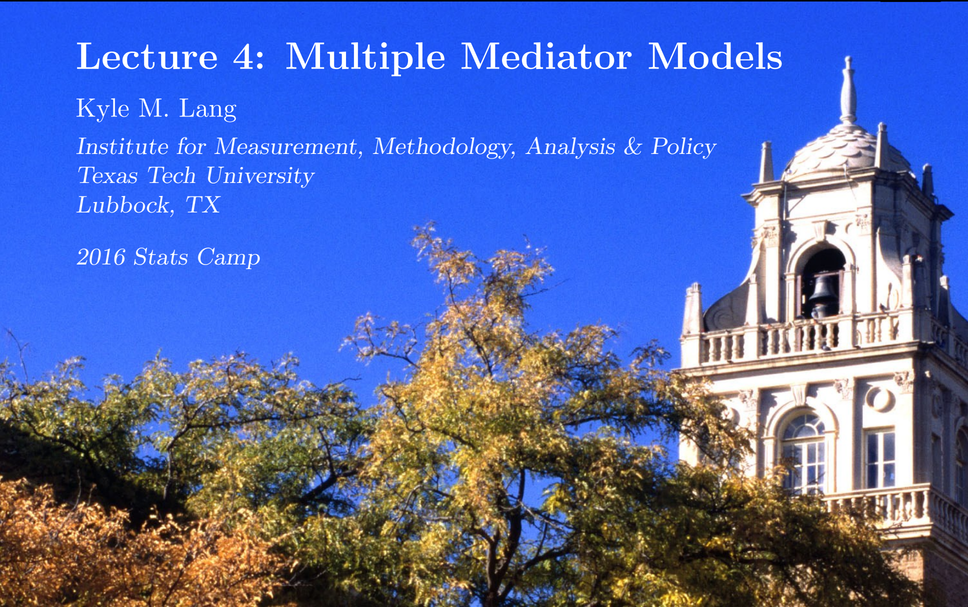
TEXAS TECH UNIVERSITY

Lecture 4: Multiple Mediator Models

Kyle M. Lang

Institute for Measurement, Methodology, Analysis & Policy
Texas Tech University
Lubbock, TX

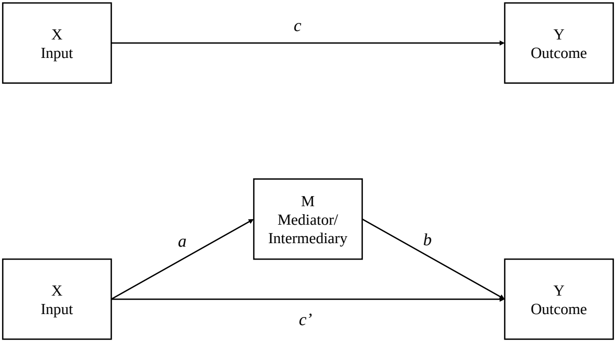
2016 Stats Camp



Outline

- Introduce the two flavors of multiple mediator model
  - Parallel Multiple Mediator Models
  - Serial Multiple Mediator Models
- Discuss methods for testing for statistical differences in the multiple indirect effects

Simple Mediation

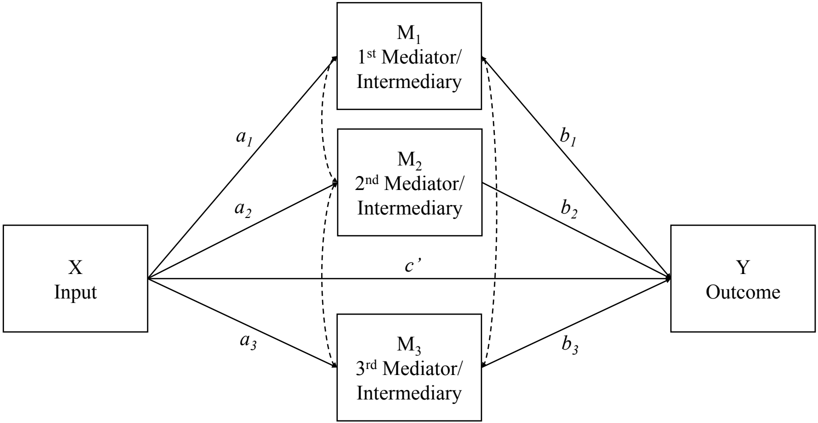


Simple Mediation is Too Simple

We can justify multiple mediator models by asking: "What mediates the effects in a simple mediation model?"

- Mediation of the direct effect leads to *parallel multiple mediator models*.
- Mediation of the a or b paths produces *serial multiple mediator models*.

Parallel Multiple Mediation



Parallel Multiple Mediation

To get all of the information in the preceding diagram, we need to fit four equations:

$$
\begin{aligned}
Y &= i_Y + b_1 M_1 + b_2 M_2 + b_3 M_3 + c' X + e_Y \\
M_1 &= i_{M1} + a_1 X + e_{M1} \\
M_2 &= i_{M2} + a_2 X + e_{M2} \\
M_3 &= i_{M3} + a_3 X + e_{M3}
\end{aligned}
$$

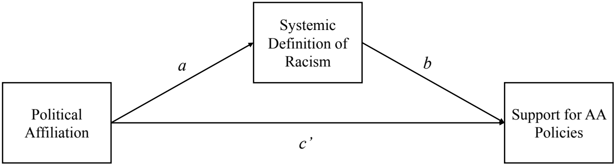
In general, a parallel mediator model with K mediator variables will required $K + 1$ separate equations.

Path modeling can make this task much simpler.

- Also allows us to explicitly estimate the correlations between parallel mediators.

