

Structural Equation Modeling & Mediation

Introduction to SEM with Lavaan



**Utrecht
University**

Kyle M. Lang

Department of Methodology & Statistics
Utrecht University

Outline

Structural Equation Modeling

Mediation

- Simple Mediation

- Bootstrapping

- Multiple Mediation

 - Parallel Mediators

 - Serial Mediators

- Mediation + SEM

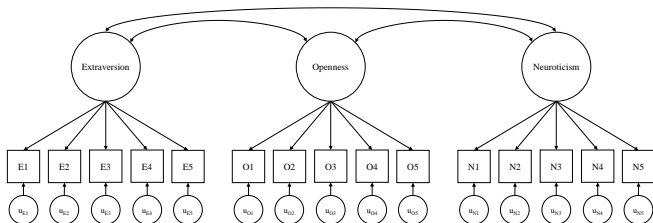


Full SEM

Structural equation modeling (SEM) simply combines path analysis and CFA.

- SEM allows us to model complicated structural relations among latent variables.

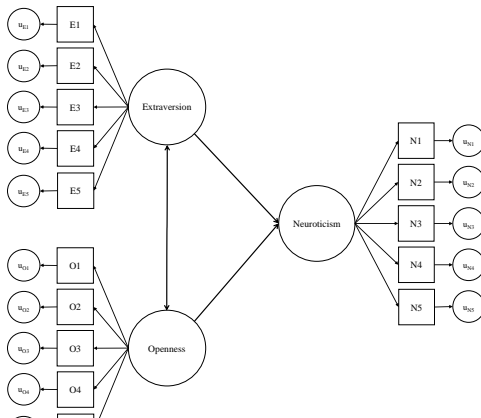
Let's consider a simple, three-factor CFA model.



CFA → SEM

We first evaluate the validity of the measurement model via CFA.

- We then convert the CFA to an SEM by converting some covariances to latent regression paths.



CFA Example

```
## Load the lavaan package and some data:
library(lavaan)
data(bfi, package = "psych")

## Specify the CFA model:
cfaMod <- '
extra =~ E1 + E2 + E3 + E4 + E5
open  =~ O1 + O2 + O3 + O4 + O5
neuro =~ N1 + N2 + N3 + N4 + N5
'
```

```
## Estimate the model:
cfaOut <- cfa(cfaMod, data = bfi, missing = "fiml", std.lv = TRUE)
```

```
## Check the fit:
fitMeasures(cfaOut,
             c("chisq", "df", "pvalue", "cfi", "tli", "rmsea", "srmr")
)
```

chisq	df	pvalue	cfi	tli	rmsea	srmr
2251.679	87.000	0.000	0.809	0.769	0.094	0.081

CFA Example

```
partSummary(cfaOut, 7)
```

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)
extra =~				
E1	0.973	0.032	30.607	0.000
E2	1.163	0.030	38.171	0.000
E3	-0.815	0.027	-30.358	0.000
E4	-0.979	0.028	-35.254	0.000
E5	-0.714	0.027	-26.638	0.000
open =~				
O1	0.630	0.025	24.886	0.000
O2	-0.605	0.036	-16.781	0.000
O3	0.897	0.029	30.765	0.000
O4	0.290	0.028	10.402	0.000
O5	-0.602	0.031	-19.734	0.000
neuro =~				
N1	1.272	0.027	47.254	0.000
N2	1.218	0.026	46.491	0.000
N3	1.157	0.029	40.195	0.000
N4	0.892	0.030	29.356	0.000
N5	0.823	0.031	26.163	0.000

CFA Example

```
partSummary(cfaOut, 8)
```

Covariances:

	Estimate	Std.Err	z-value	P(> z)
extra ~~				
open	-0.444	0.024	-18.472	0.000
neuro	0.240	0.023	10.551	0.000
open ~~				
neuro	-0.117	0.025	-4.667	0.000

CFA Example

```
partSummary(cfaOut, 9)
```

Intercepts:

	Estimate	Std.Err	z-value	P(> z)
.E1	2.974	0.031	96.223	0.000
.E2	3.143	0.030	103.424	0.000
.E3	4.002	0.026	156.117	0.000
.E4	4.421	0.028	160.350	0.000
.E5	4.417	0.025	174.595	0.000
.01	4.816	0.021	224.964	0.000
.02	2.713	0.030	91.745	0.000
.03	4.436	0.023	191.555	0.000
.04	4.892	0.023	211.519	0.000
.05	2.490	0.025	98.932	0.000
.N1	2.932	0.030	98.589	0.000
.N2	3.508	0.029	121.459	0.000
.N3	3.217	0.030	106.147	0.000
.N4	3.185	0.030	106.894	0.000
.N5	2.969	0.031	96.663	0.000
extra	0.000			
open	0.000			
neuro	0.000			

CFA Example

```
partSummary(cfaOut, 10)
```

Variances:

	Estimate	Std.Err	z-value	P(> z)
.E1	1.713	0.054	31.442	0.000
.E2	1.224	0.049	24.952	0.000
.E3	1.163	0.038	30.388	0.000
.E4	1.166	0.041	28.522	0.000
.E5	1.272	0.039	32.789	0.000
.O1	0.878	0.031	28.320	0.000
.O2	2.083	0.062	33.705	0.000
.O3	0.686	0.043	16.130	0.000
.O4	1.407	0.039	36.236	0.000
.O5	1.401	0.044	31.837	0.000
.N1	0.848	0.037	23.029	0.000
.N2	0.842	0.035	24.184	0.000
.N3	1.228	0.043	28.308	0.000
.N4	1.666	0.051	32.808	0.000
.N5	1.942	0.056	34.465	0.000
extra	1.000			
open	1.000			
neuro	1.000			

SEM Example

```
## Add structural paths:
```

```
semMod <- '  
extra =~ E1 + E2 + E3 + E4 + E5  
open  =~ O1 + O2 + O3 + O4 + O5  
neuro =~ N1 + N2 + N3 + N4 + N5
```

```
neuro ~ extra + open  
,
```

```
## Estimate the model:
```

```
semOut <- sem(semMod, data = bfi, missing = "fiml", std.lv = TRUE)
```

```
## Check the fit:
```

```
fitMeasures(semOut,  
             c("chisq", "df", "pvalue", "cfi", "tli", "rmsea", "srmr")  
             )
```

chisq	df	pvalue	cfi	tli	rmsea	srmr
2251.679	87.000	0.000	0.809	0.769	0.094	0.081

SEM Example

```
partSummary(semOut, 7)
```

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)
extra =~				
E1	0.973	0.032	30.607	0.000
E2	1.163	0.030	38.172	0.000
E3	-0.815	0.027	-30.358	0.000
E4	-0.979	0.028	-35.254	0.000
E5	-0.714	0.027	-26.638	0.000
open =~				
O1	0.630	0.025	24.886	0.000
O2	-0.605	0.036	-16.781	0.000
O3	0.897	0.029	30.765	0.000
O4	0.290	0.028	10.402	0.000
O5	-0.602	0.031	-19.734	0.000
neuro =~				
N1	1.235	0.027	45.916	0.000
N2	1.183	0.026	45.360	0.000
N3	1.123	0.028	39.976	0.000
N4	0.866	0.029	29.605	0.000
N5	0.799	0.031	26.204	0.000

SEM Example

```
partSummary(semOut, 8:9)
```

Regressions:

	Estimate	Std.Err	z-value	P(> z)
neuro ~				
extra	0.241	0.030	8.169	0.000
open	-0.014	0.031	-0.448	0.654

Covariances:

	Estimate	Std.Err	z-value	P(> z)
extra ~~				
open	-0.444	0.024	-18.472	0.000

SEM Example

```
partSummary(semOut, 10)
```

Intercepts:

	Estimate	Std.Err	z-value	P(> z)
.E1	2.974	0.031	96.223	0.000
.E2	3.143	0.030	103.424	0.000
.E3	4.002	0.026	156.117	0.000
.E4	4.421	0.028	160.350	0.000
.E5	4.417	0.025	174.595	0.000
.01	4.816	0.021	224.964	0.000
.02	2.713	0.030	91.745	0.000
.03	4.436	0.023	191.555	0.000
.04	4.892	0.023	211.520	0.000
.05	2.490	0.025	98.932	0.000
.N1	2.932	0.030	98.589	0.000
.N2	3.508	0.029	121.459	0.000
.N3	3.217	0.030	106.146	0.000
.N4	3.185	0.030	106.894	0.000
.N5	2.969	0.031	96.663	0.000
extra	0.000			
open	0.000			
.neuro	0.000			

SEM Example

```
partSummary(semOut, 11)
```

Variances:

	Estimate	Std.Err	z-value	P(> z)
.E1	1.713	0.054	31.442	0.000
.E2	1.224	0.049	24.952	0.000
.E3	1.163	0.038	30.388	0.000
.E4	1.166	0.041	28.522	0.000
.E5	1.272	0.039	32.789	0.000
.O1	0.878	0.031	28.320	0.000
.O2	2.083	0.062	33.705	0.000
.O3	0.686	0.043	16.130	0.000
.O4	1.407	0.039	36.236	0.000
.O5	1.401	0.044	31.837	0.000
.N1	0.848	0.037	23.029	0.000
.N2	0.842	0.035	24.184	0.000
.N3	1.228	0.043	28.308	0.000
.N4	1.666	0.051	32.808	0.000
.N5	1.942	0.056	34.465	0.000
extra	1.000			
open	1.000			
.neuro	1.000			

Why SEM?

