Multiple Mediation

Theory Construction and Statistical Modeling

Kyle M. Lang

Department of Methodology & Statistics
Utrecht University

Outline

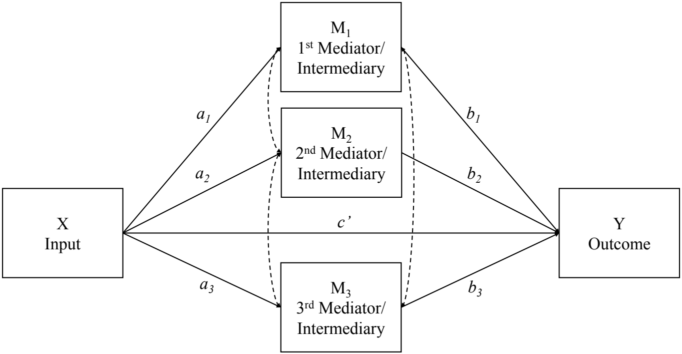
Simple Mediation is Too Simple

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 Mediators

Parallel Multiple Mediation



Parallel Multiple Mediation

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

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

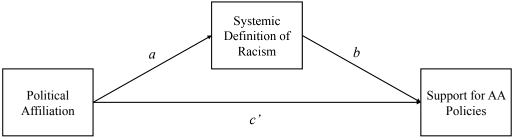
In general, a parallel mediator model with K mediator variables will required $K + 1$ separate equations.

Path modeling can make this task much simpler.

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

Parallel Multiple Mediation

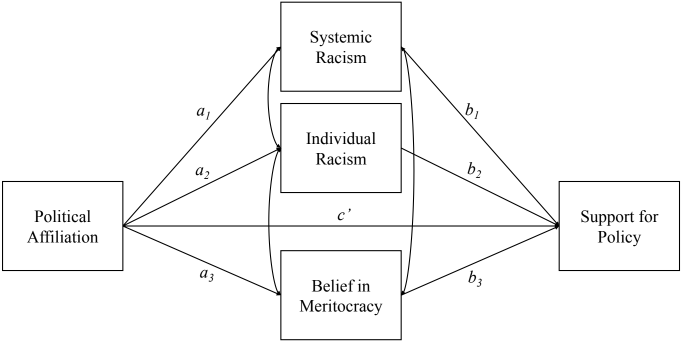
Let's reconsider the last example:



QUESTION: What might be mediating the residual direct effect?

Parallel Multiple Mediation

POTENTIAL ANSWER:



A Quick Note on Inference

In parallel multiple mediation:

- We have K *specific indirect effects*, where K is the number of mediators: $a_1b_1, a_2b_2, \ldots, a_Kb_K$.
- The *Total Indirect Effect* is equal to the sum of all the specific indirect effects: $IE_{tot} = \sum_{k=1}^{K} a_kb_k$.
- 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

```
## Read in the data
dat1 <- readRDS("../data/adamsKlpsScaleScore.rds")

## Parallel Multiple Mediator Model:
mod1.1 <- '
policy ~ 1 + b1*sysRac + b2*indRac + b3*merit + cp*polAffil
sysRac ~ 1 + a1*polAffil
indRac ~ 1 + a2*polAffil
merit  ~ 1 + a3*polAffil

sysRac ~~ indRac + merit
indRac ~~ merit

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

nSams <- 500 # No. of bootstrap resamples

## Fit the model:
out1.1 <- se
```

Example

```
## Look at results:
partSummary(out1.1, 7)

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  policy ~
    sysRac    (b1)    0.601    0.143    4.203    0.000
    indRac    (b2)    0.143    0.112    1.277    0.202
    merit     (b3)   -0.036    0.156   -0.233    0.816
    polAffil  (cp)    0.125    0.084    1.501    0.133
  sysRac ~
    polAffil  (a1)    0.170    0.062    2.751    0.006
  indRac ~
    polAffil  (a2)   -0.004    0.081   -0.052    0.958
  merit ~
    polAffil  (a3)   -0.266    0.062   -4.274    0.000
```

Example

```
partSummary(out1.1, 9)

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .policy            0.490    0.903    0.543    0.587
   .sysRac            3.197    0.267   11.960    0.000
   .indRac            3.398    0.336   10.122    0.000
   .merit             4.977    0.273   18.251    0.000
```