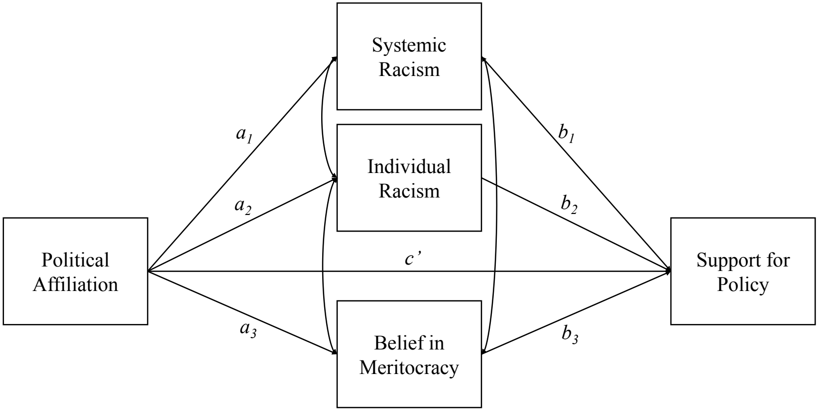
Parallel Multiple Mediation

Let's reconsider the example from last week:



QUESTION: What might be mediating the residual direct effect?

Potential Answer:



A Quick Note on Inference

In simple mediation:

- We have one indirect effect: ab.
- The total effect is equal to the direct effect plus the indirect effect $c = c' + ab$

In parallel multiple mediation:

- We have K *specific indirect effects*, where K is the number of mediators: $a_1 b_1, a_2 b_2, \ldots, a_K b_K$.
- The *Total Indirect Effect* is equal to the sum of all the specific indirect effects: $IE_{tot} = \sum_{k=1}^{K} a_k b_k$.
- The *Total Effect* is equal to the direct effect plus the total indirect effect: $c = c' + IE_{tot}$

Inference for the specific indirect effects is basically the same as it is for the sole indirect effect in simple mediation.

- CAVEAT: Each specific indirect effect must be interpreted as conditional on all other mediators in the model.

Example

```r
library(lavaan)
## Read in the data
dataDir <- "../data/"
fileName <- "adamsKlpsScaleScores.rds"
dat1 <- readRDS(paste0(dataDir, fileName))
nBoot <- 2500 # Number of bootstrap samples
bootType <- "bca.simple" # Type of CI
## Parallel Multiple Mediator Model:
mod1.1 <- "
policy ~ b1*sysRac + b2*indRac + b3*merit + cp*polAffil
sysRac ~ a1*polAffil
indRac ~ a2*polAffil
merit ~ a3*polAffil

sysRac ~~ indRac + merit
indRac ~~ merit

ab1 := a1*b1
ab2 := a2*b2
ab3 := a3*b3
totalIE := ab1 + ab2 + ab3
"
```

Example

```
## Fit the model:
out1.1 <-
    sem(mod1.1, data = dat1, se = "boot", boot = nBoot)
## Look at results:
summary(out1.1)
```

```
lavaan (0.5-20) converged normally after  21 iterations

  Number of observations                            87

  Estimator                                         ML
  Minimum Function Test Statistic                0.000
  Degrees of freedom                                 0
  Minimum Function Value               0.0000000000000

Parameter Estimates:

  Information                                 Observed
  Standard Errors                            Bootstrap
  Number of requested bootstrap draws             2500
  Number of successful bootstrap draws            2500
```

Example

```
Regressions:
                   Estimate  Std.Err  Z-value  P(>|z|)
  policy ~
    sysRac     (b1)   0.601    0.138    4.356    0.000
    indRac     (b2)   0.143    0.106    1.344    0.179
    merit      (b3)  -0.036    0.147   -0.246    0.805
    polAffil   (cp)   0.125    0.078    1.616    0.106
  sysRac ~
    polAffil   (a1)   0.170    0.063    2.696    0.007
  indRac ~
    polAffil   (a2)  -0.004    0.078   -0.055    0.956
  merit ~
    polAffil   (a3)  -0.266    0.061   -4.342    0.000

Covariances:
                   Estimate  Std.Err  Z-value  P(>|z|)
  sysRac ~~
    indRac           -0.076    0.100   -0.752    0.452
    merit            -0.217    0.093   -2.334    0.020
  indRac ~~
    merit             0.154    0.098    1.572    0.116

Variances:
```

Example

```
                     Estimate   Std.Err   Z-value   P(>|z|)
    policy              0.963     0.177     5.432     0.000
    sysRac              0.755     0.110     6.866     0.000
    indRac              1.188     0.157     7.565     0.000
    merit               0.719     0.113     6.362     0.000

Defined Parameters:
                     Estimate   Std.Err   Z-value   P(>|z|)
    ab1                 0.102     0.044     2.306     0.021
    ab2                -0.001     0.015    -0.042     0.967
    ab3                 0.010     0.041     0.236     0.813
    totalIE             0.111     0.052     2.151     0.031
```

Example

```
parameterEstimates(out1.1, boot = bootType)[ , -c(1 : 3)]
```

```
     label    est    se      z pvalue ci.lower ci.upper
1       b1  0.601 0.138  4.356  0.000    0.304    0.860
2       b2  0.143 0.106  1.344  0.179   -0.067    0.357
3       b3 -0.036 0.147 -0.246  0.805   -0.295    0.288
4       cp  0.125 0.078  1.616  0.106   -0.036    0.265
5       a1  0.170 0.063  2.696  0.007    0.039    0.290
6       a2 -0.004 0.078 -0.055  0.956   -0.165    0.148
7       a3 -0.266 0.061 -4.342  0.000   -0.386   -0.147
8          -0.076 0.100 -0.752  0.452   -0.276    0.120
9          -0.217 0.093 -2.334  0.020   -0.429   -0.058
10          0.154 0.098  1.572  0.116   -0.037    0.347
11          0.963 0.177  5.432  0.000    0.693    1.421
12          0.755 0.110  6.866  0.000    0.567    1.012
13          1.188 0.157  7.565  0.000    0.920    1.566
14          0.719 0.113  6.362  0.000    0.531    0.980
15          2.444 0.000     NA     NA    2.444    2.444
16     ab1  0.102 0.044  2.306  0.021    0.028    0.208
17     ab2 -0.001 0.015 -0.042  0.967   -0.040    0.025
18     ab3  0.010 0.041  0.236  0.813   -0.081    0.083
19 totalIE  0.111 0.052  2.151  0.031    0.016    0.224
```

Comparing Specific Indirect Effects

When we have multiple specific indirect effects in a single model, we can test if they are statistically different from one another.