The beauty of SEM is that we get to model the types of complex relations we can specify via path models while leveraging all the strengths of latent variables.

- When we fit a multiple-group SEM, we are modeling moderation by group.
 - The latent variables give us the ability to evaluate measurement invariance across groups.
 - We'll see more of these ideas in the next lecture.
- Path analysis and SEM lend themselves especially well to mediation analysis and conditional process analysis.



MEDIATION



Mediation vs. Moderation

What do we mean by *mediation* and *moderation*?

Mediation and moderation are types of hypotheses, not statistical methods or models.

- Mediation tells us *how* one variable influences another.
- Moderation tells us *when* one variable influences another.



Contextualizing Example

Say we wish to explore the process underlying exercise habits.

Our first task is to operationalize “exercise habits”

- DV: Hours per week spent in vigorous exercise (*exerciseAmount*).

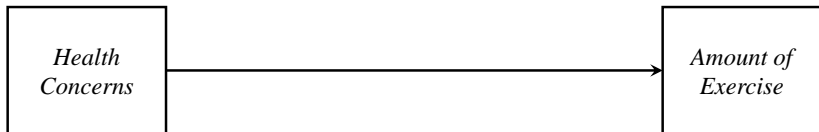
We may initial ask: what predicts devoting more time to exercise?

- IV: Concerns about negative health outcomes (*healthConcerns*).



Focal Effect Only

The *healthConcerns* → *exerciseAmount* relation is our *focal effect*

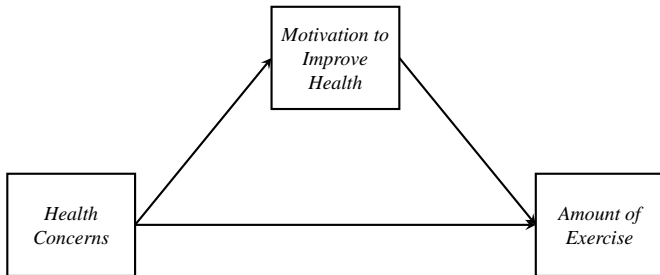


- Mediation, moderation, and conditional process analysis all attempt to describe the focal effect in more detail.
- We always begin by hypothesizing a focal effect.

The Mediation Hypothesis

A mediation analysis will attempt to describe how health concerns affect amount of exercise.

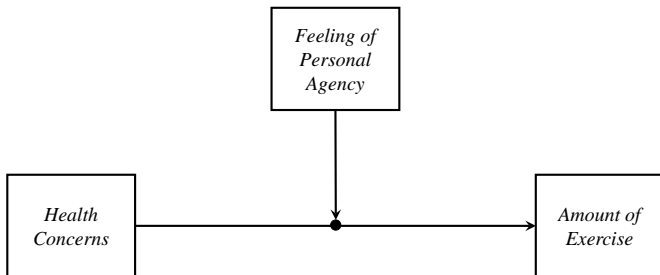
- The *how* is operationalized in terms of intermediary variables.
- Mediator: Motivation to improve health (*motivation*).



Moderation Hypothesis

A moderation hypothesis will attempt to describe when health concerns affect amount of exercise.

- The *when* is operationalized in terms of interactions between the focal predictor and contextualizing variables
- Moderator: Sense of personal agency relating to physical health (*agency*).



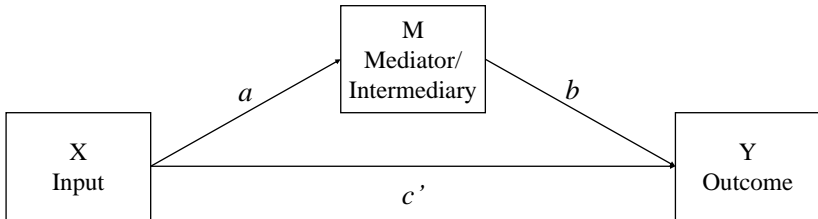
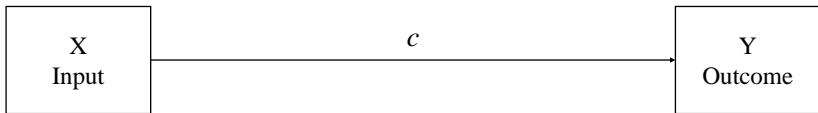
Conditional Process Analysis

Conditional process analysis combines the mediation and moderation hypotheses into models of moderated mediation.

- Given a mediation model describing *how* health concerns affect exercise amount, what other variables may modulate the indirect effect.



Path Diagrams



Necessary Equations

To get all the pieces of the preceding diagram using OLS regression, we'll need to fit three separate models.

$$Y = i_1 + cX + e_1 \quad (1)$$

$$Y = i_2 + c'X + bM + e_2 \quad (2)$$

$$M = i_3 + aX + e_3 \quad (3)$$

- Equation 1 gives us the total effect (c).
- Equation 2 gives us the direct effect (c') and the partialled effect of the mediator on the outcome (b).
- Equation 3 gives us the effect of the input on the outcome (a).

Two Measures of Indirect Effect

Indirect effects can be quantified in two different ways:

$$IE_{diff} = c - c' \quad (4)$$

$$IE_{prod} = a \cdot b \quad (5)$$

IE_{diff} and IE_{prod} are equivalent in simple mediation.

- Both give us information about the proportion of the total effect that is transmitted through the intermediary variable.
- IE_{prod} provides a more direct representation of the actual pathway we're interested in testing.
- IE_{diff} gets at our desired hypothesis indirectly.



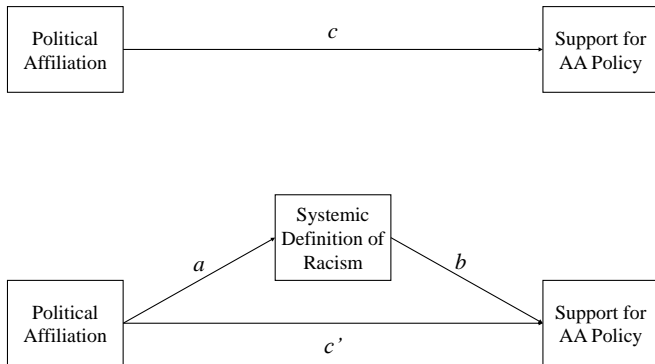
The Causal Steps Approach

Baron and Kenny (1986, p. 1176) describe three/four conditions as being sufficient to demonstrate statistical “mediation.”

1. Variations in levels of the independent variable significantly account for variations in the presumed mediator (i.e., Path a).
 - Need a significant a path.
2. Variations in the mediator significantly account for variations in the dependent variable (i.e., Path b).
 - Need a significant b path.
3. When Paths a and b are controlled, a previously significant relation between the independent and dependent variables is no longer significant.
 - Need a significant total effect
 - The direct effect must be “less” than the total effect

Example Process Model

Consider the following process.