Example

```
partSummary(out1.1, c(10, 8))

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .policy            0.963    0.174    5.527    0.000
   .sysRac            0.755    0.110    6.838    0.000
   .indRac            1.188    0.151    7.891    0.000
   .merit             0.719    0.118    6.088    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
 .sysRac ~~
   .indRac           -0.076    0.108   -0.701    0.483
   .merit            -0.217    0.096   -2.249    0.024
 .indRac ~~
   .merit             0.154    0.096    1.606    0.108
```

Example

```
partSummary(out1.1, 11)

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)
    ab1               0.102    0.041    2.467    0.014
    ab2              -0.001    0.016   -0.038    0.970
    ab3               0.010    0.043    0.225    0.822
    totalIE           0.111    0.052    2.142    0.032
```

Example

```
parameterEstimates(out1.1, boot.ci.type = "bca.simple") %>%
    select(c("label", "est", "ci.lower", "ci.upper")) %>%
    tail(4)

     label    est ci.lower ci.upper
21     ab1  0.102    0.038    0.216
22     ab2 -0.001   -0.044    0.029
23     ab3  0.010   -0.095    0.081
24 totalIE  0.111    0.011    0.230
```

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 to encode the constraint and use bootstrapping

Example

```
## Test differences in specific indirect effects:
mod1.2 <- '
policy ~ 1 + b1*sysRac + b2*indRac + b3*merit + cp*polAffil
sysRac ~ 1 + a1*polAffil
indRac ~ 1 + a2*polAffil
merit  ~ 1 + a3*polAffil

sysRac ~~ indRac + merit
indRac ~~ merit

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

ab1 == ab2 # The first two IEs are constrained to equality
'

out1.2 <- sem(mod1.2, data = dat1)
```

Example

```
## Look at results:
partSummary(out1.2, 7)

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  policy ~
    sysRac    (b1)    0.575    0.123    4.662    0.000
    indRac    (b2)    0.192    0.096    2.004    0.045
    merit     (b3)   -0.055    0.131   -0.416    0.678
    polAffil  (cp)    0.125    0.074    1.696    0.090
  sysRac ~
    polAffil  (a1)    0.027    0.025    1.082    0.279
  indRac ~
    polAffil  (a2)    0.082    0.067    1.222    0.222
  merit ~
    polAffil  (a3)   -0.217    0.055   -3.943    0.000
```

Example

```
partSummary(out1.2, 9)

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .policy            0.496    0.965    0.514    0.607
   .sysRac            3.813    0.146   26.178    0.000
   .indRac            3.025    0.313    9.668    0.000
   .merit             4.766    0.254   18.730    0.000
```

Example

```
partSummary(out1.2, c(10, 8))

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .policy            0.967    0.147    6.595    0.000
   .sysRac            0.804    0.122    6.595    0.000
   .indRac            1.206    0.183    6.595    0.000
   .merit             0.724    0.110    6.595    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  .sysRac ~~
   .indRac           -0.106    0.106   -0.995    0.320
   .merit            -0.234    0.086   -2.731    0.006
  .indRac ~~
   .merit             0.164    0.102    1.615    0.106
```

Example

```
partSummary(out1.2, 11)

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)
    ab1               0.016    0.014    1.093    0.274
    ab2               0.016    0.014    1.093    0.274
    ab3               0.012    0.029    0.412    0.680
    totalIE           0.043    0.042    1.038    0.299
```

Example

```
## 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
0.009440087
```

Example

```
## Same test as above using bootstrapping:
mod1.3 <- '
policy ~ 1 + b1*sysRac + b2*indRac + b3*merit + cp*polAffil
sysRac ~ 1 + a1*polAffil
indRac ~ 1 + a2*polAffil
merit  ~ 1 + 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", bootstrap = nSams)
```

Example

```
## Look at results:
partSummary(out1.3, 7)

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  policy ~
    sysRac    (b1)    0.601    0.128    4.678    0.000
    indRac    (b2)    0.143    0.102    1.403    0.160
    merit     (b3)   -0.036    0.149   -0.243    0.808
    polAffil  (cp)    0.125    0.077    1.633    0.102
  sysRac ~
    polAffil  (a1)    0.170    0.064    2.663    0.008
  indRac ~
    polAffil  (a2)   -0.004    0.079   -0.054    0.957
  merit ~
    polAffil  (a3)   -0.266    0.056   -4.730    0.000
```

Example

```
partSummary(out1.3, 9)

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .policy            0.490    0.872    0.562    0.574
   .sysRac            3.197    0.286   11.194    0.000
   .indRac            3.398    0.327   10.391    0.000
   .merit             4.977    0.246   20.228    0.000
```