QUESTION: How might we go about doing such a test (assuming we're using path modeling)?

Comparing Specific Indirect Effects

When we have multiple specific indirect effects in a single model, we can test if they are statistically different from one another.

QUESTION: How might we go about doing such a test (assuming we're using path modeling)?

ANSWER: There are, at least, two reasonable methods:

1. Use nested model $\Delta\chi^2$ tests
2. Define a new parameter corresponding to the null hypothesis and use bootstrapping

Example

```
## Test differences in specific indirect effects:
mod1.2 <- "
policy ~ b1*sysRac + b2*indRac + b3*merit + cp*polAffil
sysRac ~ a1*polAffil
indRac ~ a2*polAffil
merit ~ a3*polAffil

sysRac ~~ indRac + merit
indRac ~~ merit

ab1 := a1*b1
ab2 := a2*b2
ab3 := a3*b3
totalIE := ab1 + ab2 + ab3

ab1 == ab2
"
out1.2 <-
    sem(mod1.2, data = dat1, se = "boot", boot = nBoot)
summary(out1.2)
```

Example

```
lavaan (0.5-20) converged normally after 278 iterations

  Number of observations                             87

  Estimator                                          ML
  Minimum Function Test Statistic                 6.738
  Degrees of freedom                                  1
  P-value (Chi-square)                            0.009

Parameter Estimates:

  Information                                  Observed
  Standard Errors                             Bootstrap
  Number of requested bootstrap draws              2500
  Number of successful bootstrap draws             2500

Regressions:
                   Estimate  Std.Err  Z-value  P(>|z|)
  policy ~
    sysRac    (b1)    0.575    0.204    2.816    0.005
    indRac    (b2)    0.192    0.130    1.473    0.141
    merit     (b3)   -0.055    0.156   -0.351    0.726
```

Example

```
    polAffil   (cp)     0.125     0.082     1.534     0.125
  sysRac ~
    polAffil   (a1)     0.027     0.067     0.410     0.682
  indRac ~
    polAffil   (a2)     0.082     0.092     0.892     0.372
  merit ~
    polAffil   (a3)    -0.217     0.063    -3.433     0.001

Covariances:
                     Estimate   Std.Err   Z-value   P(>|z|)
  sysRac ~~
    indRac             -0.106     0.109    -0.966     0.334
    merit              -0.234     0.098    -2.396     0.017
  indRac ~~
    merit               0.164     0.100     1.637     0.102

Variances:
                     Estimate   Std.Err   Z-value   P(>|z|)
    policy              0.967     0.182     5.319     0.000
    sysRac              0.804     0.123     6.565     0.000
    indRac              1.206     0.158     7.618     0.000
    merit               0.724     0.112     6.466     0.000
```

Example

```
Defined Parameters:
                   Estimate  Std.Err  Z-value  P(>|z|)
    ab1               0.016    0.019    0.820    0.412
    ab2               0.016    0.019    0.820    0.412
    ab3               0.012    0.037    0.318    0.750
    totalIE           0.043    0.059    0.730    0.465

Constraints:
                                               |Slack|
    ab1 - (ab2)                                  0.000
```

```
## Conduct a chi-squared difference test:
chiDiff ← fitMeasures(out1.2)["chisq"] -
    fitMeasures(out1.1)["chisq"]
dfDiff ← fitMeasures(out1.2)["df"] -
    fitMeasures(out1.1)["df"]
pchisq(chiDiff, dfDiff, lower = FALSE)
```

```
      chisq
0.009440083
```

Example

```
## Test differences in specific indirect effects:
mod1.3 <- "
policy ~ b1*sysRac + b2*indRac + b3*merit + cp*polAffil
sysRac ~ a1*polAffil
indRac ~ a2*polAffil
merit ~ a3*polAffil

sysRac ~~ indRac + merit
indRac ~~ merit

ab1 := a1*b1
ab2 := a2*b2
ab3 := a3*b3
totalIE := ab1 + ab2 + ab3

test1 := ab2 - ab1
"
out1.3 <-
    sem(mod1.3, data = dat1, se = "boot", boot = nBoot)
summary(out1.3)
```

Example

```
lavaan (0.5-20) converged normally after  21 iterations

  Number of observations                            87

  Estimator                                         ML
  Minimum Function Test Statistic                0.000
  Degrees of freedom                                 0
  Minimum Function Value               0.0000000000000

Parameter Estimates:

  Information                                 Observed
  Standard Errors                            Bootstrap
  Number of requested bootstrap draws             2500
  Number of successful bootstrap draws            2499

Regressions:
                   Estimate  Std.Err  Z-value  P(>|z|)
  policy ~
    sysRac    (b1)    0.601    0.141    4.251    0.000
    indRac    (b2)    0.143    0.109    1.315    0.189
    merit     (b3)   -0.036    0.152   -0.239    0.811
```