Causal Steps Example

```
## Load some data:
dat1 <- readRDS("../data/adamsKlpsScaleScore.rds")

## Check pre-conditions:
mod1 <- lm(policy ~ polAffil, data = dat1)
mod2 <- lm(policy ~ sysRac, data = dat1)
mod3 <- lm(sysRac ~ polAffil, data = dat1)

## Partial out the mediator's effect:
mod4 <- lm(policy ~ sysRac + polAffil, data = dat1)
```

Causal Steps Example

```
summary(mod1)
```

Call:

```
lm(formula = policy ~ polAffil, data = dat1)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.7357	-0.8254	0.0643	0.6827	3.2481

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	2.71516	0.35648	7.617	3.32e-11	***
polAffil	0.23675	0.07775	3.045	0.0031	**

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.134 on 85 degrees of freedom

Multiple R-squared: 0.09836, Adjusted R-squared: 0.08775

F-statistic: 9.273 on 1 and 85 DF, p-value: 0.003096

Causal Steps Example

```
summary(mod2)
```

Call:

```
lm(formula = policy ~ sysRac, data = dat1)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.28970	-0.53821	0.08866	0.64015	3.08343

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.1218	0.4883	2.297	0.0241 *
sysRac	0.6649	0.1210	5.494	4.03e-07 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.026 on 85 degrees of freedom

Multiple R-squared: 0.262, Adjusted R-squared: 0.2534

F-statistic: 30.18 on 1 and 85 DF, p-value: 4.029e-07

Causal Steps Example

```
summary(mod3)
```

Call:

```
lm(formula = sysRac ~ polAffil, data = dat1)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.2187	-0.5449	-0.2115	0.6182	1.9516

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.19726	0.27634	11.570	<2e-16 ***
polAffil	0.17023	0.06027	2.825	0.0059 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8788 on 85 degrees of freedom

Multiple R-squared: 0.08581, Adjusted R-squared: 0.07505

F-statistic: 7.978 on 1 and 85 DF, p-value: 0.005898

Causal Steps Example

```
summary(mod4)
```

Call:

```
lm(formula = policy ~ sysRac + polAffil, data = dat1)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.1370	-0.6338	-0.0020	0.6658	3.4674

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.80704	0.51013	1.582	0.1174
sysRac	0.59680	0.12478	4.783	7.3e-06 ***
polAffil	0.13515	0.07252	1.864	0.0658 .

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.011 on 84 degrees of freedom

Multiple R-squared: 0.2913, Adjusted R-squared: 0.2745

F-statistic: 17.27 on 2 and 84 DF, p-value: 5.228e-07

Causal Steps Example

```
## Extract important parameter estimates:
```

```
a      <- coef(mod3)["polAffil"]
```

```
b      <- coef(mod4)["sysRac"]
```

```
c      <- coef(mod1)["polAffil"]
```

```
cPrime <- coef(mod4)["polAffil"]
```

```
## Compute indirect effects:
```

```
ieDiff <- unname(c - cPrime)
```

```
ieProd <- unname(a * b)
```

```
ieDiff
```

```
[1] 0.1015958
```

```
ieProd
```

```
[1] 0.1015958
```

Sobel's Z

In the previous example, do we have a *significant* indirect effect?

- The direct effect is “substantially” smaller than the total effect, but is the difference statistically significant?
- Sobel (1982) developed an asymptotic standard error for IE_{prod} that we can use to assess this hypothesis.

$$SE_{sobel} = \sqrt{a^2 \cdot SE_b^2 + b^2 \cdot SE_a^2} \quad (6)$$

$$Z_{sobel} = \frac{ab}{SE_{sobel}} \quad (7)$$

$$95\%CI_{sobel} = ab \pm 1.96 \cdot SE_{sobel} \quad (8)$$

Sobel Example

```
## SE:
seA <- (mod3 %>% vcov() %>% diag() %>% sqrt())["polAffil"]
seB <- (mod4 %>% vcov() %>% diag() %>% sqrt())["sysRac"]

se <- sqrt(b^2 * seA^2 + a^2 * seB^2) %>% unname()

## z-score:
(z <- ieProd / se)

[1] 2.432107

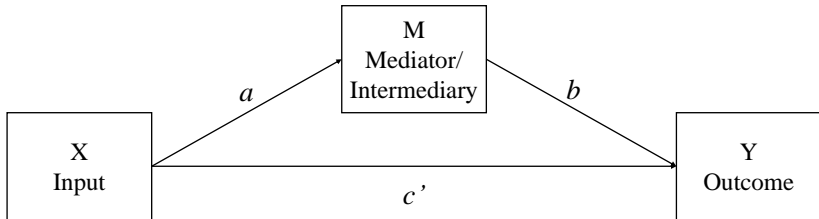
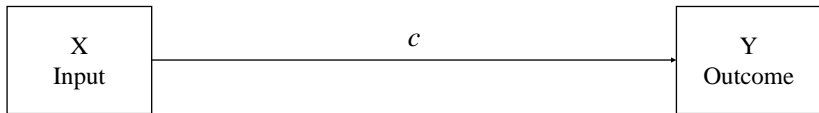
## p-value:
(p <- 2 * pnorm(z, lower = FALSE))

[1] 0.01501126

## 95% CI:
c(ieProd - 1.96 * se, ieProd + 1.96 * se)

[1] 0.01972121 0.18347034
```

Recall our Basic Path Diagram



Two Measures of Indirect Effect

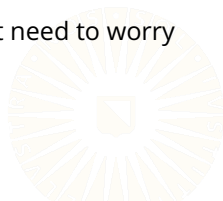
Recall the two definitions of an indirect effect:

$$IE_{diff} = c - c' \quad (9)$$

$$IE_{prod} = a \cdot b \quad (10)$$

It pays to remember a few key points:

- IE_{diff} and IE_{prod} are equivalent in simple mediation.
- IE_{diff} is only an indirect indication of IE_{prod} .
- A significant indirect effect can exist without a significant total effect.
- If we only care about the indirect effect, then we don't need to worry about the total effect.



Two Measures of Indirect Effect

Recall the two definitions of an indirect effect:

$$IE_{diff} = c - c' \quad (9)$$

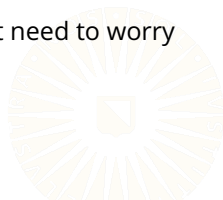
$$IE_{prod} = a \cdot b \quad (10)$$

It pays to remember a few key points:

- IE_{diff} and IE_{prod} are equivalent in simple mediation.
- IE_{diff} is only an indirect indication of IE_{prod} .
- A significant indirect effect can exist without a significant total effect.
- If we only care about the indirect effect, then we don't need to worry about the total effect.

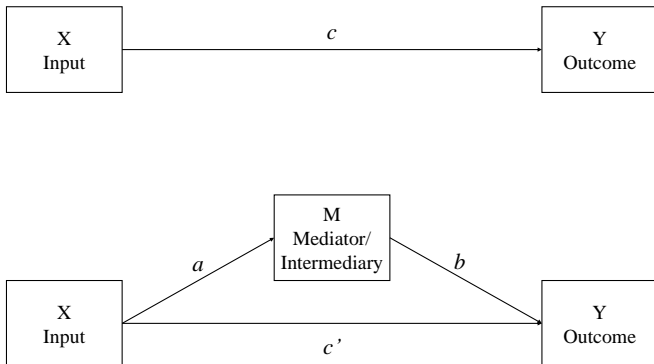
These points imply something interesting:

- We don't need to estimate c !



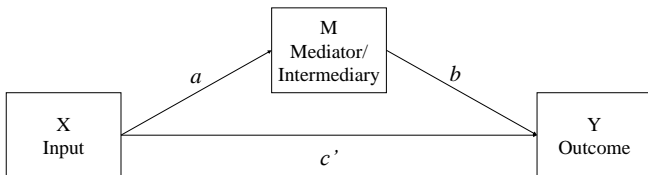
Simplifying our Path Diagram

QUESTION: If we don't care about directly estimating c , how can we simplify this diagram?

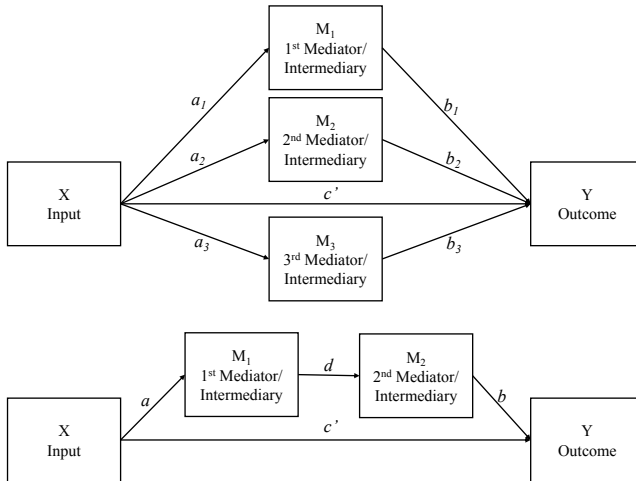


Simplifying our Path Diagram

ANSWER: We don't fit the upper model.

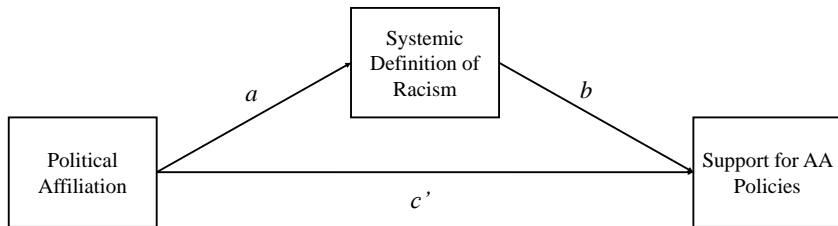


Why Path Analysis?



Example

Let's revisit the above example using path analysis in **lavaan**.



Example

```
## Load the lavaan package:
library(lavaan)

## Specify the basic path model:
mod1 <- '
policy ~ 1 + sysRac + polAffil
sysRac ~ 1 + polAffil
'

## Estimate the model:
out1 <- sem(mod1, data = dat1)
```

Example

```
## Look at the results:
```

```
partSummary(out1, 7:9)
```

Regressions:

	Estimate	Std.Err	z-value	P(> z)
policy ~				
sysRac	0.597	0.123	4.867	0.000
polAffil	0.135	0.071	1.897	0.058
sysRac ~				
polAffil	0.170	0.060	2.858	0.004

Intercepts:

	Estimate	Std.Err	z-value	P(> z)
.policy	0.807	0.501	1.610	0.107
.sysRac	3.197	0.273	11.705	0.000

Variances:

	Estimate	Std.Err	z-value	P(> z)
.policy	0.987	0.150	6.595	0.000
.sysRac	0.755	0.114	6.595	0.000

Example

```
## Include the indirect effect:
mod2 <- '
policy ~ 1 + b*sysRac + polAffil
sysRac ~ 1 + a*polAffil

ab := a*b # Define a parameter for the indirect effect
'

## Estimate the model:
out2 <- sem(mod2, data = dat1)
```

Example

```
## Look at the results:
```

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

Regressions:

		Estimate	Std.Err	z-value	P(> z)
policy ~					
sysRac	(b)	0.597	0.123	4.867	0.000
polAffil		0.135	0.071	1.897	0.058
sysRac ~					
polAffil	(a)	0.170	0.060	2.858	0.004

Intercepts:

	Estimate	Std.Err	z-value	P(> z)
.policy	0.807	0.501	1.610	0.107
.sysRac	3.197	0.273	11.705	0.000

Example

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

Variances:

	Estimate	Std.Err	z-value	P(> z)
.policy	0.987	0.150	6.595	0.000
.sysRac	0.755	0.114	6.595	0.000

Defined Parameters:

	Estimate	Std.Err	z-value	P(> z)
ab	0.102	0.041	2.464	0.014

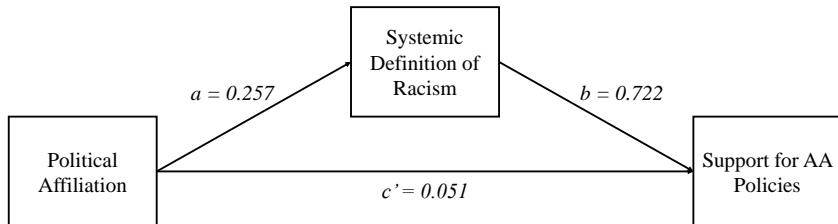
Example

We can also get CIs:

```
parameterEstimates(out2, zstat = FALSE, pvalue = FALSE, ci = TRUE)
```

	lhs	op	rhs	label	est	se	ci.lower	ci.upper
1	policy	~1			0.807	0.501	-0.175	1.789
2	policy	~	sysRac	b	0.597	0.123	0.356	0.837
3	policy	~	polAffil		0.135	0.071	-0.005	0.275
4	sysRac	~1			3.197	0.273	2.662	3.733
5	sysRac	~	polAffil	a	0.170	0.060	0.053	0.287
6	policy	~~	policy		0.987	0.150	0.694	1.280
7	sysRac	~~	sysRac		0.755	0.114	0.530	0.979
8	polAffil	~~	polAffil		2.444	0.000	2.444	2.444
9	polAffil	~1			4.310	0.000	4.310	4.310
10	ab	:=	a*b	ab	0.102	0.041	0.021	0.182

Results



We're not there yet...

Path analysis allows us to directly model complex (and simple) relations, but the preceding example still suffers from a considerable limitation.

- The significance test for the indirect effect is still conducted with the Sobel Z approach.

Path analysis (or full SEM) doesn't magically get around distributional problems associated with Sobel's Z test.

- To get a robust significance test of the indirect effect, we need to use *bootstrapping*.



Bootstrapping

Bootstrapping was introduced by Efron (1979) as a tool for non-parametric inference.

- Traditional inference requires that we assume a parametric sampling distribution for our focal parameter.
- We need to make such an assumption to compute the standard errors we require for inferences.
- If we cannot safely make these assumptions, we can use bootstrapping.



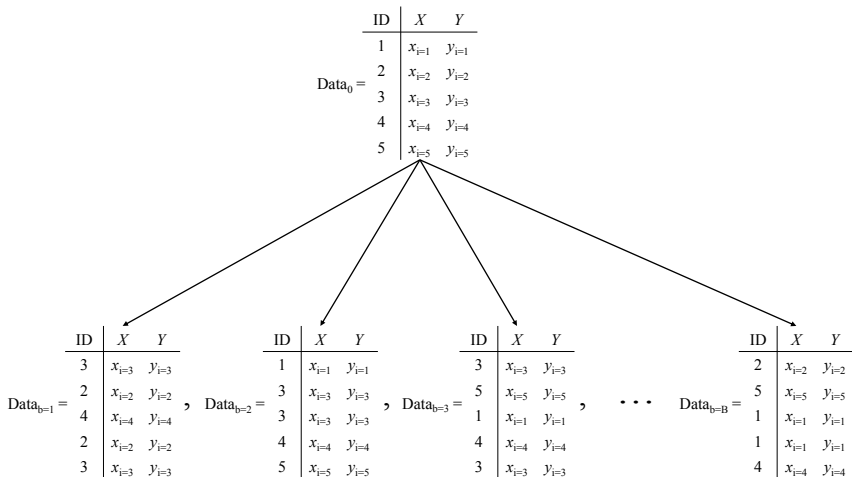
Bootstrapping

Assume our observed data $Data_0$ represent the population and:

1. Sample rows of $Data_0$, with replacement, to create B new samples $\{Data_b\}$.
2. Calculate our focal statistic on each of the B bootstrap samples.
3. Make inferences based on the empirical distribution of the B estimates calculated in Step 2



Bootstrapping



Example

Suppose I'm on the lookout for a retirement location. Since I want to relax in my old-age, I'm concerned with ensuring a low probability of dragon attacks, so I have a few salient considerations:

- Shooting for a location with no dragons, whatsoever, is a fools errand (since dragons are, obviously, ubiquitous).
- I merely require a location that has at least two times as many dragon-free days as other kinds.



Example

I've been watching several candidate locales over the course of my (long and illustrious) career, and I'm particularly hopeful about one quiet hamlet in the Patagonian highlands.

- To ensure that my required degree of dragon-freeness is met, I'll use the *Dragon Risk Index* (DRI):

$$DRI = \text{Median} \left(\frac{\text{Dragon-Free Days}}{\text{Dragonned Days}} \right)$$



Example

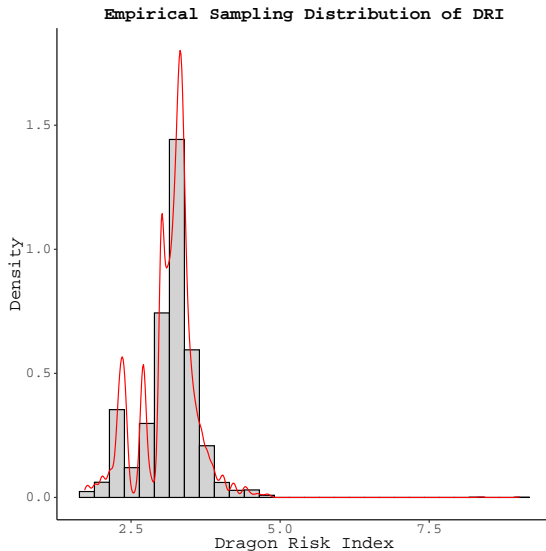
```
## Read in the observed data:
rawData <- readRDS("../data/daysData.rds")

## Compute the observed test statistic:
obsDRI <- median(rawData$goodDays / rawData$badDays)
obsDRI

[1] 3.24476

## Draw the bootstrap samples:
set.seed(235711)
nSams <- 5000
bootDRI <- rep(NA, nSams)
for(b in 1:nSams) {
  bootSam <- rawData[sample(1:nrow(rawData), replace = TRUE), ]
  bootDRI[b] <- median(bootSam$goodDays / bootSam$badDays)
}
```


Example



Example

To see if I can be confident in the dragon-freeness of my potential home, I'll summarize the preceding distribution with a (one-tailed) percentile confidence interval:

```
bootLB <- sort(bootDRI)[0.05 * nSams]
bootUB <- Inf

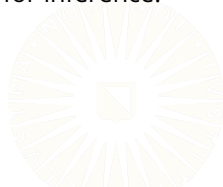
## The bootstrapped Percentile CI:
c(bootLB, bootUB)

[1] 2.288555      Inf
```

Bootstrapped Inference for Indirect Effects

We can apply the same procedure to testing the indirect effect.

- The problem with Sobel's Z is exactly the type of issue for which bootstrapping was designed
 - We don't know a reasonable finite-sample sampling distribution for the *ab* parameter.
- Bootstrapping will allow us to construct an empirical sampling distribution for *ab* and construct confidence intervals for inference.



Bootstrapped Inference for Indirect Effects

PROCEDURE:

1. Resample our observed data with replacement
2. Fit our hypothesized path model to each bootstrap sample
3. Store the value of ab that we get each time
4. Summarize the empirical distribution of ab to make inferences



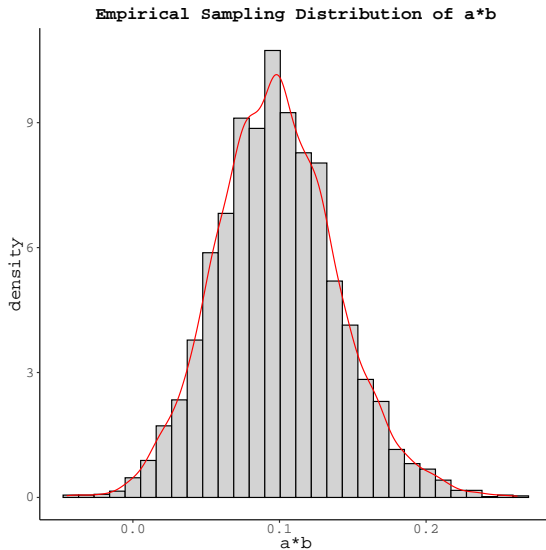
Example

```
abVec <- rep(NA, nSams)
for(i in 1:nSams) {
  ## Resample the data:
  bootSam <- dat1[sample(1:nrow(dat1), replace = TRUE), ]

  ## Fit the path model:
  bootOut <- sem(mod2, data = bootSam)

  ## Store the estimated indirect effect:
  abVec[i] <- coef(bootOut)[c("a", "b")] %>% prod()
}
```

Example



Example

```
## Calculate the percentile CI:  
lb <- sort(abVec)[0.025 * nSams]  
ub <- sort(abVec)[0.975 * nSams]  
c(lb, ub)  
  
[1] 0.01983165 0.18521052
```

Example

```
## Much more parsimoniously:
```

```
bootOut2 <- sem(mod2, data = dat1, se = "boot", bootstrap = nSams)
```

```
parameterEstimates(bootOut2, zstat = FALSE, pvalue = FALSE)
```

	lhs	op	rhs	label	est	se	ci.lower	ci.upper
1	policy	~1			0.807	0.554	-0.269	1.912
2	policy	~	sysRac	b	0.597	0.136	0.314	0.858
3	policy	~	polAffil		0.135	0.084	-0.030	0.303
4	sysRac	~1			3.197	0.276	2.674	3.782
5	sysRac	~	polAffil	a	0.170	0.063	0.039	0.292
6	policy	~~	policy		0.987	0.166	0.650	1.291
7	sysRac	~~	sysRac		0.755	0.109	0.530	0.959
8	polAffil	~~	polAffil		2.444	0.000	2.444	2.444
9	polAffil	~1			4.310	0.000	4.310	4.310
10	ab	:=	a*b	ab	0.102	0.042	0.024	0.189

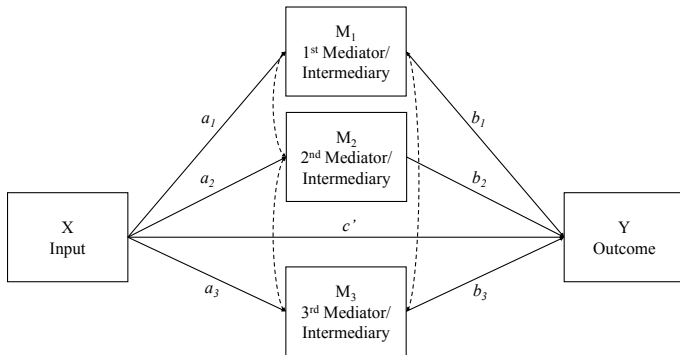
Simple Mediation is Too Simple

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 estimate four equations:

$$\begin{aligned}Y &= i_Y + b_1M_1 + b_2M_2 + b_3M_3 + c'X + e_Y \\M_1 &= i_{M1} + a_1X + e_{M1} \\M_2 &= i_{M2} + a_2X + e_{M2} \\M_3 &= i_{M3} + a_3X + e_{M3}\end{aligned}$$

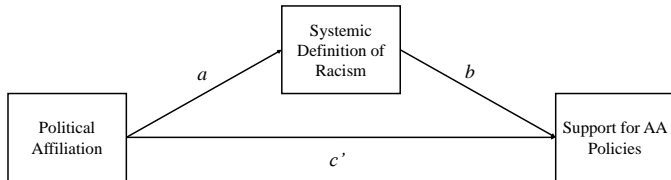
In general, a parallel mediator model with K mediator variables will required $K + 1$ separate equations.

Path modeling can make this task much simpler.

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

Parallel Multiple Mediation

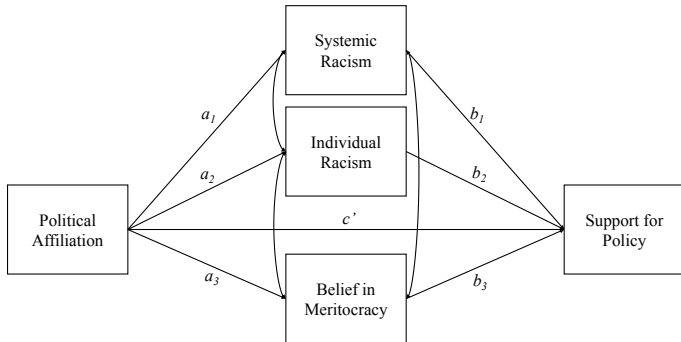
Let's reconsider the last example:



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: $a_1b_1, a_2b_2, \dots, a_Kb_K$.
- 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
'

## Fit the model:
out1.1 <- sem(mod1.1, data = dat1, se = "boot", bootstrap = 5000)
```

Example

```
## Look at results:
```

```
partSummary(out1.1, 7)
```

Regressions:

		Estimate	Std.Err	z-value	P(> z)
policy ~					
sysRac	(b1)	0.601	0.142	4.243	0.000
indRac	(b2)	0.143	0.105	1.359	0.174
merit	(b3)	-0.036	0.153	-0.238	0.812
polAffil	(cp)	0.125	0.077	1.637	0.102
sysRac ~					
polAffil	(a1)	0.170	0.065	2.623	0.009
indRac ~					
polAffil	(a2)	-0.004	0.077	-0.055	0.956
merit ~					
polAffil	(a3)	-0.266	0.060	-4.429	0.000

Example

```
partSummary(out1.1, 9)
```

Intercepts:

	Estimate	Std.Err	z-value	P(> z)
.policy	0.490	0.886	0.553	0.580
.sysRac	3.197	0.282	11.325	0.000
.indRac	3.398	0.319	10.656	0.000
.merit	4.977	0.259	19.252	0.000

Example

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

Variances:

	Estimate	Std.Err	z-value	P(> z)
.policy	0.963	0.177	5.454	0.000
.sysRac	0.755	0.110	6.880	0.000
.indRac	1.188	0.152	7.816	0.000
.merit	0.719	0.112	6.444	0.000

Covariances:

	Estimate	Std.Err	z-value	P(> z)
.sysRac ~~				
.indRac	-0.076	0.102	-0.742	0.458
.merit	-0.217	0.092	-2.343	0.019
.indRac ~~				
.merit	0.154	0.098	1.577	0.115

Example

```
partSummary(out1.1, 11)
```

Defined Parameters:

	Estimate	Std.Err	z-value	P(> z)
ab1	0.102	0.045	2.283	0.022
ab2	-0.001	0.015	-0.042	0.966
ab3	0.010	0.042	0.230	0.818
totalIE	0.111	0.053	2.109	0.035

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	0.030	0.213
22	ab2	-0.001	-0.035	0.029
23	ab3	0.010	-0.075	0.090
24	totalIE	0.111	0.015	0.220

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)?



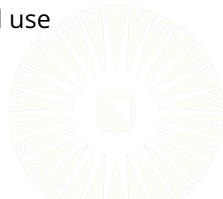
Comparing Specific Indirect Effects

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	0.497	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
0.009440066
```

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 = 5000)
```

Example

```
## Look at results:
```

```
partSummary(out1.3, 7)
```

Regressions:

		Estimate	Std.Err	z-value	P(> z)
policy ~					
sysRac	(b1)	0.601	0.144	4.161	0.000
indRac	(b2)	0.143	0.108	1.326	0.185
merit	(b3)	-0.036	0.152	-0.239	0.811
polAffil	(cp)	0.125	0.079	1.592	0.111
sysRac ~					
polAffil	(a1)	0.170	0.063	2.710	0.007
indRac ~					
polAffil	(a2)	-0.004	0.078	-0.055	0.956
merit ~					
polAffil	(a3)	-0.266	0.061	-4.390	0.000

Example

```
partSummary(out1.3, 9)
```

Intercepts:

	Estimate	Std.Err	z-value	P(> z)
.policy	0.490	0.894	0.548	0.584
.sysRac	3.197	0.274	11.684	0.000
.indRac	3.398	0.325	10.463	0.000
.merit	4.977	0.260	19.155	0.000

Example

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

Variances:

	Estimate	Std.Err	z-value	P(> z)
.policy	0.963	0.178	5.424	0.000
.sysRac	0.755	0.109	6.903	0.000
.indRac	1.188	0.155	7.680	0.000
.merit	0.719	0.111	6.479	0.000

Covariances:

	Estimate	Std.Err	z-value	P(> z)
.sysRac ~~				
.indRac	-0.076	0.102	-0.742	0.458
.merit	-0.217	0.090	-2.416	0.016
.indRac ~~				
.merit	0.154	0.099	1.553	0.120

Example

```
partSummary(out1.3, 11)
```

Defined Parameters:

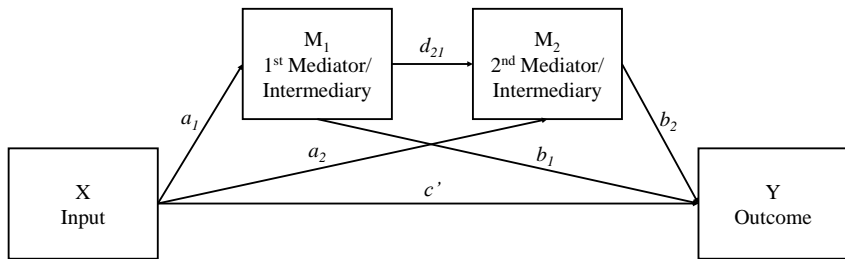
	Estimate	Std.Err	z-value	P(> z)
ab1	0.102	0.044	2.340	0.019
ab2	-0.001	0.015	-0.042	0.967
ab3	0.010	0.042	0.231	0.818
totalIE	0.111	0.050	2.216	0.027
test1	-0.103	0.048	-2.158	0.031

Example

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

	label	est	ci.lower	ci.upper
21	ab1	0.102	0.033	0.209
22	ab2	-0.001	-0.037	0.027
23	ab3	0.010	-0.079	0.088
24	totalIE	0.111	0.019	0.220
25	test1	-0.103	-0.215	-0.022

Serial Multiple Mediation



Serial Multiple Mediation

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

$$\begin{aligned}Y &= i_Y + b_1M_1 + b_2M_2 + c'X + e_Y \\M_2 &= i_{M2} + d_{21}M_1 + a_2X + e_{M2} \\M_1 &= i_{M1} + a_1X + e_{M1}\end{aligned}$$

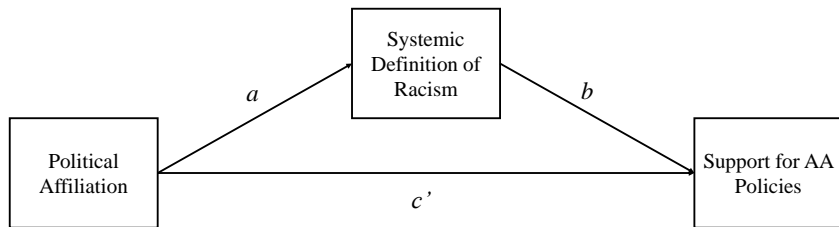
As with parallel mediator models, a serial mediator model with K mediator variables will required $K + 1$ separate equations.

Again, path modeling can make this task much simpler.

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

Serial Multiple Mediation

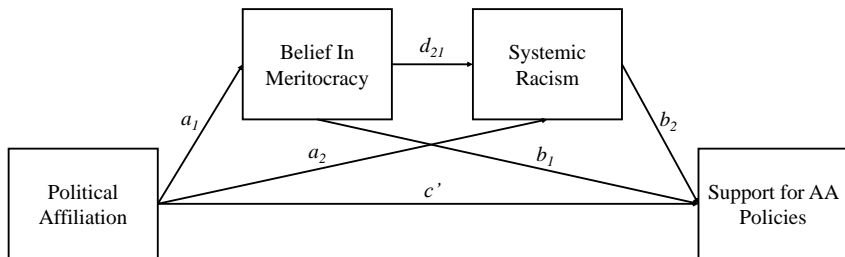
OK, back to our simple mediation example:



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

```
## 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 = 5000)
```

Example

```
## Check the results:
```

```
partSummary(out2.1, 7)
```

Regressions:

		Estimate	Std.Err	z-value	P(> z)
policy ~					
merit	(b1)	-0.008	0.147	-0.051	0.959
sysRac	(b2)	0.595	0.145	4.087	0.000
polAffil	(cp)	0.134	0.076	1.750	0.080
sysRac ~					
merit	(d21)	-0.301	0.112	-2.700	0.007
polAffil	(a2)	0.090	0.072	1.248	0.212
merit ~					
polAffil	(a1)	-0.266	0.061	-4.391	0.000

Example

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

Intercepts:

	Estimate	Std.Err	z-value	P(> z)
.policy	0.851	0.917	0.928	0.353
.sysRac	4.698	0.651	7.210	0.000
.merit	4.977	0.260	19.117	0.000

Variances:

	Estimate	Std.Err	z-value	P(> z)
.policy	0.987	0.165	5.973	0.000
.sysRac	0.689	0.092	7.513	0.000
.merit	0.719	0.115	6.261	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	0.044	1.209	0.226
fullIE	0.048	0.026	1.803	0.071
totalIE	0.103	0.049	2.095	0.036

Example

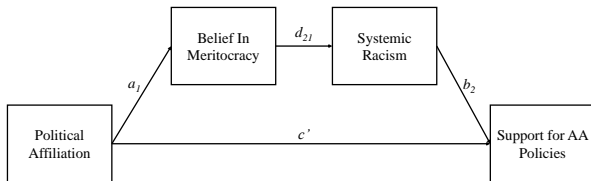
```
parameterEstimates(out2.1, boot.ci.type = "bca.simple") %>%  
  select(c("label", "est", "ci.lower", "ci.upper")) %>%  
  tail(4)
```

	label	est	ci.lower	ci.upper
15	ab1	0.002	-0.079	0.082
16	ab2	0.053	-0.030	0.147
17	fullIE	0.048	0.010	0.116
18	totalIE	0.103	0.011	0.209

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 = 5000)
```

Example

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

Regressions:

		Estimate	Std.Err	z-value	P(> z)
policy ~					
polAffil	(cp)	0.135	0.082	1.650	0.099
sysRac	(b2)	0.597	0.135	4.435	0.000
merit ~					
polAffil	(a1)	-0.266	0.061	-4.371	0.000
sysRac ~					
merit	(d21)	-0.367	0.097	-3.767	0.000

Intercepts:

	Estimate	Std.Err	z-value	P(> z)
.policy	0.807	0.563	1.433	0.152
.merit	4.977	0.260	19.106	0.000
.sysRac	5.337	0.394	13.539	0.000

Example

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

Variances:

	Estimate	Std.Err	z-value	P(> z)
.policy	0.987	0.167	5.895	0.000
.merit	0.719	0.113	6.333	0.000
.sysRac	0.705	0.092	7.635	0.000

Defined Parameters:

	Estimate	Std.Err	z-value	P(> z)
fullIE	0.058	0.025	2.303	0.021

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	-0.034	0.287
2	b2	0.597	0.304	0.846
3	a1	-0.266	-0.390	-0.152
4	d21	-0.367	-0.545	-0.159
5	fullIE	0.058	0.021	0.126

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

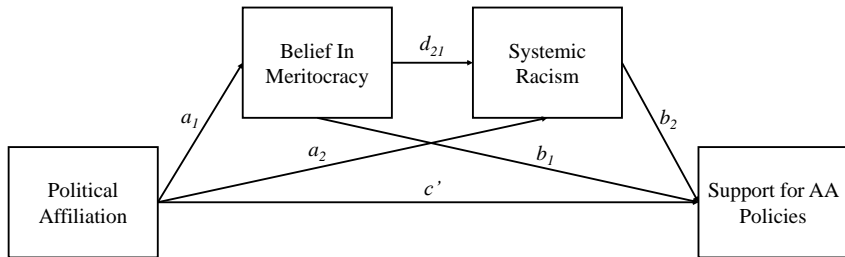
```
## Conduct a chi-squared difference test:
chiDiff <- fitMeasures(out2.3)["chisq"] - fitMeasures(out2.1)["chisq"]
dfDiff  <- fitMeasures(out2.3)["df"] - fitMeasures(out2.1)["df"]

pchisq(chiDiff, dfDiff, lower = FALSE)

      chisq
0.513125
```

Serial Multiple Mediation

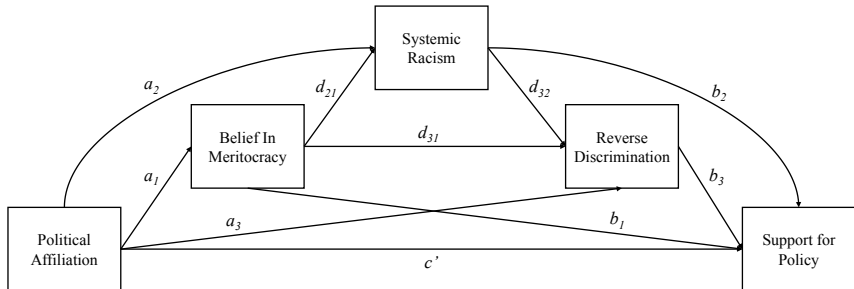
OK. We've supported an interesting hypothesis with the following model, but why stop there?



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?

$$\text{Policy} = i_Y + b_1\text{Merit} + b_2\text{SysRac} + b_3\text{RevDisc} + c'\text{PolAff} + e_Y$$

$$\text{RevDisc} = i_{M3} + d_{31}\text{Merit} + d_{32}\text{SysRac} + a_3\text{PolAff} + e_{M3}$$

$$\text{SysRac} = i_{M2} + d_{21}\text{Merit} + a_2\text{PolAff} + e_{M2}$$

$$\text{Merit} = i_{M1} + a_1\text{PolAff} + e_{M1}$$

Which produces the following set of specific indirect effects:

- a_1b_1
- a_2b_2
- a_3b_3
- $a_1d_{31}b_3$
- $a_1d_{21}b_2$
- $a_2d_{32}b_3$
- $a_1d_{21}d_{32}b_3$

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 = 5000)
```

Example

```
partSummary(out3.1, 7)
```

Regressions:

		Estimate	Std.Err	z-value	P(> z)
policy ~					
merit	(b1)	0.005	0.141	0.035	0.972
sysRac	(b2)	0.589	0.147	4.003	0.000
revDisc	(b3)	-0.026	0.082	-0.323	0.747
polAffil	(cp)	0.130	0.079	1.633	0.103
revDisc ~					
merit	(d31)	0.473	0.190	2.493	0.013
sysRac	(d32)	-0.196	0.240	-0.818	0.413
polAffil	(a3)	-0.149	0.133	-1.122	0.262
sysRac ~					
merit	(d21)	-0.301	0.112	-2.681	0.007
polAffil	(a2)	0.090	0.073	1.236	0.217
merit ~					
polAffil	(a1)	-0.266	0.061	-4.382	0.000

Example

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

Intercepts:

	Estimate	Std.Err	z-value	P(> z)
.policy	0.933	1.005	0.928	0.353
.revDisc	3.108	1.572	1.977	0.048
.sysRac	4.698	0.660	7.115	0.000
.merit	4.977	0.263	18.894	0.000

Variances:

	Estimate	Std.Err	z-value	P(> z)
.policy	0.985	0.165	5.968	0.000
.revDisc	2.361	0.307	7.683	0.000
.sysRac	0.689	0.093	7.387	0.000
.merit	0.719	0.112	6.434	0.000

Example

```
partSummary(out3.1, 10)
```

Defined Parameters:

	Estimate	Std.Err	z-value	P(> z)
ab1	-0.001	0.039	-0.034	0.973
ab2	0.053	0.044	1.199	0.231
ab3	0.004	0.017	0.232	0.816
partIE1	0.003	0.012	0.268	0.788
partIE2	0.047	0.026	1.796	0.072
partIE3	0.000	0.003	0.160	0.873
fullIE	0.000	0.002	0.177	0.859
totalIE	0.107	0.052	2.055	0.040

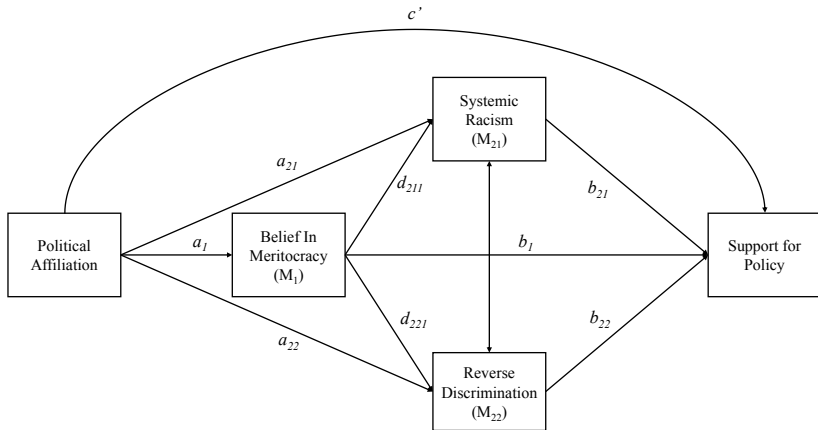
Example

```
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	0.076
22	ab2	0.053	-0.029	0.146
23	ab3	0.004	-0.018	0.060
24	partIE1	0.003	-0.014	0.039
25	partIE2	0.047	0.012	0.121
26	partIE3	0.000	-0.002	0.016
27	fullIE	0.000	-0.002	0.011
28	totalIE	0.107	0.012	0.219

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 = 5000)
```

Example

```
partSummary(out4.1, 7)
```

Regressions:

		Estimate	Std.Err	z-value	P(> z)
policy ~					
merit	(b1)	0.005	0.143	0.035	0.972
sysRac	(b21)	0.589	0.149	3.969	0.000
revDisc	(b22)	-0.026	0.080	-0.329	0.742
polAffl	(cp)	0.130	0.079	1.634	0.102
sysRac ~					
merit	(d211)	-0.301	0.110	-2.752	0.006
polAffl	(a21)	0.090	0.072	1.249	0.212
revDisc ~					
merit	(d221)	0.532	0.189	2.822	0.005
polAffl	(a22)	-0.167	0.136	-1.227	0.220
merit ~					
polAffl	(a1)	-0.266	0.061	-4.358	0.000

Example

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

Covariances:

	Estimate	Std.Err	z-value	P(> z)
.sysRac ~~				
.revDisc	-0.135	0.161	-0.841	0.400

Intercepts:

	Estimate	Std.Err	z-value	P(> z)
.policy	0.933	1.014	0.921	0.357
.sysRac	4.698	0.646	7.267	0.000
.revDisc	2.187	1.167	1.873	0.061
.merit	4.977	0.264	18.866	0.000

Variances:

	Estimate	Std.Err	z-value	P(> z)
.policy	0.985	0.166	5.940	0.000
.sysRac	0.689	0.092	7.499	0.000
.revDisc	2.388	0.306	7.793	0.000
.merit	0.719	0.112	6.421	0.000

Example