Example

```
partSummary(out1.3, c(10, 8))

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .policy            0.963    0.168    5.736    0.000
   .sysRac            0.755    0.112    6.758    0.000
   .indRac            1.188    0.147    8.103    0.000
   .merit             0.719    0.112    6.430    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
 .sysRac ~~
   .indRac           -0.076    0.101   -0.750    0.453
   .merit            -0.217    0.091   -2.370    0.018
 .indRac ~~
   .merit             0.154    0.097    1.594    0.111
```

Example

```
partSummary(out1.3, 11)

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)
    ab1               0.102    0.044    2.332    0.020
    ab2              -0.001    0.015   -0.042    0.967
    ab3               0.010    0.041    0.238    0.812
    totalIE           0.111    0.052    2.143    0.032
    test1            -0.103    0.047   -2.197    0.028
```

Example

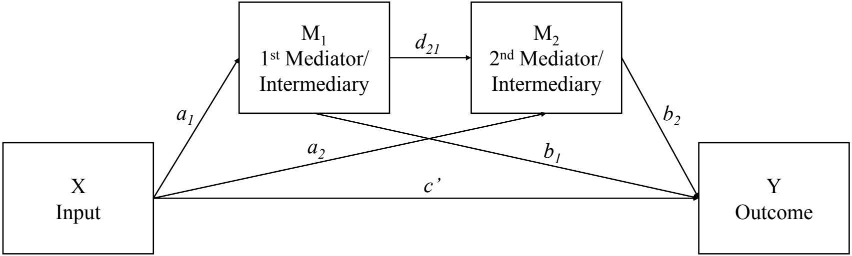
```
parameterEstimates(out1.3, boot.ci.type = "bca.simple") %>%
    select(c("label", "est", "ci.lower", "ci.upper")) %>%
    tail(5)

Warning in norm.inter(t, adj.alpha):  extreme order statistics used as
endpoints

     label    est ci.lower ci.upper
21     ab1  0.102    0.031    0.210
22     ab2 -0.001   -0.037    0.027
23     ab3  0.010   -0.073    0.094
24 totalIE  0.111    0.024    0.242
25   test1 -0.103   -0.214   -0.026
```

Serial Mediators

Serial Multiple Mediation



Serial Multiple Mediation

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

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

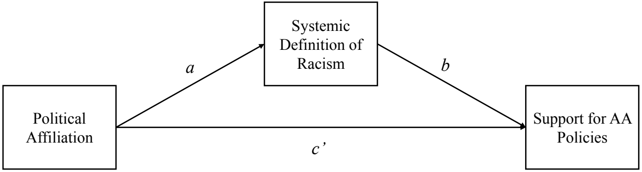
As with parallel mediator models, a serial mediator model with K mediator variables will required $K + 1$ separate equations.

Again, path modeling can make this task much simpler.

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

Serial Multiple Mediation

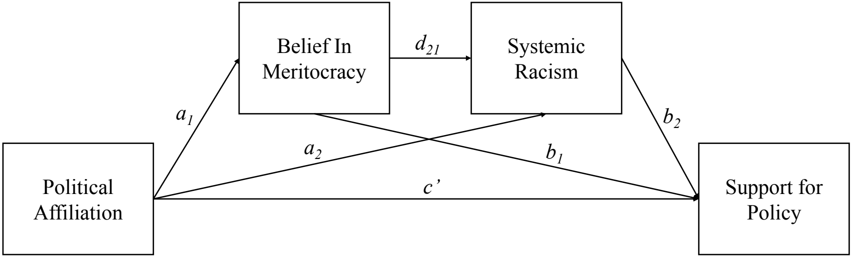
OK, back to our simple mediation example:



QUESTION: What could be mediating the a path?

Serial Multiple Mediation

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: $IE_{spec} = \{a_1b_1, a_2b_2, a_1d_{21}b_2\}$
- 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.