Example

```
    polAffil   (cp)     0.125     0.077     1.639     0.101
  sysRac ~
    polAffil   (a1)     0.170     0.066     2.599     0.009
  indRac ~
    polAffil   (a2)    -0.004     0.077    -0.055     0.956
  merit ~
    polAffil   (a3)    -0.266     0.060    -4.438     0.000

Covariances:
                   Estimate  Std.Err  Z-value  P(>|z|)
  sysRac ~~
    indRac           -0.076    0.101   -0.747    0.455
    merit            -0.217    0.090   -2.403    0.016
  indRac ~~
    merit             0.154    0.099    1.555    0.120

Variances:
                   Estimate  Std.Err  Z-value  P(>|z|)
    policy            0.963    0.178    5.406    0.000
    sysRac            0.755    0.107    7.043    0.000
    indRac            1.188    0.154    7.708    0.000
    merit             0.719    0.110    6.539    0.000
```

Example

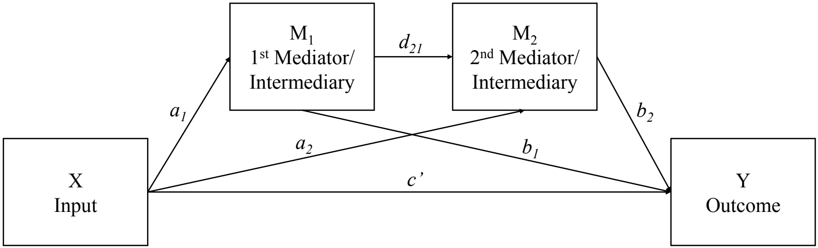
```
Defined Parameters:
                    Estimate   Std.Err   Z-value   P(>|z|)
    ab1                0.102     0.046     2.227     0.026
    ab2               -0.001     0.014    -0.042     0.966
    ab3                0.010     0.041     0.236     0.813
    totalIE            0.111     0.052     2.127     0.033
    test1             -0.103     0.050    -2.074     0.038
```

Example

```
parameterEstimates(out1.3, boot = bootType)[ , -c(1 : 3)]
```

```
     label    est    se      z pvalue ci.lower ci.upper
1       b1  0.601 0.141  4.251  0.000    0.327    0.884
2       b2  0.143 0.109  1.315  0.189   -0.080    0.346
3       b3 -0.036 0.152 -0.239  0.811   -0.324    0.289
4       cp  0.125 0.077  1.639  0.101   -0.027    0.274
5       a1  0.170 0.066  2.599  0.009    0.030    0.288
6       a2 -0.004 0.077 -0.055  0.956   -0.162    0.145
7       a3 -0.266 0.060 -4.438  0.000   -0.396   -0.159
8          -0.076 0.101 -0.747  0.455   -0.287    0.109
9          -0.217 0.090 -2.403  0.016   -0.429   -0.066
10          0.154 0.099  1.555  0.120   -0.033    0.359
11          0.963 0.178  5.406  0.000    0.695    1.461
12          0.755 0.107  7.043  0.000    0.577    1.014
13          1.188 0.154  7.708  0.000    0.926    1.553
14          0.719 0.110  6.539  0.000    0.532    0.969
15          2.444 0.000     NA     NA    2.444    2.444
16     ab1  0.102 0.046  2.227  0.026    0.032    0.225
17     ab2 -0.001 0.014 -0.042  0.966   -0.035    0.026
18     ab3  0.010 0.041  0.236  0.813   -0.079    0.085
19 totalIE  0.111 0.052  2.127  0.033    0.018    0.224
20   test1 -0.103 0.050 -2.074  0.038   -0.222   -0.020
```

Serial Multiple Mediation



Serial Multiple Mediation

To get all of the information in the preceding diagram, we need to fit three equations:

$$
\begin{aligned}
Y &= i_Y + b_1 M_1 + b_2 M_2 + c' X + e_Y \\
M_2 &= i_{M2} + d_{21} M_1 + a_2 X + e_{M2} \\
M_1 &= i_{M1} + a_1 X + e_{M1}
\end{aligned}
$$

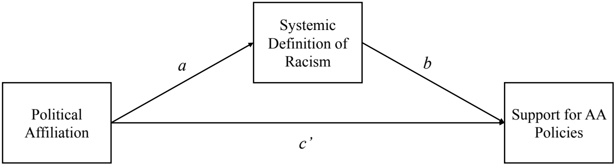
As with parallel mediator models, a serial mediator model with K mediator variables will required $K + 1$ separate equations.

Again, path modeling can make this task much simpler.

- Also allows us to fit more parsimonious, restricted models.

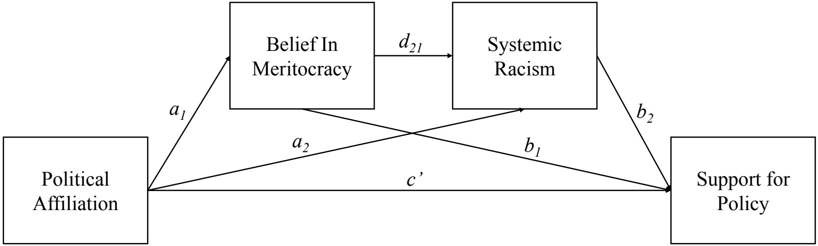
Serial Multiple Mediation

Okay, back to our simple mediation example:



QUESTION: What could be mediating the a path?

POTENTIAL ANSWER:



A Quick Note on Inference

Parallel multiple mediation operates much like a number of combined simple mediation models, serial multiple mediation is not so straight-forward.

