.092    0.048    1.897    0.058
##   s ~
##     groups           -0.011    0.016   -0.689    0.491
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##  .i ~~
##    .s                -0.004    0.006   -0.679    0.497
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .A.0               0.000
##    .A.2               0.000
##    .A.4               0.000
##    .A.6               0.000
##    .A.8               0.000
##    .A.10              0.000
##    .i                 2.627    0.079   33.071    0.000
##    .s                 0.001    0.027    0.029    0.977
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .A.0               0.228    0.030    7.645    0.000
##    .A.2               0.196    0.019   10.128    0.000
##    .A.4               0.228    0.025    9.211    0.000
##    .A.6               0.143    0.022    6.593    0.000
##    .A.8               0.111    0.025    4.418    0.000
##    .A.10              0.145    0.034    4.322    0.000
##    .i                  0.142    0.024    6.030    0.000
##    .s                  0.005    0.002    2.010    0.044
```

```
semPaths(fit2)
```



```
get_fixef <- function(fit, target.var){
    df <- parameterestimates(fit) %>%
      filter(lhs != rhs & lhs %in% target.var & op %in% c("~", "~1")) %>%
      unite(term, lhs, op, rhs, sep = "")
    v <- as.vector(df$est)
    names(v) <- df$term
    return(v)
  }

# example for categorical
fixed.frame <-
  data.frame(
    expand.grid(
      # here, you add values for your time variable and predictors
      Intercept = 1,
      age0 = seq(0,10,2),
      groupsNone = c(0,1))) %>%
  # now take care of interactions and add an intercept
  mutate(`age0:groupsNone` = age0*groupsNone) %>%
  select(groupsNone, `age0:groupsNone`, Intercept, age0)

fixed.frame$y <- as.vector(as.matrix(fixed.frame) %*% get_fixef(fit2, c("i", "s")))

fixed.frame %>%
  mutate(groups = factor(groupsNone, levels = c(0,1), labels = c("Jail", "None")),
         age = age0 + 14) %>%
  ggplot(aes(x = age, y = y, color = groups)) +
    geom_line(size = 2) +
    labs(x = "Age", y = "Sensation Seeking Composite",
         title = "2 Group Time Invariant Conditional Growth Models") +
    theme_classic() +
    theme(axis.text = element_text(face = "bold", size = rel(1.2)),
          axis.title = element_text(face = "bold", size = rel(1.2)),
          legend.title = element_text(face = "bold", size = rel(1.2)),
          plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
```



Tables

```
table_fun <- function(mod, int.var, slp.var, covar){
  var <- c(int.var, slp.var, covar)
  df <- parameterestimates(mod) %>%
      filter((lhs != rhs & lhs %in% var & op %in% c("~", "~1")) |
             (lhs == rhs & lhs %in% var & op == "~~")) %>%
      mutate(term = ifelse(lhs %in% int.var & rhs %in% covar, rhs,
                    ifelse(lhs %in% slp.var & rhs %in% covar, paste("Slope",rhs, sep = ":"),
                    ifelse(op %in% c("~1", "~~") , plyr::mapvalues(lhs, var, c("Intercept","Slope",cova
                    lhs))),
             type = plyr::mapvalues(op, c("~", "~1", "~~"), c("Fixed", "Fixed", "Random"))) %>%
    select(term, type, est, ci.lower, ci.upper)

  fitmeas <- data.frame(fitmeasures(mod)) %>%
    mutate(term = rownames(.), type = "fitmeas") %>%
    dplyr::rename(est = fitmeasures.mod.) %>%
    filter(term %in% c("cfi", "rmsea", "chisq", "df")) %>%
    mutate(term = mapvalues(term, unique(term), c("$\\chi^2$", "df", "CFI", "RMSEA")))

  df <- df %>% full_join(fitmeas)
}
```

Continuous Predictor

```
tab <- table_fun(fit1, "i", "s", "DemPweight")

(form.tab <- tab %>%
  filter(type != "fitmeas") %>%
  mutate(sig = ifelse(sign(ci.lower) == sign(ci.upper), "sig", "nonsig")) %>%
  mutate_at(vars(est:ci.upper), funs(sprintf("%.2f", .))) %>%
  mutate(CI = sprintf("[%s, %s]", ci.lower, ci.upper)) %>%
  mutate_at(vars(est, CI), funs(ifelse(sig == "sig", sprintf("\\textbf{%s}", .), .))) %>%
  select(term:est, CI) %>%
  full_join(tab %>% filter(type == "fitmeas") %>%
              mutate(est = sprintf("%.2f", est)) %>%
              select(term:est)) %>%
  mutate(type = factor(type, levels = c("Fixed", "Random", "fitmeas"))) %>%
  arrange(type))
```