```
partSummary(out4.1, 11)
```

Defined Parameters:

	Estimate	Std.Err	z-value	P(> z)
ab1	-0.001	0.040	-0.034	0.973
ab21	0.053	0.044	1.198	0.231
ab22	0.004	0.018	0.249	0.803
fullIE21	0.047	0.026	1.828	0.068
fullIE22	0.004	0.013	0.277	0.782
totalIE	0.107	0.052	2.067	0.039

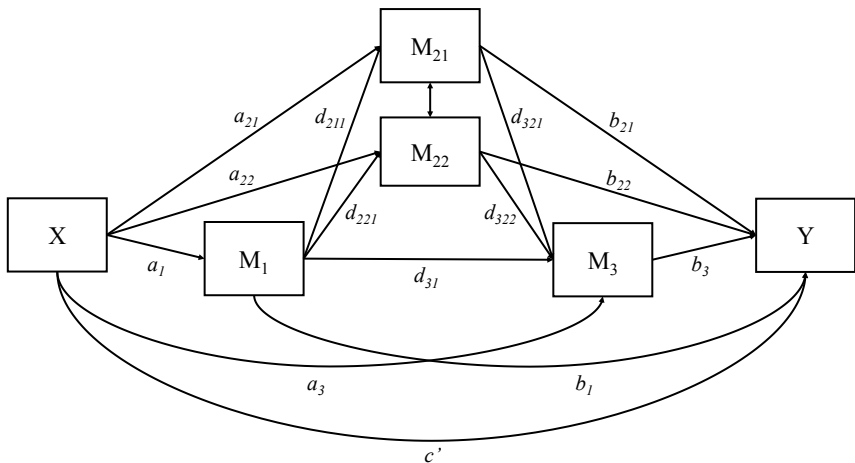
Example

```
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	-0.084	0.077
22	ab21	0.053	-0.024	0.157
23	ab22	0.004	-0.018	0.062
24	fullIE21	0.047	0.011	0.117
25	fullIE22	0.004	-0.016	0.042
26	totalIE	0.107	0.013	0.224

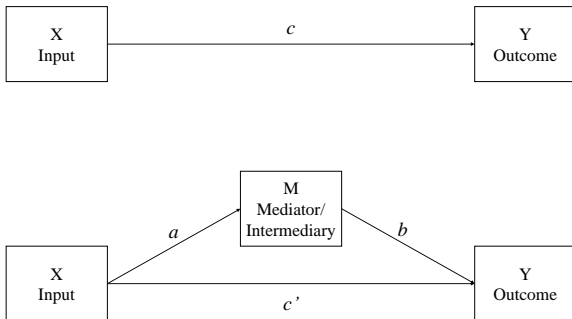
Practice

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



Boring Model

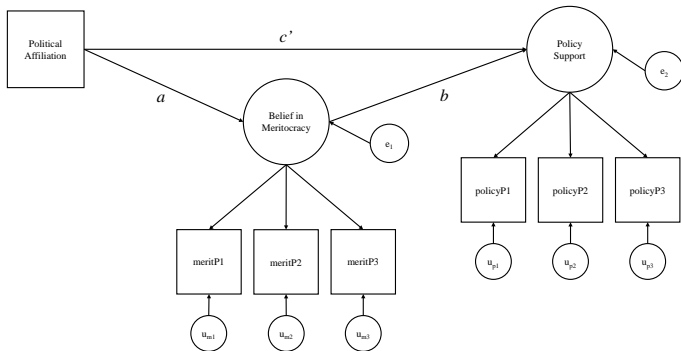
So far, all of our models have looked something like:



But there is no reason that we need to restrict ourselves to mucking around with observed variables.

Better Model

We can (and should) test for indirect effects using full SEMs such as as:



Measurement error can be a big problem for mediation analysis, so latent variable modeling is highly recommended.

Example

```
dat1 <- readRDS("../data/adamsKlpsData.rds") %>% select(-merit, -policy)

## Specify the CFA model:
mod5.1 <- '
merit =~ meritP1 + meritP2 + meritP3
policy =~ policyP1 + policyP2 + policyP3
'

## Fit the CFA and check model:
out5.1 <- cfa(mod5.1, data = dat1, std.lv = TRUE)

## Check model fit:
fitMeasures(out5.1,
             c("chisq", "df", "pvalue", "cfi", "tli", "rmsea", "srmr"))

```

chisq	df	pvalue	cfi	tli	rmsea	srmr
16.869	8.000	0.031	0.922	0.853	0.113	0.065

Example

```
partSummary(out5.1, 7)
```

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)
merit =~				
meritP1	0.690	0.134	5.155	0.000
meritP2	0.968	0.142	6.830	0.000
meritP3	0.748	0.137	5.458	0.000
policy =~				
policyP1	0.851	0.186	4.570	0.000
policyP2	0.996	0.167	5.967	0.000
policyP3	1.121	0.177	6.339	0.000

Example

```
partSummary(out5.1, 8:9)
```

Covariances:

	Estimate	Std.Err	z-value	P(> z)
merit ~~ policy	-0.336	0.131	-2.563	0.010

Variances:

	Estimate	Std.Err	z-value	P(> z)
.meritP1	0.865	0.165	5.248	0.000
.meritP2	0.445	0.201	2.211	0.027
.meritP3	0.833	0.172	4.857	0.000
.policyP1	1.836	0.324	5.671	0.000
.policyP2	0.942	0.256	3.683	0.000
.policyP3	0.857	0.297	2.882	0.004
merit	1.000			
policy	1.000			

Example

```
## Specify the structural model:
mod5.2 <- '
merit  =~ meritP1 + meritP2 + meritP3
policy =~ policyP1 + policyP2 + policyP3

policy ~ b*merit + polAffil
merit  ~ a*polAffil

ab := a*b
'

## Fit the structural model and test the indirect effect:
out5.2 <-
  sem(mod5.2, data = dat1, std.lv = TRUE, se = "boot", bootstrap = 2500)
```

Example

```
partSummary(out5.2, 7:8)
```

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)
merit =~				
meritP1	0.545	0.124	4.396	0.000
meritP2	0.858	0.132	6.506	0.000
meritP3	0.609	0.116	5.252	0.000
policy =~				
policyP1	0.799	0.194	4.107	0.000
policyP2	0.924	3.111	0.297	0.766
policyP3	1.001	1.708	0.586	0.558

Regressions:

		Estimate	Std.Err	z-value	P(> z)
policy ~					
merit	(b)	-0.195	0.210	-0.929	0.353
polAffil		0.169	0.148	1.148	0.251
merit ~					
polAffil	(a)	-0.411	0.102	-4.034	0.000

Example

```
partSummary(out5.2, 9:10)
```

Variances:

	Estimate	Std.Err	z-value	P(> z)
.meritP1	0.922	0.182	5.078	0.000
.meritP2	0.341	0.226	1.507	0.132
.meritP3	0.869	0.183	4.755	0.000
.policyP1	1.801	0.337	5.338	0.000
.policyP2	0.918	166.242	0.006	0.996
.policyP3	0.922	103.049	0.009	0.993
.merit	1.000			
.policy	1.000			

Defined Parameters:

	Estimate	Std.Err	z-value	P(> z)
ab	0.080	0.109	0.732	0.464

Example

```
parameterEstimates(out5.2, boot.ci.type = "bca.simple") %>%  
  select(c("label", "est", "ci.lower", "ci.upper")) %>%  
  filter(label != "")
```

	label	est	ci.lower	ci.upper
1	b	-0.195	-0.593	0.214
2	a	-0.411	-0.642	-0.240
3	ab	0.080	-0.091	0.274

Interpretation of Indirect Effects

Indirect effects are composed parameters, but they can be interpreted independently of their constituent paths.

- The $X \rightarrow M \rightarrow Y$ indirect effect, ab , is interpreted as:
 - The expected change in Y for a unit change in X that is transmitted indirectly through M .
 - For a unit change in X , Y is expected to change by ab units, indirectly through M .
 - Participants who differ by one unit on X are expected to differ by ab units on Y as a result of the effect of X on M which, in turn, affects Y .
- The interpretation/scaling of the indirect effect is entirely defined by the input, X , and outcome, Y .
 - The scaling of the intermediary variable, M , does not affect the interpretation of the indirect effect.

References

- Baron, R. M., & Kenny, D. A. (1986). The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of Personality and Social Psychology*, 51(6), 1173.
- Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1), 1–26. doi: 10.1214/aos/1176344552
- Sobel, M. E. (1982). Asymptotic confidence intervals for indirect effects in structural equation models. *Sociological Methodology*, 13(1982), 290–312.