In serial multiple mediation:

- Every possible path from X to Y that passes through, at least, one mediator is a specific indirect effect.
  - With the saturated two-mediator model shown above, we have: $IE_{spec} = \{a_1 b_1, a_2 b_2, a_1 d_{21} b_2\}$
- The *Total Indirect Effect* is, again, equal to the sum of all the specific indirect effects: $IE_{tot} = \sum_{k=1}^{|\{IE_{spec}\}|} IE_{spec,k}$.
- The *Total Effect* is equal to the direct effect plus the total indirect effect: $c = c' + IE_{tot}$

A Quick Note on Inference

Inference for the specific indirect effects is basically the same as it is for the sole indirect effect in simple mediation, when using path modeling or bootstrapping.

- CAVEAT: Normal-theory, Sobel-Type, standard errors for the specific indirect effects that involve more than two constituent paths can be very complex.
  - This isn't really a problem since you should always use bootstrapping, anyway!

Example

```r
## Serial Multiple Mediator Model:
mod2.1 <- "
policy ~ b1*merit + b2*sysRac + cp*polAffil
sysRac ~ d21*merit + a2*polAffil
merit ~ a1*polAffil

ab1 := a1*b1
ab2 := a2*b2
fullIE := a1*d21*b2
totalIE := ab1 + ab2 + fullIE
"
out2.1 <-
    sem(mod2.1, data = dat1, se = "boot", boot = nBoot)
summary(out2.1)
```

Example

```
lavaan (0.5-20) converged normally after  16 iterations

  Number of observations                            87

  Estimator                                         ML
  Minimum Function Test Statistic                0.000
  Degrees of freedom                                 0

Parameter Estimates:

  Information                                 Observed
  Standard Errors                            Bootstrap
  Number of requested bootstrap draws             2500
  Number of successful bootstrap draws            2499

Regressions:
                   Estimate  Std.Err  Z-value  P(>|z|)
  policy ~
    merit     (b1)   -0.008    0.145   -0.052    0.959
    sysRac    (b2)    0.595    0.142    4.184    0.000
    polAffil  (cp)    0.134    0.076    1.763    0.078
  sysRac ~
```

Example

```
    merit     (d21)   -0.301    0.110   -2.733    0.006
    polAffil   (a2)    0.090    0.072    1.253    0.210
  merit ~
    polAffil   (a1)   -0.266    0.061   -4.384    0.000

Variances:
                   Estimate  Std.Err  Z-value  P(>|z|)
    policy            0.987    0.164    6.013    0.000
    sysRac            0.689    0.094    7.309    0.000
    merit             0.719    0.112    6.389    0.000

Defined Parameters:
                   Estimate  Std.Err  Z-value  P(>|z|)
    ab1               0.002    0.040    0.050    0.960
    ab2               0.053    0.044    1.215    0.225
    fullIE            0.048    0.026    1.822    0.068
    totalIE           0.103    0.048    2.145    0.032
```

Example

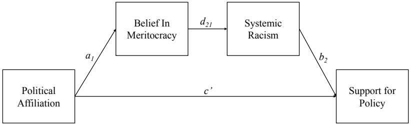
```
parameterEstimates(out2.1, boot = bootType)[ , -c(1 : 3)]
```

```
     label    est    se      z pvalue ci.lower ci.upper
1       b1 -0.008 0.145 -0.052  0.959   -0.286    0.275
2       b2  0.595 0.142  4.184  0.000    0.317    0.861
3       cp  0.134 0.076  1.763  0.078   -0.019    0.281
4      d21 -0.301 0.110 -2.733  0.006   -0.508   -0.076
5       a2  0.090 0.072  1.253  0.210   -0.073    0.220
6       a1 -0.266 0.061 -4.384  0.000   -0.384   -0.148
7           0.987 0.164  6.013  0.000    0.733    1.390
8           0.689 0.094  7.309  0.000    0.537    0.919
9           0.719 0.112  6.389  0.000    0.535    0.980
10          2.444 0.000     NA     NA    2.444    2.444
11     ab1  0.002 0.040  0.050  0.960   -0.080    0.081
12     ab2  0.053 0.044  1.215  0.225   -0.031    0.146
13  fullIE  0.048 0.026  1.822  0.068    0.012    0.117
14 totalIE  0.103 0.048  2.145  0.032    0.011    0.202
```

Restricted Models

In this example, the a_2 and b_1 paths are non-significant as are the simple specific indirect effects $a_1 b_1$ and $a_2 b_2$.

- There is a school of thinking that would prescribe constraining the a_2 and b_1 paths to zero as in:



- This model will ascribe a larger effect size to $a_1 d_{21} b_2$ since it must convey all of the indirect influence of X on Y.
  - We should first fit a saturated model, but subsequently culling non-significant paths can, sometimes, be appropriate.

Example

```
mod2.2 <- "
policy ~ cp*polAffil + b2*sysRac
merit ~ a1*polAffil
sysRac ~ d21*merit

fullIE := a1*d21*b2
"
out2.2 <-
    sem(mod2.2, data = dat1, se = "boot", boot = nBoot)
summary(out2.2)
```

```
lavaan (0.5-20) converged normally after  13 iterations

  Number of observations                             87

  Estimator                                          ML
  Minimum Function Test Statistic                 1.991
  Degrees of freedom                                  2
  P-value (Chi-square)                            0.370

Parameter Estimates:
```

Example

```
  Information                                    Observed
  Standard Errors                               Bootstrap
  Number of requested bootstrap draws                2500
  Number of successful bootstrap draws               2500

Regressions:
                   Estimate  Std.Err  Z-value  P(>|z|)
  policy ~
    polAffil  (cp)    0.135    0.083    1.638    0.102
    sysRac    (b2)    0.597    0.137    4.359    0.000
  merit ~
    polAffil  (a1)   -0.266    0.061   -4.353    0.000
  sysRac ~
    merit    (d21)   -0.367    0.098   -3.764    0.000

Variances:
                   Estimate  Std.Err  Z-value  P(>|z|)
    policy            0.987    0.166    5.946    0.000
    merit             0.719    0.116    6.218    0.000
    sysRac            0.705    0.094    7.520    0.000

Defined Parameters:
                   Estimate  Std.Err  Z-value  P(>|z|)
```