```
##                 term    type            est                     CI
## 1         DemPweight   Fixed           0.05          [-0.08, 0.18]
## 2   Slope:DemPweight   Fixed          -0.02          [-0.06, 0.03]
## 3         DemPweight   Fixed \\textbf{0.66} \\textbf{[0.66, 0.66]}
## 4          Intercept   Fixed \\textbf{2.73} \\textbf{[2.64, 2.83]}
## 5              Slope   Fixed          -0.00          [-0.03, 0.03]
## 6          Intercept  Random \\textbf{0.13} \\textbf{[0.09, 0.17]}
## 7              Slope  Random \\textbf{0.01} \\textbf{[0.00, 0.01]}
## 8         DemPweight  Random \\textbf{0.10} \\textbf{[0.10, 0.10]}
## 9           $\\chi^2$ fitmeas          45.35                   <NA>
## 10                df fitmeas          20.00                   <NA>
## 11               CFI fitmeas           0.92                   <NA>
## 12             RMSEA fitmeas           0.04                   <NA>
```

```
options(knitr.kable.NA = '')
form.tab %>% select(-type) %>%
  papaja::apa_table(caption = "Sample Lavaan Table",
      stub_indents = list("Fixed" = c(1:5),
                          "Random" = c(6:8),
                          "Fit Measures" = c(9:12)))
```

Categorical Predictor

```
tab <- table_fun(fit2, "i", "s", "groups")

(form.tab <- tab %>%
  filter(type != "fitmeas") %>%
  mutate(sig = ifelse(sign(ci.lower) == sign(ci.upper), "sig", "nonsig")) %>%
  mutate_at(vars(est:ci.upper), funs(sprintf("%.2f", .))) %>%
```

Table 1: Sample Lavaan Table

term	est	CI
Fixed		
DemPweight	0.05	[-0.08, 0.18]
Slope:DemPweight	-0.02	[-0.06, 0.03]
DemPweight	**0.66**	**[0.66, 0.66]**
Intercept	**2.73**	**[2.64, 2.83]**
Slope	-0.00	[-0.03, 0.03]
Random		
Intercept	**0.13**	**[0.09, 0.17]**
Slope	**0.01**	**[0.00, 0.01]**
DemPweight	**0.10**	**[0.10, 0.10]**
Fit Measures		
χ^2	45.35	
df	20.00	
CFI	0.92	
RMSEA	0.04	

```
  mutate(CI = sprintf("[%s, %s]", ci.lower, ci.upper)) %>%
  mutate_at(vars(est, CI), funs(ifelse(sig == "sig", sprintf("\\textbf{%s}", .), .))) %>%
  select(term:est, CI) %>%
  full_join(tab %>% filter(type == "fitmeas") %>%
              mutate(est = sprintf("%.2f", est)) %>%
              select(term:est)) %>%
  mutate(type = factor(type, levels = c("Fixed", "Random", "fitmeas"))) %>%
  arrange(type))
```

```
##            term    type            est                     CI
## 1        groups   Fixed           0.09          [-0.00, 0.19]
## 2  Slope:groups   Fixed          -0.01          [-0.04, 0.02]
## 3        groups   Fixed \\textbf{1.56} \\textbf{[1.56, 1.56]}
## 4     Intercept   Fixed \\textbf{2.63} \\textbf{[2.47, 2.78]}
## 5         Slope   Fixed           0.00          [-0.05, 0.05]
## 6     Intercept  Random \\textbf{0.14} \\textbf{[0.10, 0.19]}
## 7         Slope  Random \\textbf{0.00} \\textbf{[0.00, 0.01]}
## 8        groups  Random \\textbf{0.25} \\textbf{[0.25, 0.25]}
## 9     $\\chi^2$ fitmeas          38.40                   <NA>
## 10           df fitmeas          20.00                   <NA>
## 11          CFI fitmeas           0.93                   <NA>
## 12        RMSEA fitmeas           0.04                   <NA>
```

```
options(knitr.kable.NA = '')
form.tab %>% select(-type) %>%
  papaja::apa_table(caption = "Sample Lavaan Table",
      stub_indents = list("Fixed" = c(1:5),
                          "Random" = c(6:8),
                          "Fit Measures" = c(9:12)))
```

Table 2: Sample Lavaan Table

term	est	CI
Fixed		
groups	0.09	[-0.00, 0.19]
Slope:groups	-0.01	[-0.04, 0.02]
groups	**1.56**	**[1.56, 1.56]**
Intercept	**2.63**	**[2.47, 2.78]**
Slope	0.00	[-0.05, 0.05]
Random		
Intercept	**0.14**	**[0.10, 0.19]**
Slope	**0.00**	**[0.00, 0.01]**
groups	**0.25**	**[0.25, 0.25]**
Fit Measures		
χ^2	38.40	
df	20.00	
CFI	0.93	
RMSEA	0.04	