- 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 ~ 1 + b1*merit + b2*sysRac + cp*polAffil
sysRac ~ 1 + d21*merit + a2*polAffil
merit  ~ 1 + 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", bootstrap = nSams)
```

Example

```
## Check the results:
partSummary(out2.1, 7)

Regressions:
                     Estimate  Std.Err  z-value  P(>|z|)
  policy ~
    merit      (b1)    -0.008    0.145   -0.052    0.959
    sysRac     (b2)     0.595    0.147    4.057    0.000
    polAffil   (cp)     0.134    0.077    1.728    0.084
  sysRac ~
    merit     (d21)    -0.301    0.114   -2.652    0.008
    polAffil   (a2)     0.090    0.071    1.265    0.206
  merit ~
    polAffil   (a1)    -0.266    0.060   -4.450    0.000
```

Example

```
partSummary(out2.1, 8:9)

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .policy            0.851    0.899    0.947    0.344
   .sysRac            4.698    0.655    7.168    0.000
   .merit             4.977    0.261   19.039    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .policy            0.987    0.166    5.947    0.000
   .sysRac            0.689    0.091    7.569    0.000
   .merit             0.719    0.108    6.629    0.000
```

Example

```
partSummary(out2.1, 10)

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)
    ab1               0.002    0.040    0.050    0.960
    ab2               0.053    0.045    1.198    0.231
    fullIE            0.048    0.026    1.837    0.066
    totalIE           0.103    0.049    2.112    0.035
```

Example

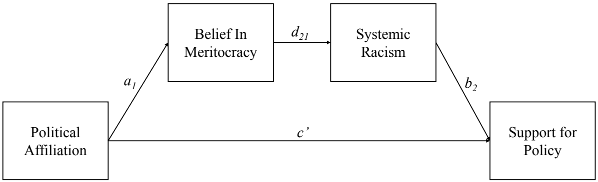
```
parameterEstimates(out2.1, boot.ci.type = "bca.simple") %>%
    select(c("label", "est", "ci.lower", "ci.upper")) %>%
    tail(4)

     label   est ci.lower ci.upper
15     ab1 0.002   -0.079    0.079
16     ab2 0.053   -0.031    0.152
17  fullIE 0.048    0.014    0.126
18 totalIE 0.103    0.016    0.229
```

Restricted Models

In the preceding example, the a_2 and b_1 paths and the specific indirect effects a_1b_1 and a_2b_2 were all non-significant.

- 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 $a_1d_{21}b_2$ since it must convey all of the indirect influence of X on Y.

Example

```
mod2.2 <- '
policy ~ 1 + cp*polAffil + b2*sysRac
merit  ~ 1 + a1*polAffil
sysRac ~ 1 + d21*merit

fullIE := a1*d21*b2
'

out2.2 <- sem(mod2.2, data = dat1, se = "boot", bootstrap = nSams)
```

Example

```
partSummary(out2.2, 7:8)

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  policy ~
    polAffil  (cp)    0.135    0.082    1.648    0.099
    sysRac    (b2)    0.597    0.139    4.307    0.000
  merit ~
    polAffil  (a1)   -0.266    0.060   -4.473    0.000
  sysRac ~
    merit    (d21)   -0.367    0.098   -3.730    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .policy            0.807    0.575    1.403    0.161
   .merit             4.977    0.254   19.573    0.000
   .sysRac            5.337    0.399   13.384    0.000
```

Example

```
partSummary(out2.2, 9:10)

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .policy            0.987    0.163    6.044    0.000
   .merit             0.719    0.112    6.438    0.000
   .sysRac            0.705    0.095    7.424    0.000

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)
    fullIE            0.058    0.026    2.286    0.022
```

Example

```
parameterEstimates(out2.2, boot.ci.type = "bca.simple") %>%
    select(c("label", "est", "ci.lower", "ci.upper")) %>%
    filter(label != "")

   label    est ci.lower ci.upper
1     cp  0.135   -0.025    0.286
2     b2  0.597    0.325    0.864
3     a1 -0.266   -0.379   -0.141
4    d21 -0.367   -0.567   -0.169
5 fullIE  0.058    0.021    0.121
```

Example

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

```
mod2.3 <- '
policy ~ 1 + cp*polAffil + b1*merit + b2*sysRac
merit  ~ 1 + a1*polAffil
sysRac ~ 1 + 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)
```

Example

```
partSummary(out2.3, 7:8)

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  policy ~
    polAffil  (cp)    0.108    0.074    1.469    0.142
    merit     (b1)   -0.150    0.046   -3.243    0.001
    sysRac    (b2)    0.521    0.113    4.624    0.000
  merit ~
    polAffil  (a1)   -0.271    0.057   -4.769    0.000
  sysRac ~
    polAffil  (a2)    0.078    0.023    3.364    0.001
    merit    (d21)   -0.287    0.073   -3.925    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .policy            1.794    0.435    4.124    0.000
   .merit             4.998    0.261   19.122    0.000
   .sysRac            4.695    0.237   19.826    0.000
```

Example

```
partSummary(out2.3, 9:10)

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .policy            1.001    0.152    6.595    0.000
   .merit             0.719    0.109    6.595    0.000
   .sysRac            0.690    0.105    6.595    0.000

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)
    ab1               0.041    0.014    2.873    0.004
    ab2               0.041    0.014    2.873    0.004
    fullIE            0.041    0.014    2.873    0.004
    totalIE           0.122    0.042    2.873    0.004
```

Example

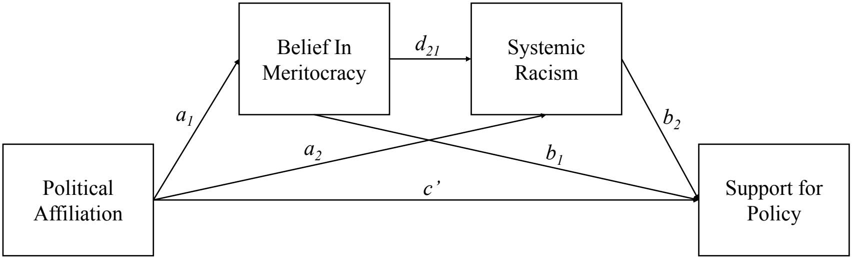
```r
## Conduct a chi-squared difference test:
chiDiff <- fitMeasures(out2.3)["chisq"] - fitMeasures(out2.1)["chisq"]
dfDiff  <- fitMeasures(out2.3)["df"] - fitMeasures(out2.1)["df"]

pchisq(chiDiff, dfDiff, lower = FALSE)

    chisq
0.5131246
```

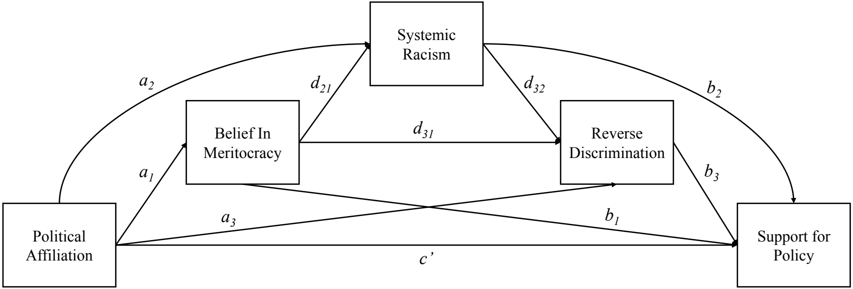
Serial Multiple Mediation

OK. We’ve supported an interesting hypothesis with the following model, but why stop there?



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  ~ 1 + b1*merit + b2*sysRac + b3*revDisc + cp*polAffil
revDisc ~ 1 + d31*merit + d32*sysRac + a3*polAffil
sysRac  ~ 1 + d21*merit + a2*polAffil
merit   ~ 1 + 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", bootstrap = nSams)
```

Example

```
partSummary(out3.1, 7)

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  policy ~
    merit     (b1)    0.005    0.135    0.037    0.971
    sysRac    (b2)    0.589    0.149    3.947    0.000
    revDisc   (b3)   -0.026    0.077   -0.344    0.731
    polAffil  (cp)    0.130    0.076    1.700    0.089
  revDisc ~
    merit    (d31)    0.473    0.188    2.522    0.012
    sysRac   (d32)   -0.196    0.233   -0.842    0.400
    polAffil  (a3)   -0.149    0.128   -1.165    0.244
  sysRac ~
    merit    (d21)   -0.301    0.110   -2.747    0.006
    polAffil  (a2)    0.090    0.075    1.194    0.233
  merit ~
    polAffil  (a1)   -0.266    0.059   -4.480    0.000
```

Example

```
partSummary(out3.1, 8:9)

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .policy            0.933    1.036    0.900    0.368
   .revDisc           3.108    1.528    2.033    0.042
   .sysRac            4.698    0.665    7.061    0.000
   .merit             4.977    0.260   19.118    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .policy            0.985    0.157    6.291    0.000
   .revDisc           2.361    0.295    8.010    0.000
   .sysRac            0.689    0.095    7.236    0.000
   .merit             0.719    0.121    5.960    0.000
```

Example

```
partSummary(out3.1, 10)

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)
    ab1              -0.001    0.037   -0.036    0.972
    ab2               0.053    0.046    1.158    0.247
    ab3               0.004    0.015    0.260    0.794
    partIE1           0.003    0.011    0.292    0.771
    partIE2           0.047    0.025    1.909    0.056
    partIE3           0.000    0.003    0.163    0.871
    fullIE            0.000    0.003    0.165    0.869
    totalIE           0.107    0.053    2.036    0.042
```

Example

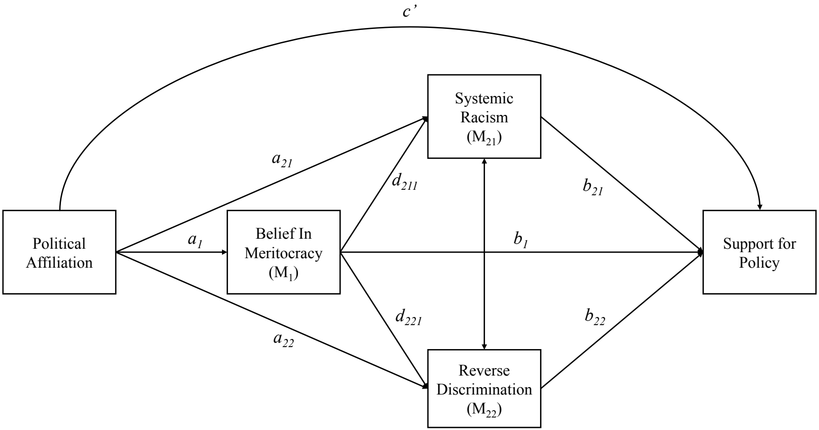
```
parameterEstimates(out3.1, boot.ci.type = "bca.simple") %>%
    select(c("label", "est", "ci.lower", "ci.upper")) %>%
    tail(8)

      label    est ci.lower ci.upper
21      ab1 -0.001   -0.078    0.074
22      ab2  0.053   -0.042    0.145
23      ab3  0.004   -0.013    0.054
24 partIE1  0.003   -0.012    0.040
25 partIE2  0.047    0.010    0.109
26 partIE3  0.000   -0.001    0.017
27  fullIE  0.000   -0.001    0.013
28 totalIE  0.107    0.002    0.213
```

Hybrid Multiple Mediation

Hybrid Multiple Mediation

We can also combine parallel and serial mediation models:



Example

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

Example

```
partSummary(out4.1, 7)

Regressions:
                    Estimate  Std.Err  z-value  P(>|z|)
  policy ~
    merit     (b1)     0.005    0.149    0.033    0.973
    sysRac   (b21)     0.589    0.143    4.117    0.000
    revDisc  (b22)    -0.026    0.083   -0.317    0.751
    polAffl   (cp)     0.130    0.080    1.623    0.105
  sysRac ~
    merit   (d211)    -0.301    0.109   -2.770    0.006
    polAffl  (a21)     0.090    0.071    1.268    0.205
  revDisc ~
    merit   (d221)     0.532    0.191    2.791    0.005
    polAffl  (a22)    -0.167    0.129   -1.291    0.197
  merit ~
    polAffl   (a1)    -0.266    0.062   -4.297    0.000
```

Example

```
partSummary(out4.1, 8:10)

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
 .sysRac ~~
   .revDisc          -0.135    0.165   -0.817    0.414

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .policy            0.933    1.024    0.911    0.362
   .sysRac            4.698    0.635    7.395    0.000
   .revDisc           2.187    1.146    1.907    0.056
   .merit             4.977    0.268   18.606    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .policy            0.985    0.177    5.551    0.000
   .sysRac            0.689    0.098    7.061    0.000
   .revDisc           2.388    0.318    7.506    0.000
   .merit             0.719    0.112    6.433    0.000
```

Example

```
partSummary(out4.1, 11)

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)
    ab1              -0.001    0.041   -0.033    0.974
    ab21              0.053    0.044    1.209    0.227
    ab22              0.004    0.018    0.243    0.808
    fullIE21          0.047    0.025    1.913    0.056
    fullIE22          0.004    0.013    0.281    0.778
    totalIE           0.107    0.054    1.973    0.048
```

Example

```r
parameterEstimates(out4.1, boot.ci.type = "bca.simple") %>%
    select(c("label", "est", "ci.lower", "ci.upper")) %>%
    tail(6)

      label    est ci.lower ci.upper
21      ab1 -0.001   -0.076    0.089
22     ab21  0.053   -0.025    0.157
23     ab22  0.004   -0.021    0.064
24 fullIE21  0.047    0.012    0.121
25 fullIE22  0.004   -0.015    0.044
26  totalIE  0.107    0.010    0.226
```

Practice

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