Example

```
fullIE                    0.058     0.025     2.318     0.020
```

Example

```
parameterEstimates(out2.2, boot = bootType)[ , -c(1 : 3)]
```

```
   label    est    se      z pvalue ci.lower ci.upper
1     cp  0.135 0.083  1.638  0.102   -0.031    0.286
2     b2  0.597 0.137  4.359  0.000    0.311    0.858
3     a1 -0.266 0.061 -4.353  0.000   -0.392   -0.148
4    d21 -0.367 0.098 -3.764  0.000   -0.546   -0.166
5         0.987 0.166  5.946  0.000    0.731    1.402
6         0.719 0.116  6.218  0.000    0.527    0.991
7         0.705 0.094  7.520  0.000    0.552    0.926
8         2.444 0.000     NA     NA    2.444    2.444
9 fullIE  0.058 0.025  2.318  0.020    0.019    0.123
```

Example

As in parallel multiple mediation, we can test for differences in the specific indirect effects of a serial multiple mediator model:

```
## Test Differences between Indirect Effects
## in Serial Multiple Mediator Model (Method 1):
mod2.3 <- "
policy ~ cp*polAffil + b1*merit + b2*sysRac
merit ~ a1*polAffil
sysRac ~ a2*polAffil + d21*merit

ab1 := a1*b1
ab2 := a2*b2
fullIE := a1*d21*b2
totalIE := ab1 + ab2 + fullIE

fullIE == ab1
fullIE == ab2
"
out2.3 <-
    sem(mod2.3, data = dat1, se = "boot", boot = nBoot)
summary(out2.3)
```

Example

```
lavaan (0.5-20) converged normally after 213 iterations

  Number of observations                            87

  Estimator                                         ML
  Minimum Function Test Statistic                1.334
  Degrees of freedom                                 2
  P-value (Chi-square)                           0.513

Parameter Estimates:

  Information                                 Observed
  Standard Errors                            Bootstrap
  Number of requested bootstrap draws             2500
  Number of successful bootstrap draws            2500

Regressions:
                   Estimate  Std.Err  Z-value  P(>|z|)
  policy ~
    polAffil  (cp)    0.108    0.084    1.281    0.200
    merit     (b1)   -0.150    0.047   -3.183    0.001
    sysRac    (b2)    0.521    0.125    4.157    0.000
```

Example

```
  merit ~
    polAffil   (a1)    -0.271     0.057    -4.750     0.000
  sysRac ~
    polAffil   (a2)     0.078     0.025     3.125     0.002
    merit     (d21)    -0.287     0.075    -3.814     0.000

Variances:
                   Estimate   Std.Err   Z-value   P(>|z|)
    policy            1.001     0.171     5.854     0.000
    merit             0.719     0.114     6.330     0.000
    sysRac            0.690     0.090     7.632     0.000

Defined Parameters:
                   Estimate   Std.Err   Z-value   P(>|z|)
    ab1               0.041     0.014     2.983     0.003
    ab2               0.041     0.014     2.983     0.003
    fullIE            0.041     0.014     2.983     0.003
    totalIE           0.122     0.041     2.983     0.003

Constraints:
                                                  |Slack|
    fullIE - (ab1)                                  0.000
    fullIE - (ab2)                                  0.000
```

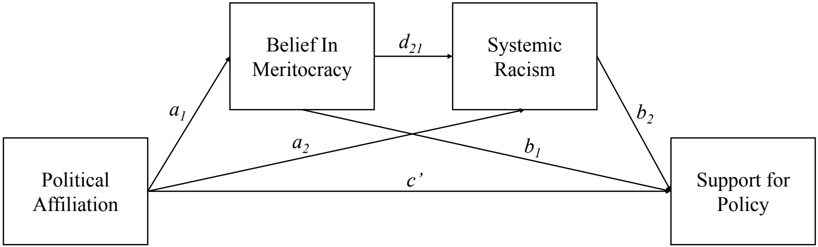
Example

```
## Conduct a chi-squared difference test:
chiDiff <- fitMeasures(out2.3)["chisq"] -
    fitMeasures(out2.1)["chisq"]
dfDiff <- fitMeasures(out2.3)["df"] -
    fitMeasures(out2.1)["df"]
pchisq(chiDiff, dfDiff, lower = FALSE)
```

```
    chisq
0.5131246
```

Serial Multiple Mediation

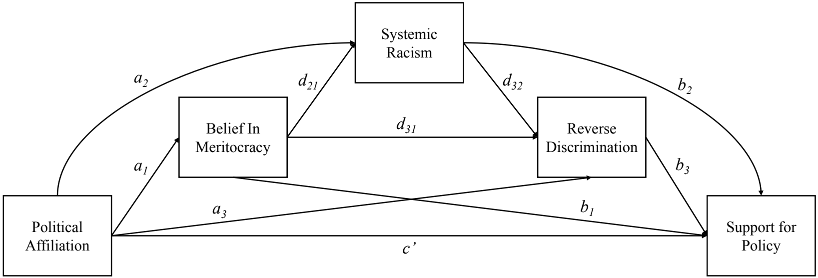
Okay, so we've supported an interesting hypothesis with the following model, but why stop there?



QUESTION: What might mediated the b_2 path?

Serial Multiple Mediation

Potential Answer:



Serial Multiple Mediation

QUESTION: How many equations do we need to get the information in the preceding diagram?

Serial Multiple Mediation

QUESTION: How many equations do we need to get the information in the preceding diagram?

$$
\begin{aligned}
Policy &= i_Y + b_1 Merit + b_2 SysRac + b_3 RevDisc + c' PolAff + e_Y \\
RevDisc &= i_{M3} + d_{31} Merit + d_{32} SysRac + a_3 PolAff + e_{M3} \\
SysRac &= i_{M2} + d_{21} Merit + a_2 PolAff + e_{M2} \\
Merit &= i_{M1} + a_1 PolAff + e_{M1}
\end{aligned}
$$

Which produces the following set of specific indirect effects:

- $a_1 b_1$
- $a_2 b_2$
- $a_3 b_3$
- $a_1 d_{31} b_3$
- $a_1 d_{21} b_2$
- $a_2 d_{32} b_3$
- $a_1 d_{21} d_{32} b_3$

Example

```
## Serial Multiple Mediator Model with 3 Mediators:
mod3.1 <- "
policy ~ b1*merit + b2*sysRac + b3*revDisc + cp*polAffil
revDisc ~ d31*merit + d32*sysRac + a3*polAffil
sysRac ~ d21*merit + a2*polAffil
merit ~ a1*polAffil

ab1 := a1*b1
ab2 := a2*b2
ab3 := a3*b3

partIE1 := a1*d31*b3
partIE2 := a1*d21*b2
partIE3 := a2*d32*b3

fullIE := a1*d21*d32*b3

totalIE := ab1 + ab2 + ab3 + partIE1 + partIE2 + partIE3 +
    fullIE
"
out3.1 <-
    sem(mod3.1, data = dat1, se = "boot", boot = nBoot)
```

Example

```
summary(out3.1)
```

```
lavaan (0.5-20) converged normally after  23 iterations

  Number of observations                            87

  Estimator                                         ML
  Minimum Function Test Statistic                0.000
  Degrees of freedom                                 0

Parameter Estimates:

  Information                                 Observed
  Standard Errors                            Bootstrap
  Number of requested bootstrap draws             2500
  Number of successful bootstrap draws            2498

Regressions:
                   Estimate  Std.Err  Z-value  P(>|z|)
  policy ~
    merit     (b1)    0.005    0.144    0.035    0.972
    sysRac    (b2)    0.589    0.151    3.895    0.000
```

Example

```
    revDisc    (b3)   -0.026    0.080   -0.330    0.741
    polAffil   (cp)    0.130    0.080    1.616    0.106
  revDisc ~
    merit     (d31)    0.473    0.190    2.490    0.013
    sysRac    (d32)   -0.196    0.243   -0.806    0.420
    polAffil   (a3)   -0.149    0.131   -1.140    0.254
  sysRac ~
    merit     (d21)   -0.301    0.109   -2.765    0.006
    polAffil   (a2)    0.090    0.071    1.270    0.204
  merit ~
    polAffil   (a1)   -0.266    0.061   -4.340    0.000

Variances:
                   Estimate  Std.Err  Z-value  P(>|z|)
    policy            0.985    0.164    6.023    0.000
    revDisc           2.361    0.307    7.698    0.000
    sysRac            0.689    0.091    7.612    0.000
    merit             0.719    0.111    6.482    0.000

Defined Parameters:
                   Estimate  Std.Err  Z-value  P(>|z|)
    ab1              -0.001    0.040   -0.033    0.973
    ab2               0.053    0.043    1.224    0.221
```

Example

```
ab3          0.004  0.016  0.244  0.807
partIE1      0.003  0.012  0.273  0.785
partIE2      0.047  0.026  1.831  0.067
partIE3      0.000  0.003  0.150  0.881
fullIE       0.000  0.002  0.191  0.849
totalIE      0.107  0.052  2.052  0.040
```

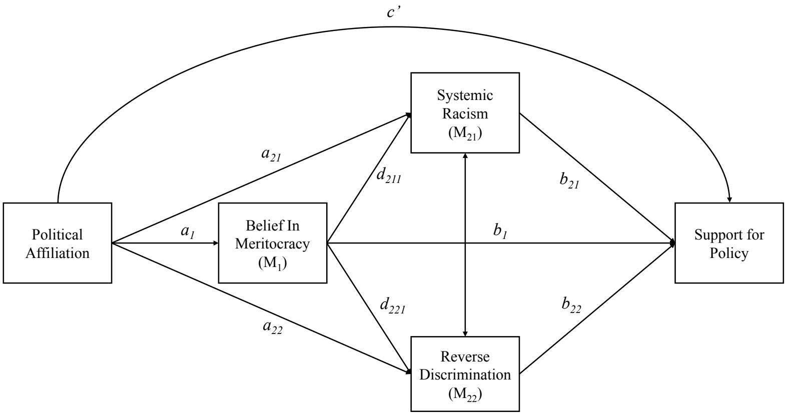
Example

```
parameterEstimates(out3.1, boot = bootType)[ , -c(1 : 3)]
```

	label	est	se	z	pvalue	ci.lower	ci.upper
1	b1	0.005	0.144	0.035	0.972	-0.270	0.299
2	b2	0.589	0.151	3.895	0.000	0.281	0.877
3	b3	-0.026	0.080	-0.330	0.741	-0.198	0.121
4	cp	0.130	0.080	1.616	0.106	-0.036	0.283
5	d31	0.473	0.190	2.490	0.013	0.073	0.827
6	d32	-0.196	0.243	-0.806	0.420	-0.660	0.306
7	a3	-0.149	0.131	-1.140	0.254	-0.398	0.099
8	d21	-0.301	0.109	-2.765	0.006	-0.506	-0.080
9	a2	0.090	0.071	1.270	0.204	-0.052	0.227
10	a1	-0.266	0.061	-4.340	0.000	-0.383	-0.147
11		0.985	0.164	6.023	0.000	0.740	1.443
12		2.361	0.307	7.698	0.000	1.869	3.106
13		0.689	0.091	7.612	0.000	0.549	0.907
14		0.719	0.111	6.482	0.000	0.536	0.981
15		2.444	0.000	NA	NA	2.444	2.444
16	ab1	-0.001	0.040	-0.033	0.973	-0.085	0.075
17	ab2	0.053	0.043	1.224	0.221	-0.022	0.155
18	ab3	0.004	0.016	0.244	0.807	-0.014	0.062
19	partIE1	0.003	0.012	0.273	0.785	-0.013	0.038
20	partIE2	0.047	0.026	1.831	0.067	0.011	0.115
21	partIE3	0.000	0.003	0.150	0.881	-0.001	0.018
22	fullIE	0.000	0.002	0.191	0.849	-0.002	0.009
23	totalIE	0.107	0.052	2.052	0.040	0.010	0.213

Hybrid Multiple Mediation

We can also combine parallel and serial mediation models:



Example

```
## Hybrid Multiple Mediator Model:
mod4.1 <- "
policy ~ b1*merit + b21*sysRac + b22*revDisc + cp*polAffil
sysRac ~ d211*merit + a21*polAffil
revDisc ~ d221*merit + a22*polAffil
merit ~ a1*polAffil

sysRac ~~ revDisc

ab1 := a1*b1
ab21 := a21*b21
ab22 := a22*b22

fullIE21 := a1*d211*b21
fullIE22 := a1*d221*b22

totalIE := ab1 + ab21 + ab22 + fullIE21 + fullIE22
"
out4.1 <-
    sem(mod4.1, data = dat1, se = "boot", boot = nBoot)
summary(out4.1)
```

Example

```
lavaan (0.5-20) converged normally after  22 iterations

  Number of observations                            87

  Estimator                                         ML
  Minimum Function Test Statistic                0.000
  Degrees of freedom                                 0
  Minimum Function Value               0.0000000000000

Parameter Estimates:

  Information                                 Observed
  Standard Errors                            Bootstrap
  Number of requested bootstrap draws             2500
  Number of successful bootstrap draws            2499

Regressions:
                   Estimate  Std.Err  Z-value  P(>|z|)
  policy ~
    merit     (b1)    0.005    0.142    0.035    0.972
    sysRac   (b21)    0.589    0.150    3.924    0.000
    revDisc  (b22)   -0.026    0.080   -0.328    0.743
```

Example

```
    polAffl    (cp)     0.130     0.080     1.627     0.104
  sysRac ~
    merit    (d211)    -0.301     0.111    -2.721     0.006
    polAffl   (a21)     0.090     0.070     1.282     0.200
  revDisc ~
    merit    (d221)     0.532     0.192     2.777     0.005
    polAffl   (a22)    -0.167     0.138    -1.214     0.225
  merit ~
    polAffl    (a1)    -0.266     0.061    -4.337     0.000

Covariances:
                   Estimate   Std.Err   Z-value   P(>|z|)
  sysRac ~~
    revDisc            -0.135     0.157    -0.859     0.390

Variances:
                   Estimate   Std.Err   Z-value   P(>|z|)
    policy              0.985     0.162     6.098     0.000
    sysRac              0.689     0.092     7.463     0.000
    revDisc             2.388     0.303     7.881     0.000
    merit               0.719     0.113     6.378     0.000

Defined Parameters:
```

Example

	Estimate	Std.Err	Z-value	P(>|z|)
ab1	-0.001	0.039	-0.034	0.973
ab21	0.053	0.043	1.247	0.212
ab22	0.004	0.018	0.251	0.801
fullIE21	0.047	0.025	1.871	0.061
fullIE22	0.004	0.014	0.276	0.783
totalIE	0.107	0.052	2.047	0.041

Example

```
parameterEstimates(out4.1, boot = bootType)[ , -c(1 : 3)]
```

	label	est	se	z	pvalue	ci.lower	ci.upper
1	b1	0.005	0.142	0.035	0.972	-0.266	0.285
2	b21	0.589	0.150	3.924	0.000	0.301	0.891
3	b22	-0.026	0.080	-0.328	0.743	-0.186	0.129
4	cp	0.130	0.080	1.627	0.104	-0.029	0.283
5	d211	-0.301	0.111	-2.721	0.006	-0.520	-0.085
6	a21	0.090	0.070	1.282	0.200	-0.045	0.222
7	d221	0.532	0.192	2.777	0.005	0.137	0.899
8	a22	-0.167	0.138	-1.214	0.225	-0.447	0.099
9	a1	-0.266	0.061	-4.337	0.000	-0.396	-0.153
10		-0.135	0.157	-0.859	0.390	-0.459	0.165
11		0.985	0.162	6.098	0.000	0.750	1.477
12		0.689	0.092	7.463	0.000	0.538	0.904
13		2.388	0.303	7.881	0.000	1.894	3.158
14		0.719	0.113	6.378	0.000	0.535	1.003
15		2.444	0.000	NA	NA	2.444	2.444
16	ab1	-0.001	0.039	-0.034	0.973	-0.081	0.074
17	ab21	0.053	0.043	1.247	0.212	-0.019	0.151
18	ab22	0.004	0.018	0.251	0.801	-0.018	0.064
19	fullIE21	0.047	0.025	1.871	0.061	0.013	0.123
20	fullIE22	0.004	0.014	0.276	0.783	-0.015	0.042
21	totalIE	0.107	0.052	2.047	0.041	0.005	0.218

Practice

List all of the specific indirect effects present in this model: