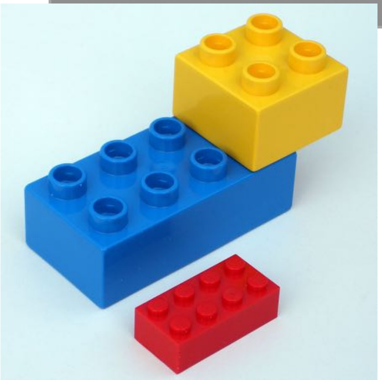
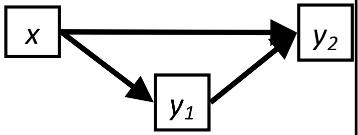
Introduction to Local Estimation



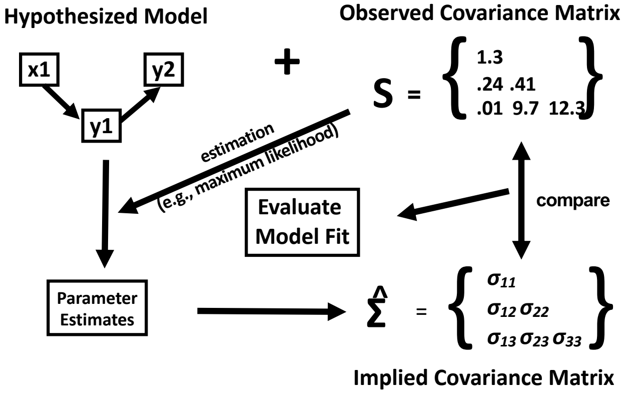


lm(y2 ~ x+y1)
lm(y1 ~ x)

Overview

1. Covariance vs. piecewise SEM
2. Assessing Model Fit: Tests of directed separation
3. Introduction to *piecewiseSEM*
4. Further adventures into *piecewiseSEM*

Covariance Based SEM



$$S = \left\{ \begin{matrix} 1.3 & & \\ .24 & .41 & \\ .01 & 9.7 & 12.3 \end{matrix} \right\}$$

$$\hat{\Sigma} = \left\{ \begin{matrix} \sigma_{11} & & \\ \sigma_{12} & \sigma_{22} & \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{matrix} \right\}$$

Evaluating Covariance-Based Models

The log likelihood ratio, F_{ML} follows $\chi 2$ distribution such that

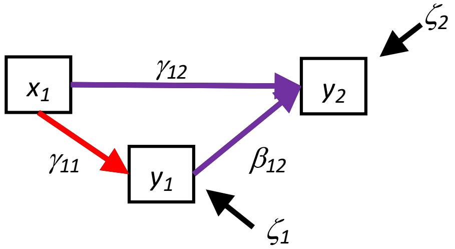
$$\chi^2 = (n-1)F_{ML}$$

- ***Note scaling by sample size***
- ***Large χ^2 implies LACK of fit***

Assumptions of Covariance Based Estimation

- Multivariate normality
  - Of residuals
  - Of data
- Linearity
- All correlations accounted for in model structure
- But… latent variables make magic possible

Comparison. Piecewise: Graphs to equations



$$y_1 = \gamma_{11}x_1 + \zeta_1$$

$$y_2 = \gamma_{12}x_1 + \beta_{12}y_1 + \zeta_2$$

Model fit evaluated via tests of conditional independence

Comparison. Traditional vs. piecewise SEM

Traditional SEM

Piecewise SEM

Comparison. Traditional vs. piecewise SEM

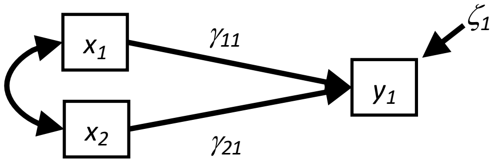
Variance-covariance	Piecewise
Single (global) variance-covariance matrix estimated	Variance-covariance matrices estimated for each endogenous variable
Simultaneous solution (computationally intensive)	Multiple solutions (modularized)
Fit to normal distribution	Incorporates various distributions (Poisson, Gamma, etc.)
Assumes independence	Can model non-independence (blocked, temporal, spatial, etc.)
Latent & composite variables	No latent or composite variables (yet*)
Non-recursive (cyclic)	Only for recursive (acyclic)
Multi-group models	Can estimate random components, but no formal χ^2 test

Multiple Regression: Hidden Assumptions

equation
form

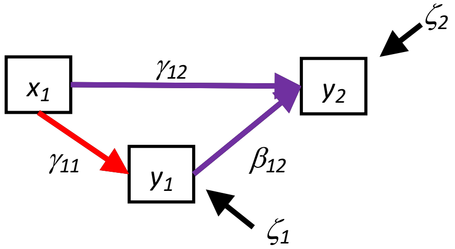
$$y_1 = \gamma_{11}x_1 + \gamma_{21}x_2 + \zeta_1$$

graphical
form



- To estimate γ11 and γ21 you need to control for the unanalyzed correlation between x1 and x2.
- In a factorial experiment, this correlation is 0!

Comparison. Graphs to equations



$$y_1 = \gamma_{11}x_1 + \zeta_1$$

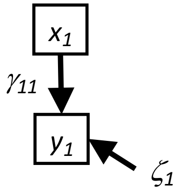
$$y_2 = \gamma_{12}x_1 + \beta_{12}y_1 + \zeta_2$$

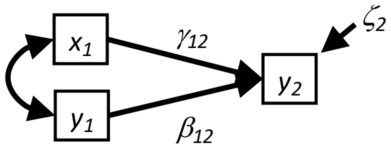
Comparison. Graphs to equations

equation
form

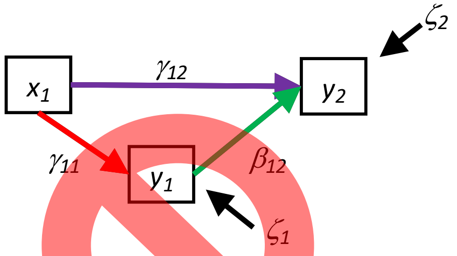
$$y_1 = \gamma_{11}x_1 + \zeta_1$$

$$y_2 = \gamma_{12}x_1 + \beta_{12}y_1 + \zeta_2$$





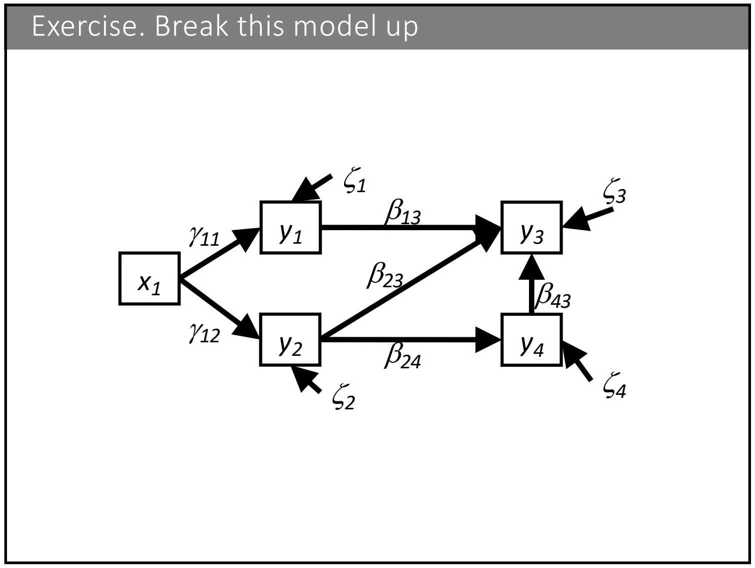
Comparison. Graphs to equations



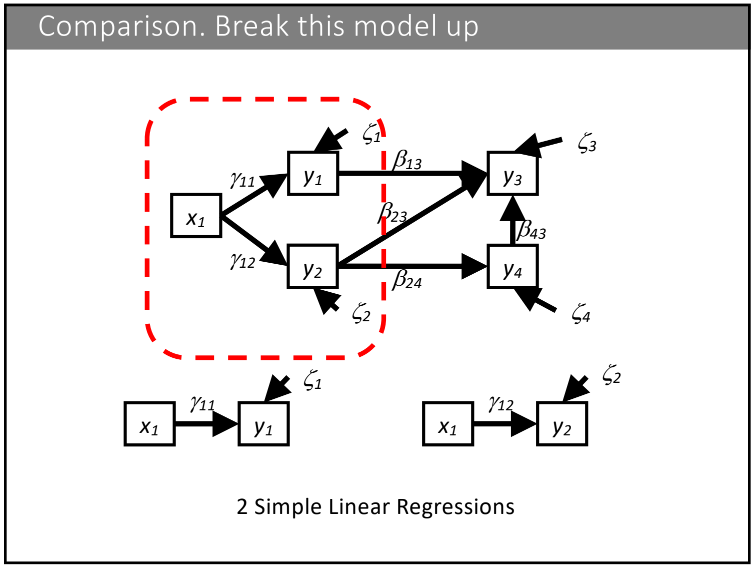
$$y_1 = \gamma_{11}x_1 + \zeta_1$$

$$y_2 = \beta_{12}y_1 + \zeta_2$$

$$y_2 = \gamma_{12}x_1 + \zeta_2$$

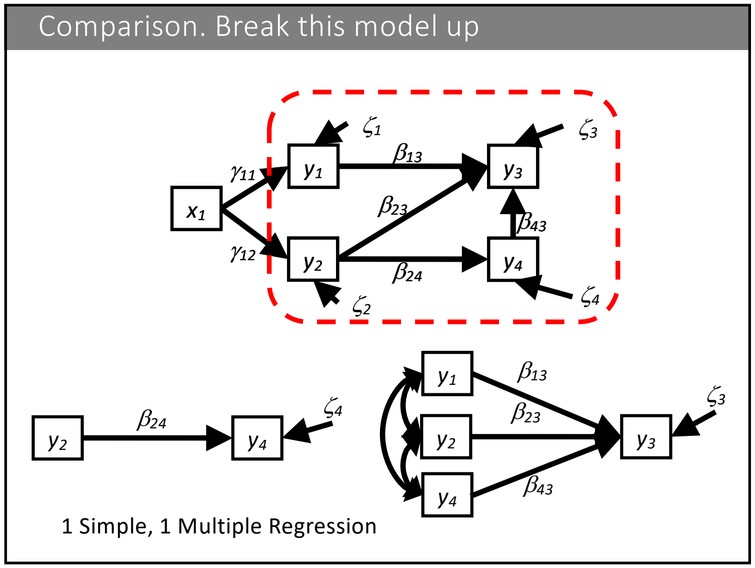


Exercise. Break this model up



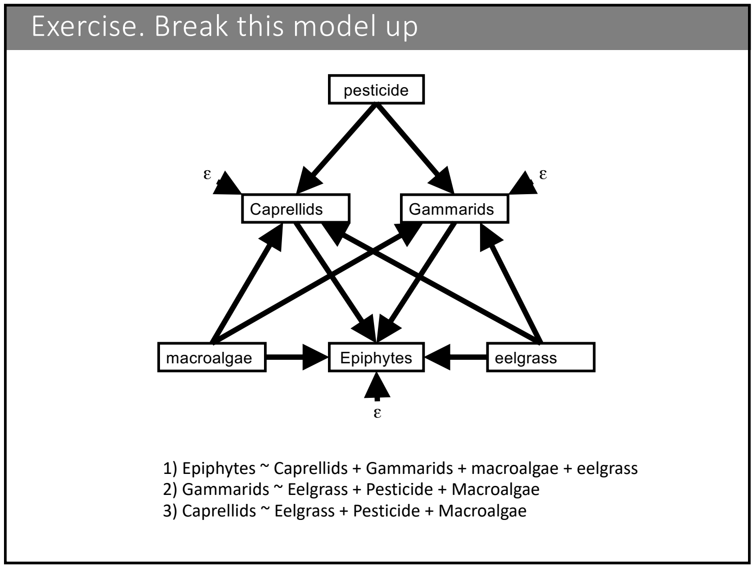
Comparison. Break this model up

2 Simple Linear Regressions



Comparison. Break this model up

1 Simple, 1 Multiple Regression



Exercise. Break this model up

1) Epiphytes ~ Caprellids + Gammarids + macroalgae + eelgrass
2) Gammarids ~ Eelgrass + Pesticide + Macroalgae
3) Caprellids ~ Eelgrass + Pesticide + Macroalgae

ACTIVITY

- Take your own causal diagram
- Break it up into the component models

Overview

1. Covariance vs. piecewise SEM
2. Assessing Model Fit: Tests of directed separation
3. Introduction to *piecewiseSEM*
4. Further adventures into *piecewiseSEM*

Directed Separation. Model fit

Does the model fit the data?

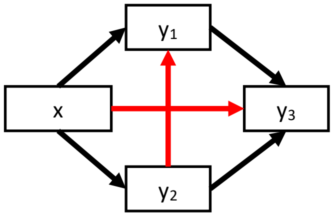
=

Does the model represent the data well?

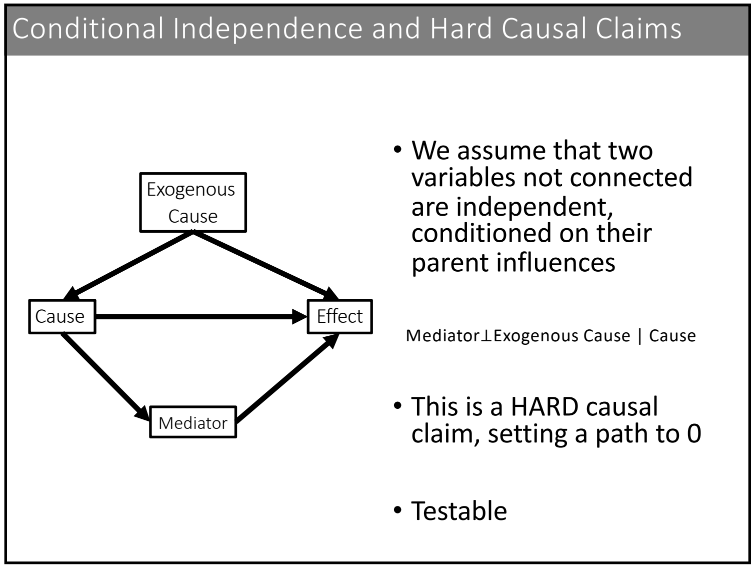
=

Are we missing important information?

Directed Separation. Model fit



Did we get the topology right or are there unrecognized significant relationships?

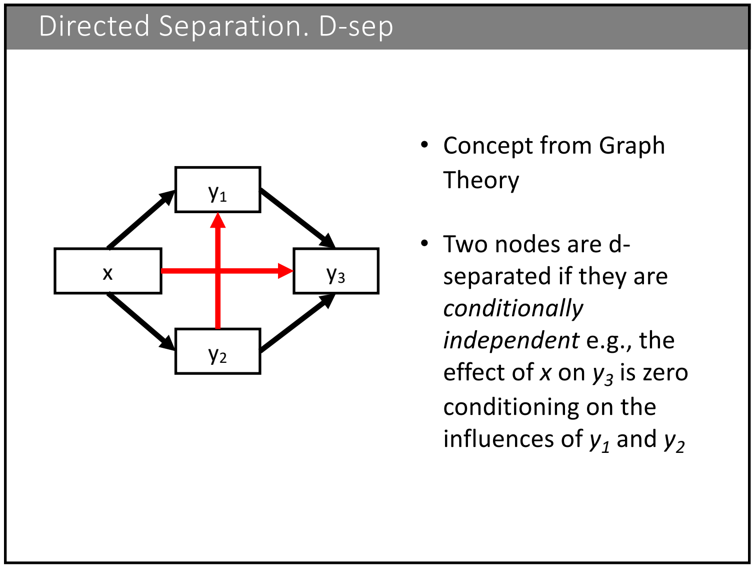


Conditional Independence and Hard Causal Claims

- We assume that two variables not connected are independent, conditioned on their parent influences

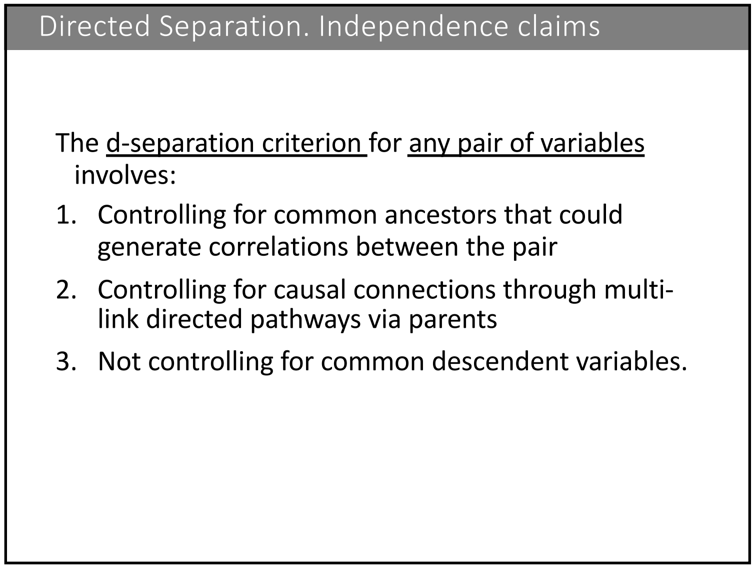
$\text{Mediator} \perp \text{Exogenous Cause} \mid \text{Cause}$

- This is a HARD causal claim, setting a path to 0
- Testable



Directed Separation. D-sep

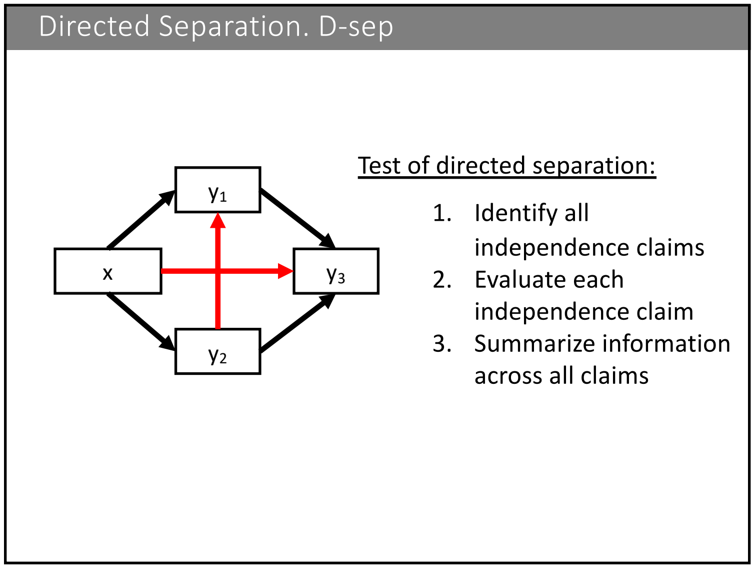
- Concept from Graph Theory
- Two nodes are d-separated if they are *conditionally independent* e.g., the effect of *x* on y_3 is zero conditioning on the influences of y_1 and y_2



Directed Separation. Independence claims

The d-separation criterion for any pair of variables involves:

1. Controlling for common ancestors that could generate correlations between the pair
2. Controlling for causal connections through multi-link directed pathways via parents
3. Not controlling for common descendent variables.

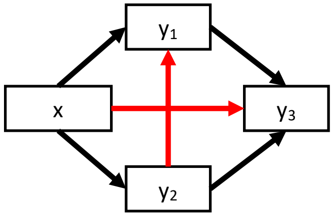


Directed Separation. D-sep

Test of directed separation:

1. Identify all independence claims
2. Evaluate each independence claim
3. Summarize information across all claims

Directed Separation. The Basis Set

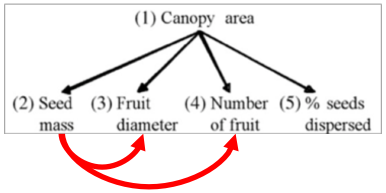


Basis set = the smallest possible set of independence claims from a graph

1. $x \perp y_3 \mid (y_1, y_2)$
2. $y_1 \perp y_2 \mid (x)$

Directed Separation. Deriving the basis set

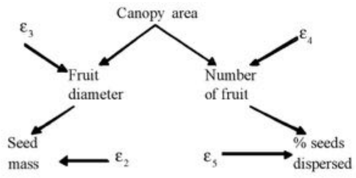
What is the basis set?



1. mass ⊥ dia | (canopy)
2. mass ⊥ # | (canopy)
3. mass ⊥ %| (canopy)
4. dia ⊥ # | (canopy)
5. dia ⊥ % | (canopy)
6. % ⊥ # | (canopy)

Directed Separation. Deriving the basis set

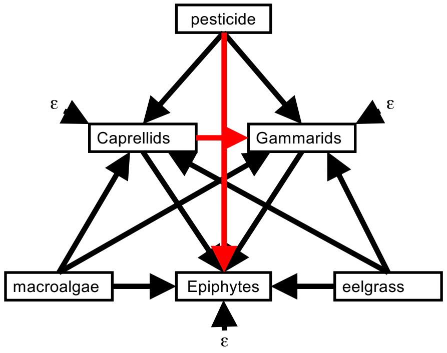
What is the basis set?



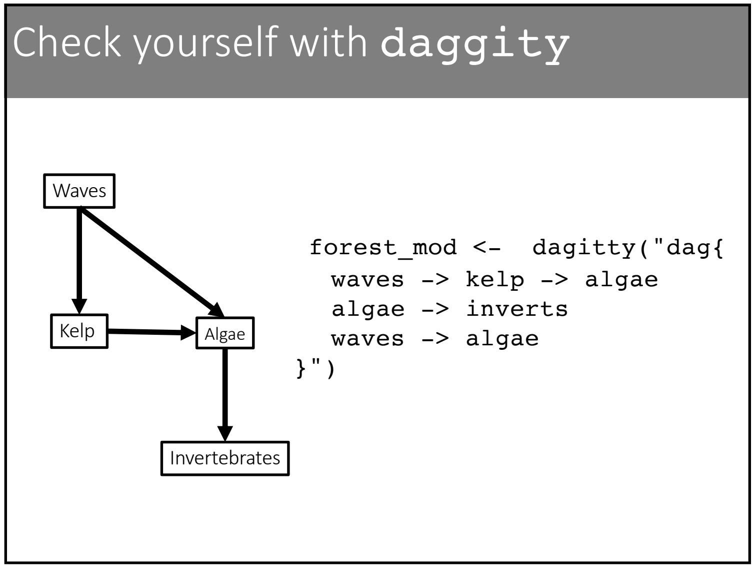
1. dia ⊥ # | (canopy)
2. dia ⊥ % | (canopy, #)
3. canopy⊥ mass | (dia)
4. mass ⊥ #| (dia, canopy)
5. mass ⊥ % | (dia, #)
6. canopy ⊥ % | (#)

Directed Separation. Deriving the basis set

What is the basis set?

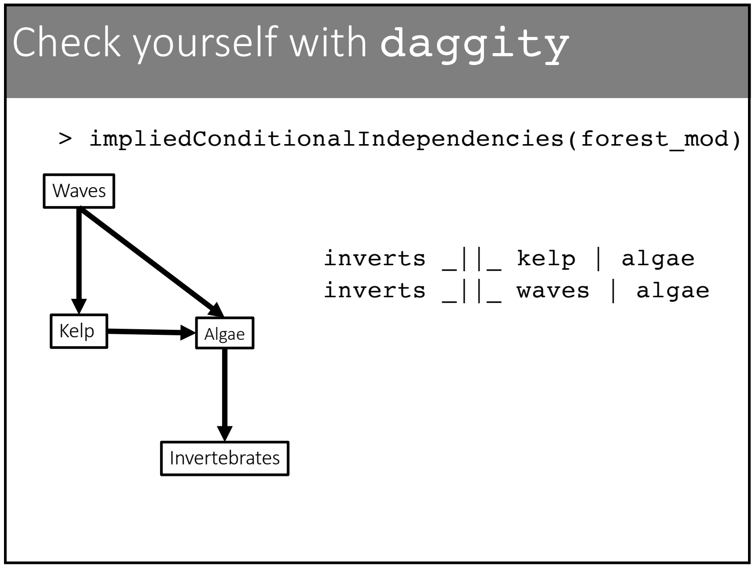


1. Pesticide ⊥ epiphytes (macroalgae, eelgrass, caprellids, gammarids)
2. Caprellids ⊥ gammarids (macroalage, eelgrass, pesticide)



Check yourself with **daggity**

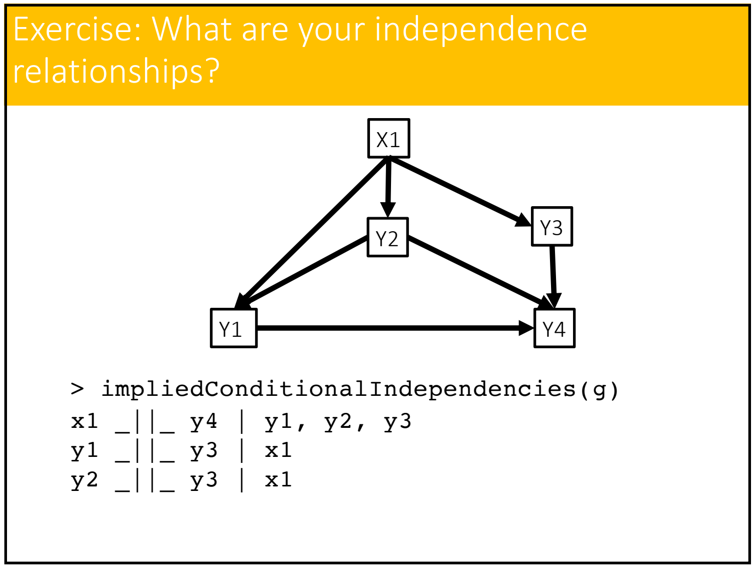
```
forest_mod <-  dagitty("dag{
  waves -> kelp -> algae
  algae -> inverts
  waves -> algae
}")
```



Check yourself with **daggity**

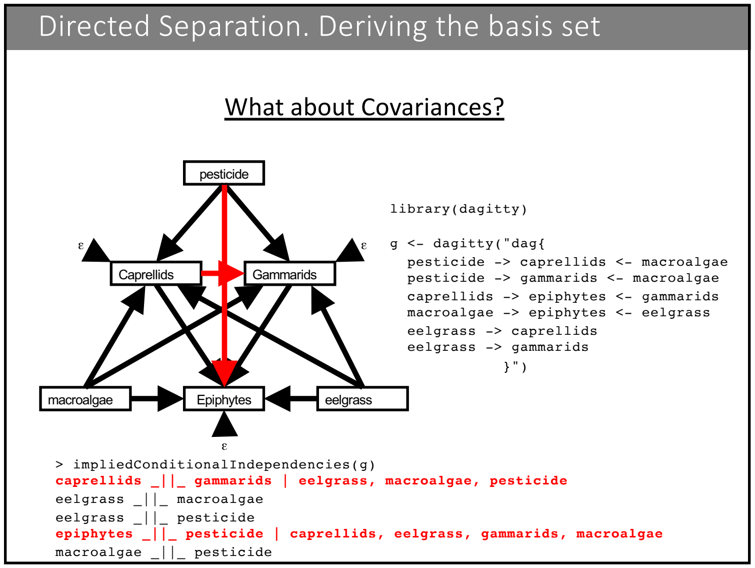
```
> impliedConditionalIndependencies(forest_mod)
```

```
inverts _||_ kelp | algae
inverts _||_ waves | algae
```



Exercise: What are your independence relationships?

```
> impliedConditionalIndependencies(g)
x1 _||_ y4 | y1, y2, y3
y1 _||_ y3 | x1
y2 _||_ y3 | x1
```



Directed Separation. Deriving the basis set

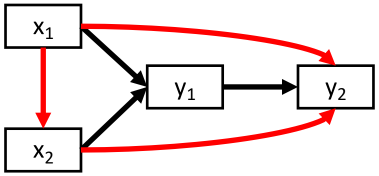
What about Covariances?

```
library(dagitty)

g <- dagitty("dag{
  pesticide -> caprellids <- macroalgae
  pesticide -> gammarids <- macroalgae
  caprellids -> epiphytes <- gammarids
  macroalgae -> epiphytes <- eelgrass
  eelgrass -> caprellids
  eelgrass -> gammarids
           }")
```

```
> impliedConditionalIndependencies(g)
caprellids _||_ gammarids | eelgrass, macroalgae, pesticide
eelgrass _||_ macroalgae
eelgrass _||_ pesticide
epiphytes _||_ pesticide | caprellids, eelgrass, gammarids, macroalgae
macroalgae _||_ pesticide
```

Directed Separation. A note

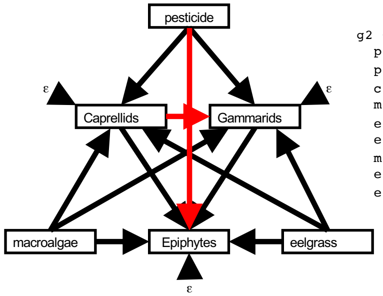
- Basis set generally excludes relationships among exogenous variables



1. $x_1 \perp y_2 \mid (y_1)$
2. $x_2 \perp y_2 \mid (y_1)$

Directed Separation. Deriving the basis set

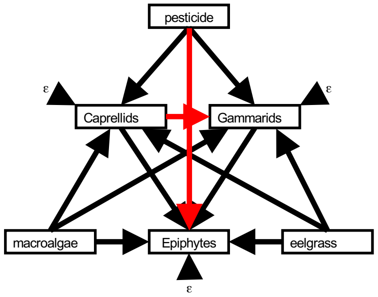
Coding Covarianecs



```
g2 <- dagitty("dag{
  pesticide -> caprellids <- macroalgae
  pesticide -> gammarids <- macroalgae
  caprellids -> epiphytes <- gammarids
  macroalgae -> epiphytes <- eelgrass
  eelgrass -> caprellids
  eelgrass -> gammarids
  macroalgae <-> pesticide
  eelgrass <-> macroalgae
  eelgrass <-> pesticide
            }")
```

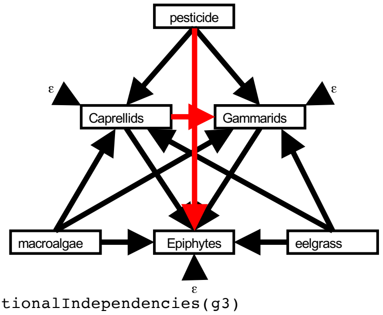
```
> impliedConditionalIndependencies(g2)

caprellids _||_ gammarids | eelgrass, macroalgae, pesticide
eelgrass _||_ pesticide
```

Using lavaan shortcuts with `dagitty`



```
library(lavaan)

mod <- "
  epiphytes ~ eelgrass + macroalgae + caprellids + gammarids
  caprellids ~ eelgrass + macroalgae + pesticide
  gammarids ~ eelgrass + macroalgae + pesticide
"

g3 <- lavaanToGraph(lavaanify(mod))
```

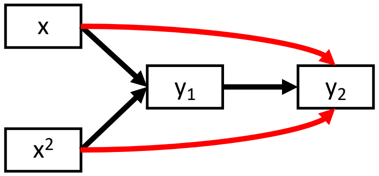
Using lavaan shortcuts with `dagitty`



```
> impliedConditionalIndependencies(g3)

caprellids _||_ gammarids | eelgrass, macroalgae, pesticide
eelgrass _||_ pesticide
```

Directed Separation. Nonlinearities

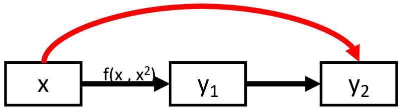
- Basis set generally excludes non-linear components (polynomials)



1. $x \perp y_2 \mid (y_1)$

Directed Separation. Nonlinearities

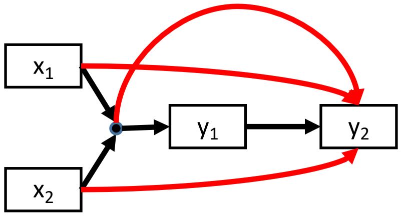
- Why? Paths are not inherently linear



1. $x \perp y_2 \mid (y_1)$

Directed Separation. Nonlinearities

- Basis set generally excludes non-linear components (interactions)

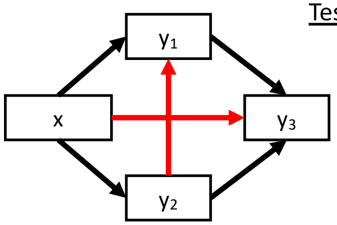


1. $x_1 \perp y_2 \mid (y_1)$
2. $x_2 \perp y_2 \mid (y_1)$

ACTIVITY

- Take your causal diagram
- Derive the basis set
- Try it by hand before `dagitty`

Directed Separation. D-sep



Test of directed separation:

1. Identify all independence claims
2. **Evaluate each independence claim**
3. Summarize information across all claims

Directed Separation. Fisher’s *C*

- Summarize independence claims across basis set:

$$C = -2*\sum \ln(p_i)$$

p_i = the *P*-values of all tests of conditional independence

- *C* has a χ^2-square distribution with 2*k* degrees of freedom
- *k* = # of elements of the basis set

Directed Separation. Fisher’s *C*

What is $p < 0.05$?

- You are likely missing some associations
- You *reject* this model
- The way forward: adding links or different model structure? (look at d-sep tests)
- To re-iterate, ***p* ≥ 0.05 is GOOD**

Directed Separation. Model selection

- Fisher’s *C* can be used to construct model AIC:

$$AIC = C + 2K$$

- *K* = # of likelihood parameters estimated (not to be confused with *k*)
- Can be extended to small sample size:

$$AICc = C + 2K(n / (n - K - 1))$$

Directed Separation. Complexity and sample size

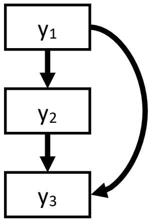
- The *t*-rule applies in generating model d.f. for the *C* statistic
- Shipley suggests need only enough individual d.f. to fit each component model
  - This is OK, until you try and do prediction
- Or, *d*-rule (Grace et al 2015):
  - *d* = # of samples / # of pathways
  - *d* ≥ 5

Directed Separation. Sample size

- More is always better…
  - Low sample size leads to non-significant d-sep tests
  - Low sample size also leads to non-significant path coefficients
  - End up with a 'good fitting' model that says *nothing*

Directed Separation. Saturated models

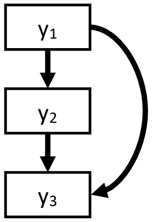
What is the basis set?



- There is no basis set for a saturated model (all paths are represented)
- No d-sep tests, so, AIC = 2K

Directed Separation. Saturated models

What is the basis set?



Options:

1. Remove the mediating variable y_2 and test submodel
2. Rely on other indicators of model fit (e.g., R^2)

A global test is not the be all-end all of models

Overview

1. Covariance vs. piecewise SEM
2. Assessing Model Fit: Tests of directed separation
3. Introduction to *piecewiseSEM*
4. Further adventures into *piecewiseSEM*

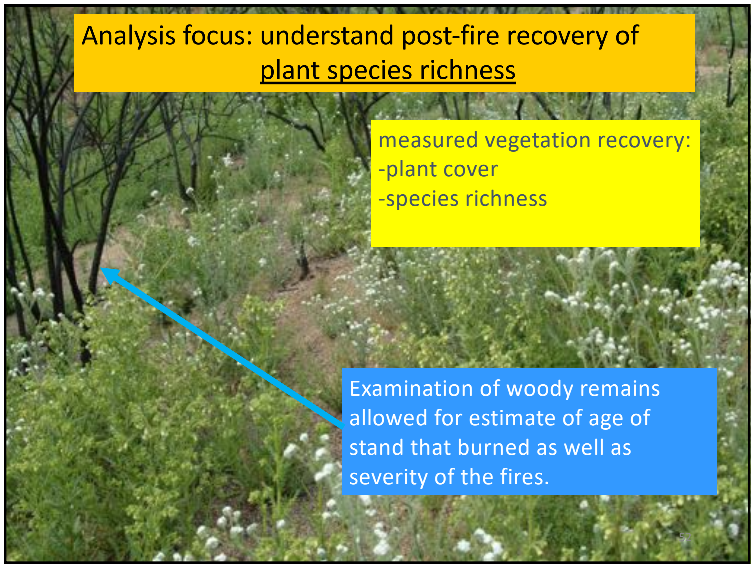
piecewiseSEM.

piecewiseSEM: Piecewise structural equation modeling in R for ecology, evolution, and systematics

```
install.packages(“devtools”)
library(devtools)
install_github(“jslefche/piecewiseSEM@devel”)
library(piecewiseSEM)
```

Mediation in Analysis of Post-Fire Recovery of Plant Communities in California Shrublands

Five year study of wildfires in Southern California in 1993.
90 plots (20 x 50m)



Analysis focus: understand post-fire recovery of <u>plant species richness</u>

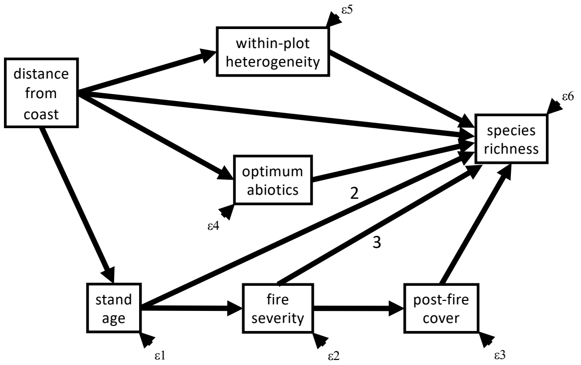
measured vegetation recovery:
-plant cover
-species richness

Examination of woody remains allowed for estimate of age of stand that burned as well as severity of the fires.

Other factors measured included:
- local abiotic conditions (aspect, soils)
- spatial heterogeneity
- landscape-level conditions (location, elevation)

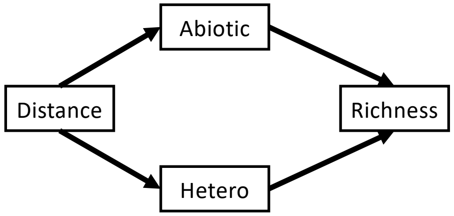
53

Realized Models with Data



Model 1 - all unnumbered paths
Model 2 – includes additional mechanism: loss of species from seedbank in older stands
Model 3 – includes selective destruction of seeds in seedbank in severe fires

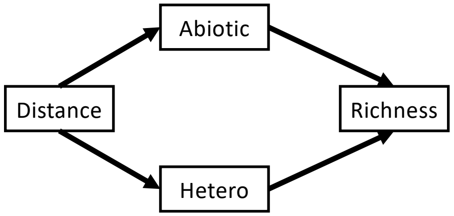
piecewiseSEM. Keeley example

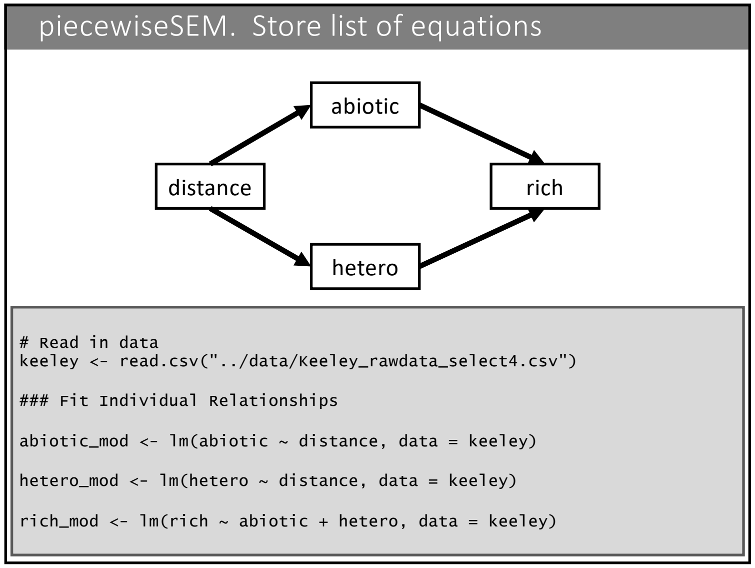


1. Create list of structured equations
2. Conduct d-sep tests (evaluate fit)
3. Extract coefficients

piecewiseSEM. Keeley example

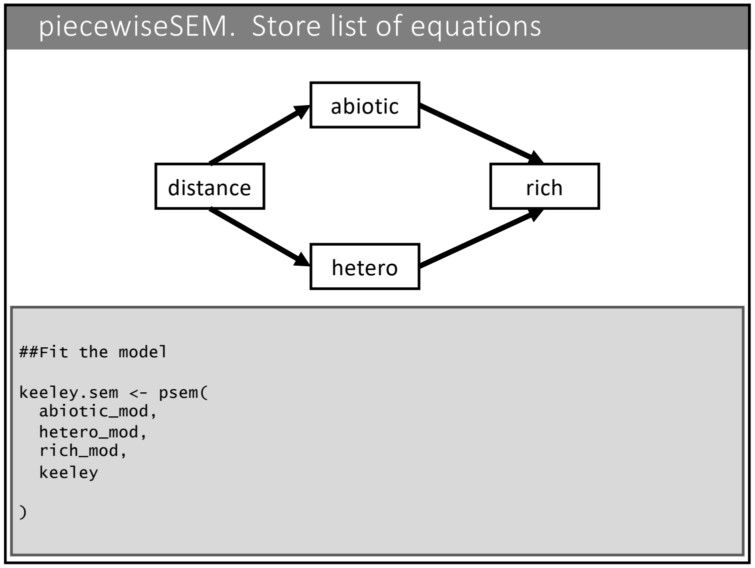
Break this model up into component models





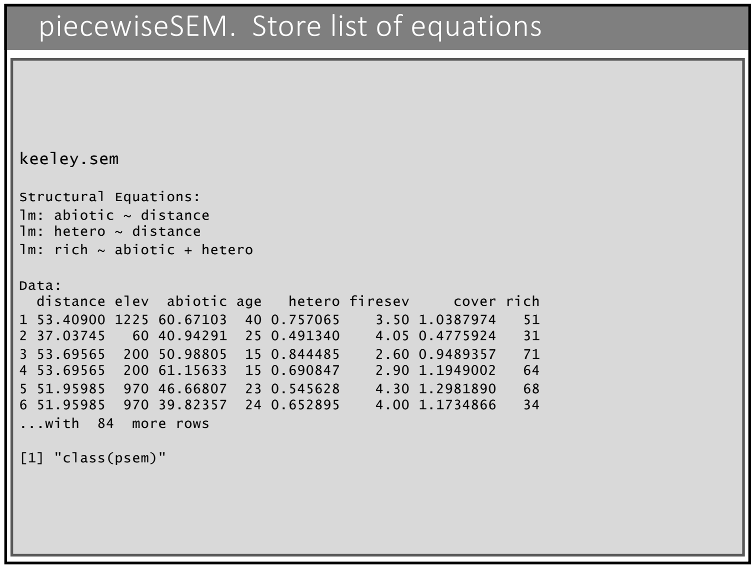
piecewiseSEM. Store list of equations

```
# Read in data
keeley <- read.csv("../data/Keeley_rawdata_select4.csv")

### Fit Individual Relationships

abiotic_mod <- lm(abiotic ~ distance, data = keeley)

hetero_mod <- lm(hetero ~ distance, data = keeley)

rich_mod <- lm(rich ~ abiotic + hetero, data = keeley)
```



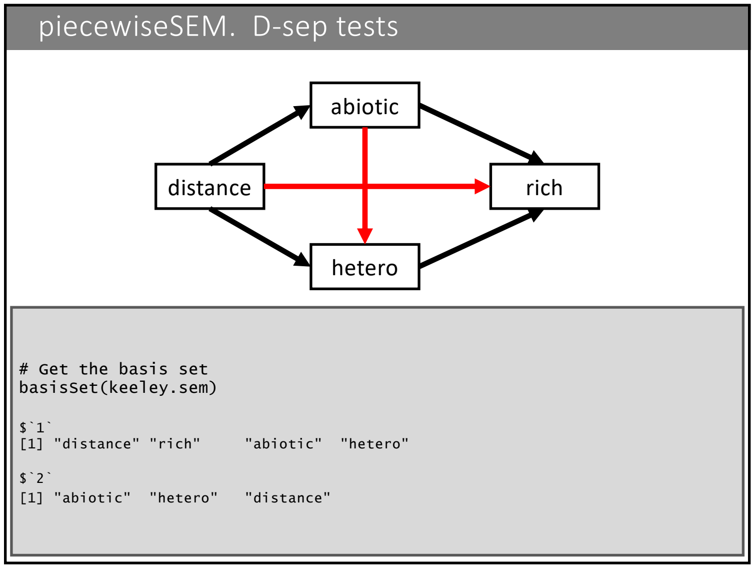
piecewiseSEM. Store list of equations

```
##Fit the model

keeley.sem <- psem(
  abiotic_mod,
  hetero_mod,
  rich_mod,
  keeley

)
```



piecewiseSEM. Store list of equations

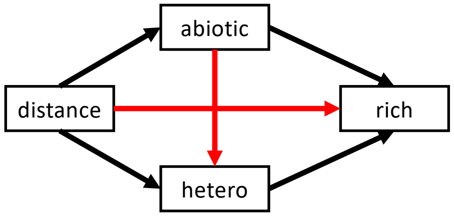
```
keeley.sem

Structural Equations:
lm: abiotic ~ distance
lm: hetero ~ distance
lm: rich ~ abiotic + hetero

Data:
  distance elev  abiotic age   hetero firesev     cover rich
1 53.40900 1225 60.67103  40 0.757065    3.50 1.0387974   51
2 37.03745   60 40.94291  25 0.491340    4.05 0.4775924   31
3 53.69565  200 50.98805  15 0.844485    2.60 0.9489357   71
4 53.69565  200 61.15633  15 0.690847    2.90 1.1949002   64
5 51.95985  970 46.66807  23 0.545628    4.30 1.2981890   68
6 51.95985  970 39.82357  24 0.652895    4.00 1.1734866   34
...with  84  more rows

[1] "class(psem)"
```



piecewiseSEM. D-sep tests

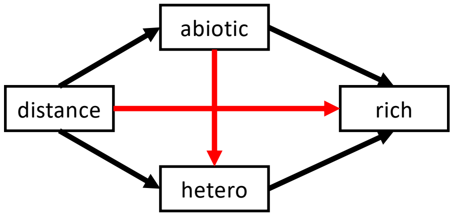
```
# Get the basis set
basisSet(keeley.sem)

$`1`
[1] "distance" "rich"     "abiotic"  "hetero"

$`2`
[1] "abiotic"  "hetero"   "distance"
```

piecewiseSEM. D-sep tests



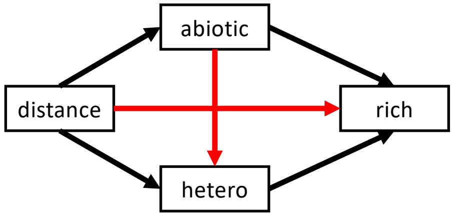
```
claim1 <- lm(rich ~ distance + abiotic + hetero, keeley)

coefs(claim1)

  Response Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate
1     rich  distance   0.6404    0.1565 86     8.0933  0.0001       0.3743 ***
2     rich   abiotic   0.5233    0.1756 86     2.9793  0.0038       0.2660  **
3     rich    hetero  33.4010   11.1187 86     3.0040  0.0035       0.2539  **
```

piecewiseSEM. D-sep tests



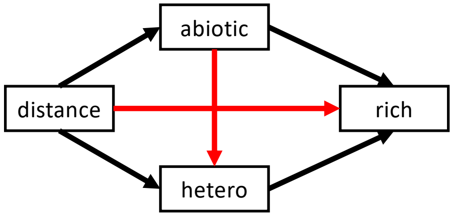
```
claim2 <- lm(hetero ~ abiotic + distance, keeley)

coefs(claim2)

  Response Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate
1   hetero   abiotic   0.0022    0.0017 87     1.3296  0.1871       0.1491
2   hetero  distance   0.0036    0.0015 87     2.4742  0.0153       0.2774 *
```

piecewiseSEM. D-sep tests

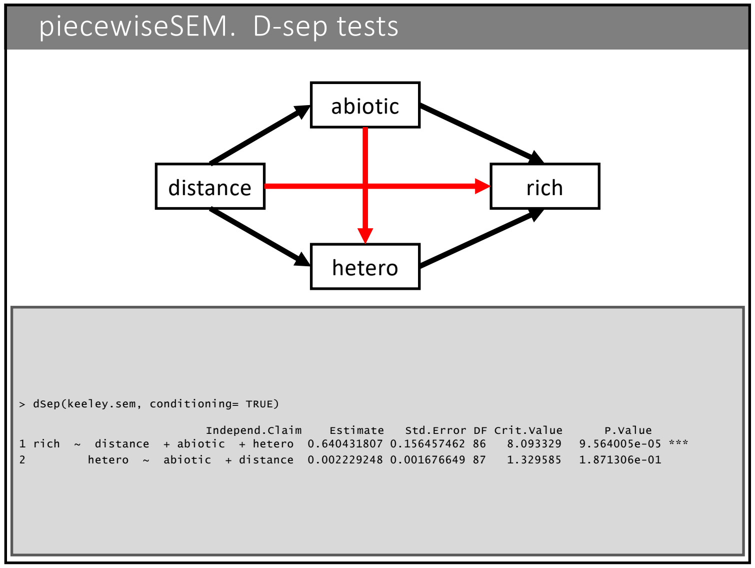


```
# Compute Fisher's C & compare to Chi-square distribution
C <- -2 * (log(coefs(claim1)[1, 7]) + log(coefs(claim2)[1, 7]))

1 - pchisq(C, 2 * 2)

[1] 0.0002223955
```

piecewiseSEM. D-sep tests

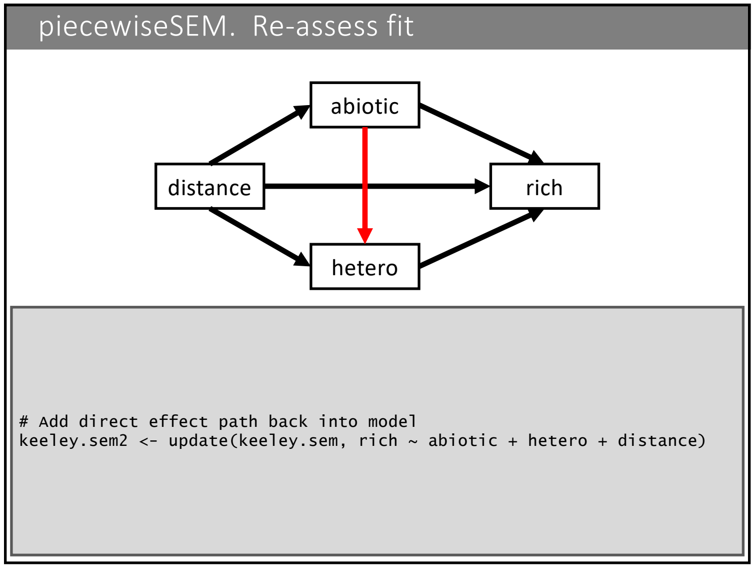


```
> dSep(keeley.sem)

           Independ.Claim    Estimate   Std.Error DF Crit.Value      P.Value
1  rich  ~  distance + ... 0.640431807 0.156457462 86   8.093329 9.564005e-05 ***
2 hetero  ~  abiotic + ... 0.002229248 0.001676649 87   1.329585 1.871306e-01
```



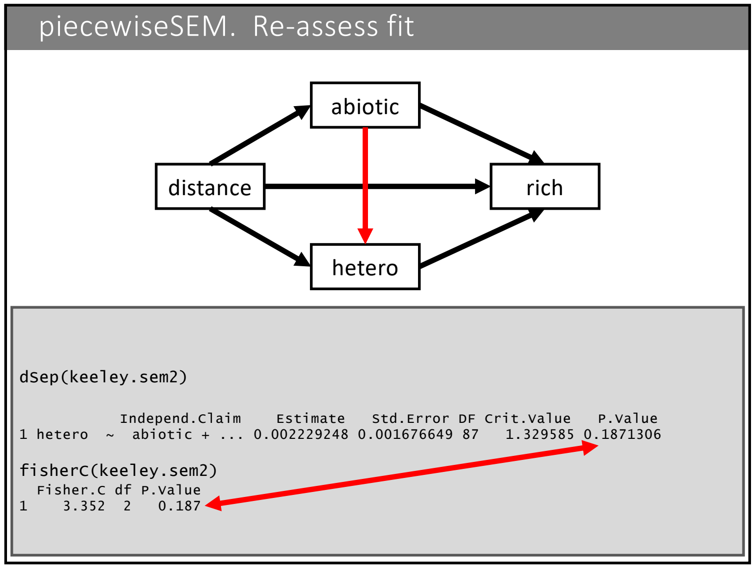
piecewiseSEM. D-sep tests

```
> dSep(keeley.sem, conditioning= TRUE)

                              Independ.Claim    Estimate   Std.Error DF Crit.Value      P.Value
1 rich  ~  distance  + abiotic  + hetero  0.640431807 0.156457462 86   8.093329  9.564005e-05 ***
2          hetero  ~  abiotic  + distance  0.002229248 0.001676649 87   1.329585  1.871306e-01
```



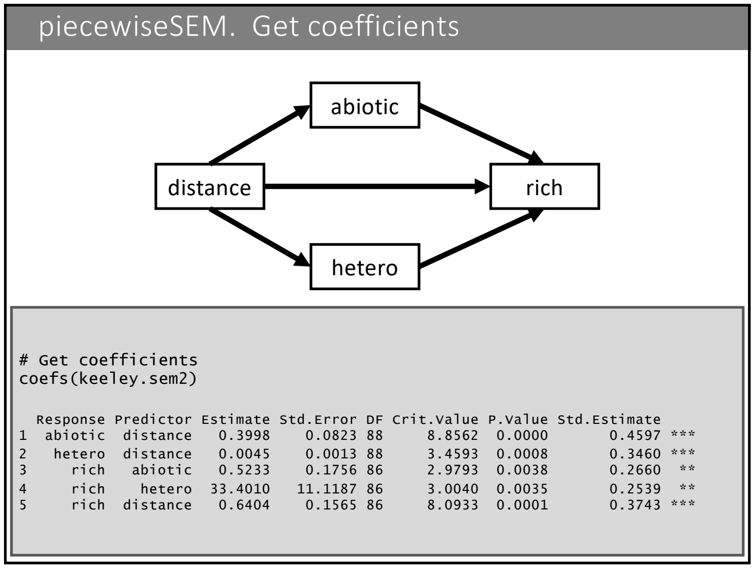
piecewiseSEM. Re-assess fit

```
# Add direct effect path back into model
keeley.sem2 <- update(keeley.sem, rich ~ abiotic + hetero + distance)
```



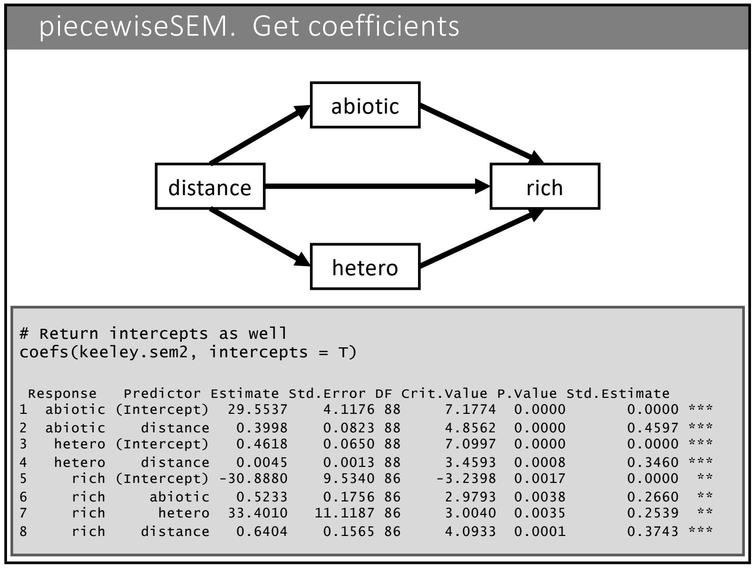
piecewiseSEM. Re-assess fit

```
dSep(keeley.sem2)

          Independ.Claim     Estimate   Std.Error DF Crit.Value   P.Value
1 hetero  ~  abiotic + ... 0.002229248 0.001676649 87    1.329585 0.1871306

fisherC(keeley.sem2)
  Fisher.C df P.Value
1    3.352  2    0.187
```



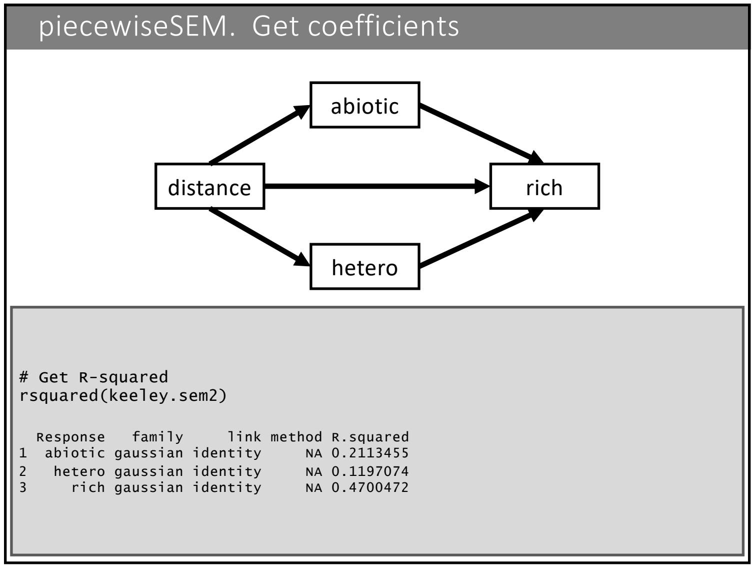
piecewiseSEM. Get coefficients

```
# Get coefficients
coefs(keeley.sem2)

  Response Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate
1  abiotic  distance   0.3998    0.0823 88     8.8562  0.0000       0.4597 ***
2   hetero  distance   0.0045    0.0013 88     3.4593  0.0008       0.3460 ***
3     rich   abiotic   0.5233    0.1756 86     2.9793  0.0038       0.2660  **
4     rich    hetero  33.4010   11.1187 86     3.0040  0.0035       0.2539  **
5     rich  distance   0.6404    0.1565 86     8.0933  0.0001       0.3743 ***
```



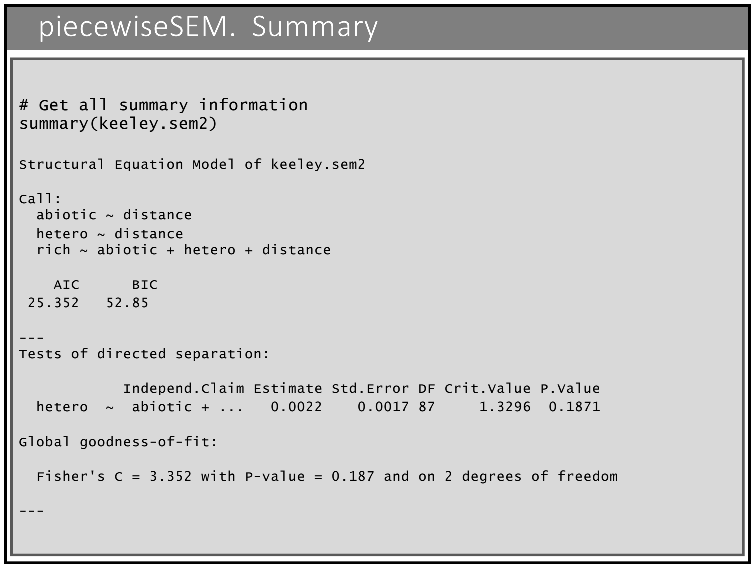
piecewiseSEM. Get coefficients

```
# Return intercepts as well
coefs(keeley.sem2, intercepts = T)

 Response   Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate
1  abiotic (Intercept)  29.5537    4.1176 88     7.1774  0.0000       0.0000 ***
2  abiotic    distance   0.3998    0.0823 88     4.8562  0.0000       0.4597 ***
3   hetero (Intercept)   0.4618    0.0650 88     7.0997  0.0000       0.0000 ***
4   hetero    distance   0.0045    0.0013 88     3.4593  0.0008       0.3460 ***
5     rich (Intercept) -30.8880    9.5340 86    -3.2398  0.0017       0.0000  **
6     rich     abiotic   0.5233    0.1756 86     2.9793  0.0038       0.2660  **
7     rich      hetero  33.4010   11.1187 86     3.0040  0.0035       0.2539  **
8     rich    distance   0.6404    0.1565 86     4.0933  0.0001       0.3743 ***
```



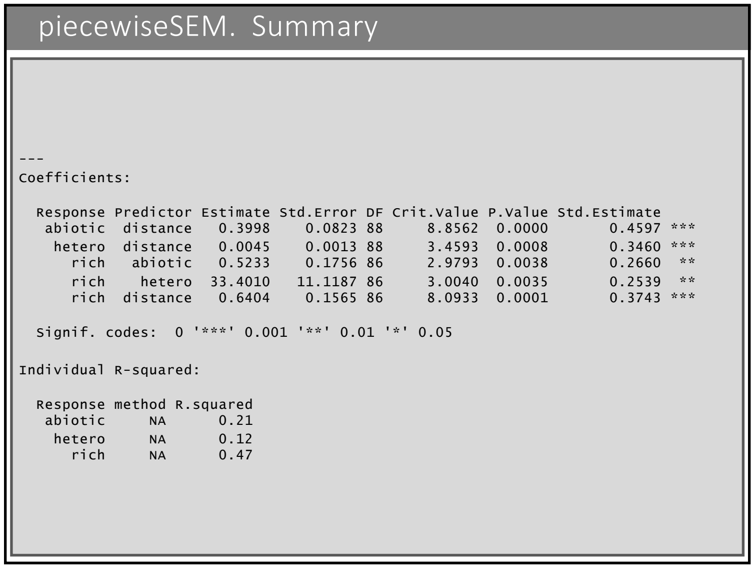
piecewiseSEM. Get coefficients

```
# Get R-squared
rsquared(keeley.sem2)

  Response   family     link method R.squared
1  abiotic gaussian identity     NA 0.2113455
2   hetero gaussian identity     NA 0.1197074
3     rich gaussian identity     NA 0.4700472
```



piecewiseSEM. Summary

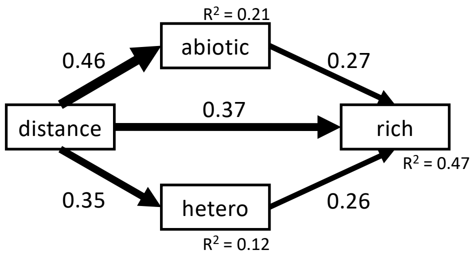
```
# Get all summary information
summary(keeley.sem2)

Structural Equation Model of keeley.sem2

Call:
  abiotic ~ distance
  hetero ~ distance
  rich ~ abiotic + hetero + distance

    AIC      BIC
 25.352   52.85

---
Tests of directed separation:

            Independ.Claim Estimate Std.Error DF Crit.Value P.Value
  hetero  ~  abiotic + ...   0.0022    0.0017 87     1.3296  0.1871

Global goodness-of-fit:

  Fisher's C = 3.352 with P-value = 0.187 and on 2 degrees of freedom

---
```



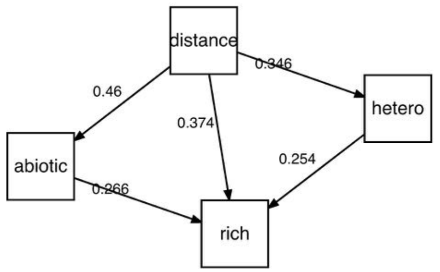
piecewiseSEM. Summary

```
---
Coefficients:

  Response Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate
   abiotic  distance   0.3998    0.0823 88     8.8562  0.0000       0.4597 ***
    hetero  distance   0.0045    0.0013 88     3.4593  0.0008       0.3460 ***
      rich   abiotic   0.5233    0.1756 86     2.9793  0.0038       0.2660  **
      rich    hetero  33.4010   11.1187 86     3.0040  0.0035       0.2539  **
      rich  distance   0.6404    0.1565 86     8.0933  0.0001       0.3743 ***

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

Individual R-squared:

  Response method R.squared
   abiotic     NA      0.21
    hetero     NA      0.12
      rich     NA      0.47
```

piecewiseSEM. Summary

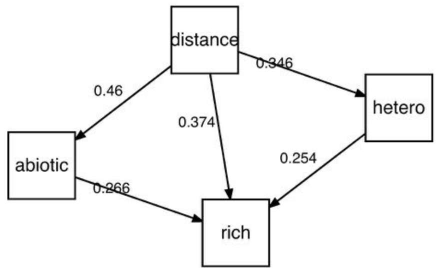


piecewiseSEM. plot



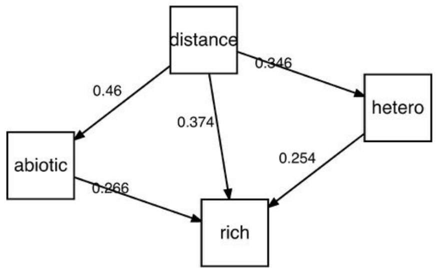
```
#plot the model
plot(keeley.sem2)
```

piecewiseSEM. Making a better plot

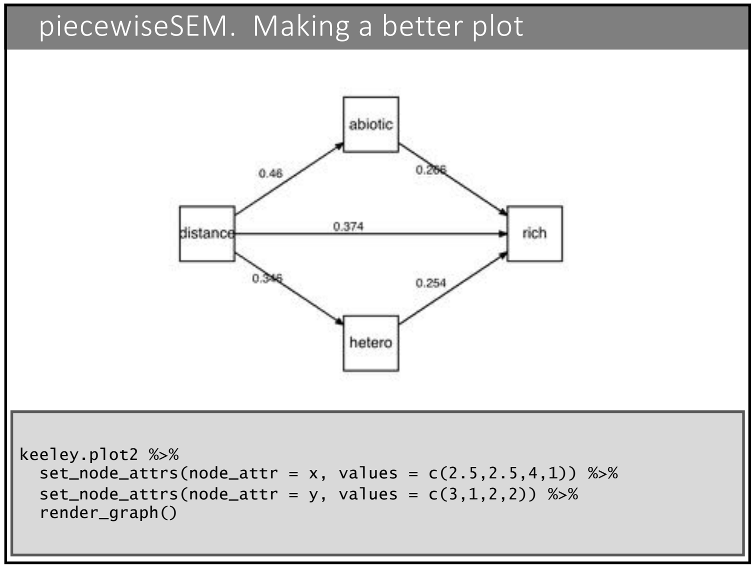


```
#make a better plot
keeley.plot2 <- plot(keeley.sem2, return=TRUE)

#what's in there?
get_node_df(keeley.plot2)
```

piecewiseSEM. Making a better plot

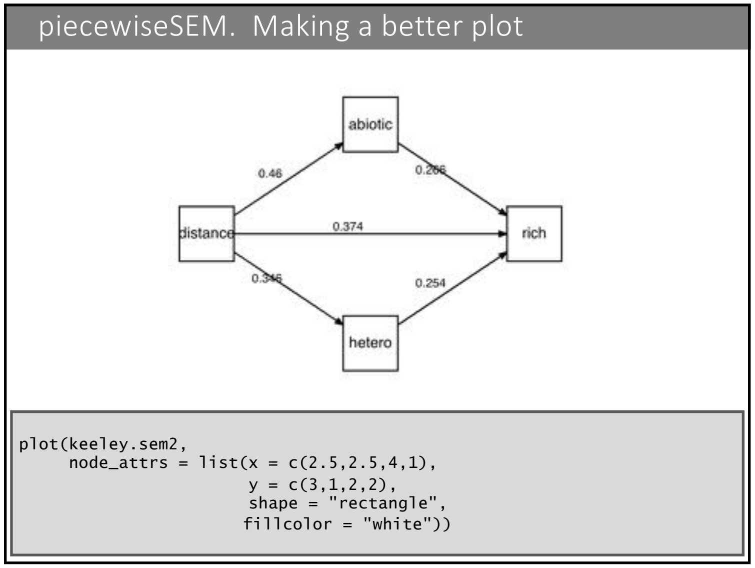


```
> get_node_df(keeley.plot2)
  id  type    label    nodes     shape color fillcolor
1  1 lower  abiotic  abiotic rectangle black     white
2  2 lower   hetero   hetero rectangle black     white
3  3 lower     rich     rich rectangle black     white
4  4 lower distance distance rectangle black     white
```



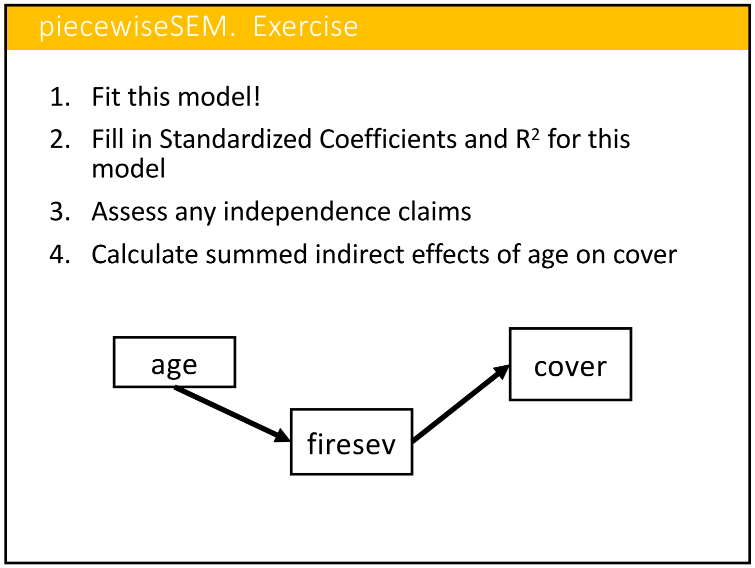
piecewiseSEM. Making a better plot

```
keeley.plot2 %>%
  set_node_attrs(node_attr = x, values = c(2.5,2.5,4,1)) %>%
  set_node_attrs(node_attr = y, values = c(3,1,2,2)) %>%
  render_graph()
```



piecewiseSEM. Making a better plot

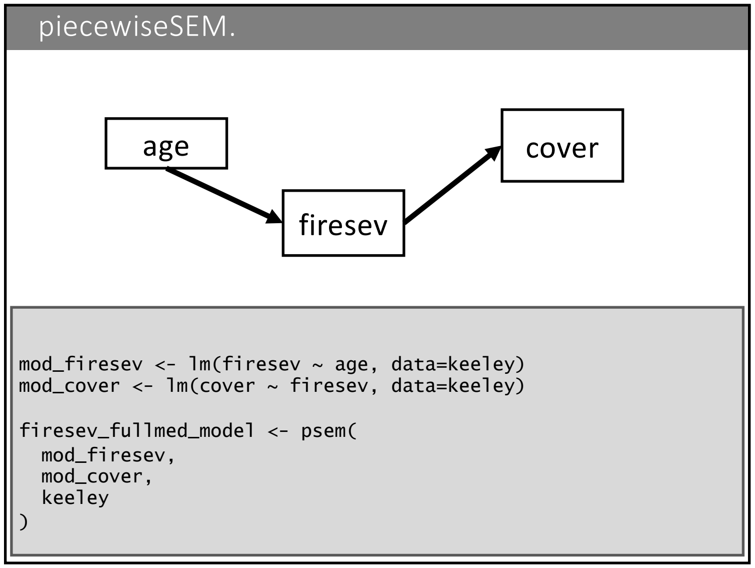
```
plot(keeley.sem2,
     node_attrs = list(x = c(2.5,2.5,4,1),
                       y = c(3,1,2,2),
                      shape = "rectangle",
                     fillcolor = "white"))
```



piecewiseSEM. Exercise

1. Fit this model!
2. Fill in Standardized Coefficients and R^2 for this model
3. Assess any independence claims
4. Calculate summed indirect effects of age on cover

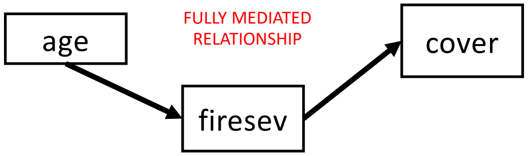
age
firesev
cover



piecewiseSEM.

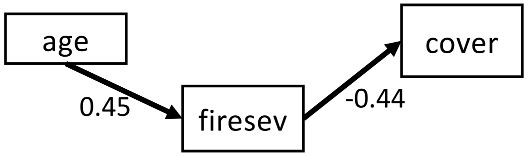
```
mod_firesev <- lm(firesev ~ age, data=keeley)
mod_cover <- lm(cover ~ firesev, data=keeley)

firesev_fullmed_model <- psem(
  mod_firesev,
  mod_cover,
  keeley
)
```

piecewiseSEM.



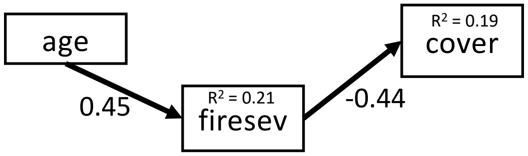
```
> dSep(firesev_fullmed_model)

       Independ.Claim      Estimate   Std.Error DF Crit.Value    P.Value
1 cover  ~  age + ... -0.004832969 0.002682241 87   -1.80184 0.07503437
```

piecewiseSEM.



```
> coefs(firesev_fullmed_model)

   Response Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate
1  firesev       age   0.0597    0.0125 88     4.7781       0       0.4539 ***
2    cover   firesev  -0.0839    0.0184 88    -4.5594       0      -0.4371 ***
```

piecewiseSEM.

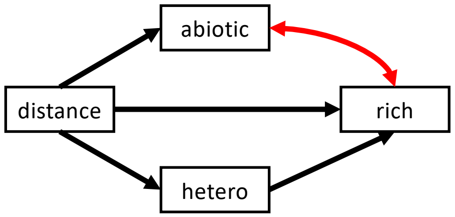


```
> rsquared(firesev_fullmed_model)

  Response   family     link method R.squared
1  firesev gaussian identity     NA 0.2059938
2    cover gaussian identity     NA 0.1910867
```

Overview

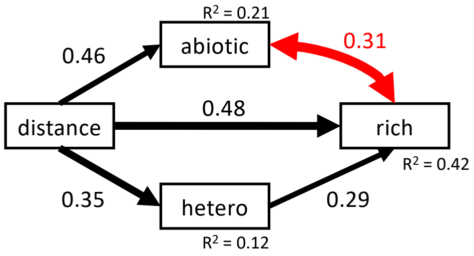
1. Covariance vs. piecewise SEM
2. Assessing Model Fit: Tests of directed separation
3. Introduction to *piecewiseSEM*
4. Further adventures into *piecewiseSEM*

piecewiseSEM. Correlated errors

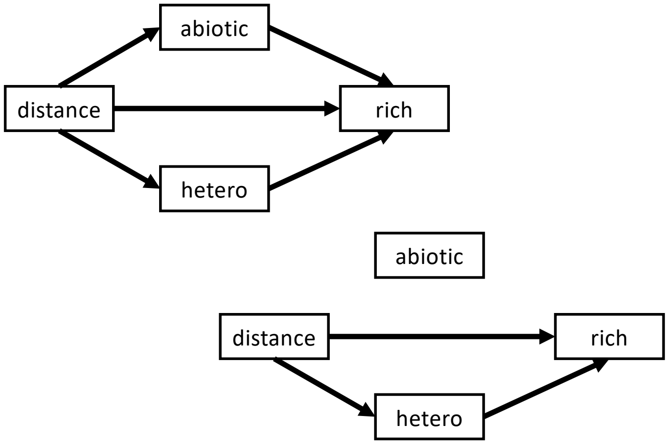


```
keeley.sem3 <- psem(
  lm(abiotic ~ distance, data = keeley),
  lm(hetero ~ distance, data = keeley),
  lm(rich ~ distance + hetero, data = keeley),
  rich %~~% abiotic
)

summary(keeley.sem3)
```

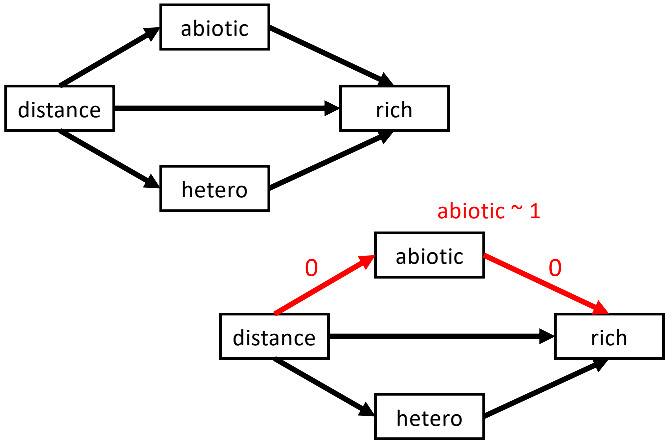
piecewiseSEM. Correlated errors



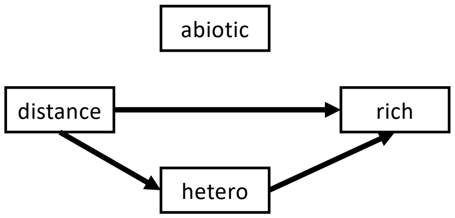
piecewiseSEM. AIC comparisons



piecewiseSEM. Model comparisons

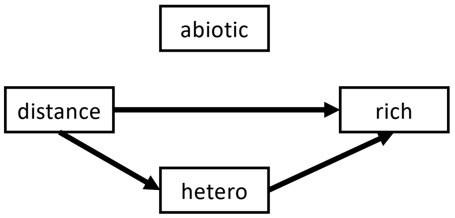


piecewiseSEM. Fit new model



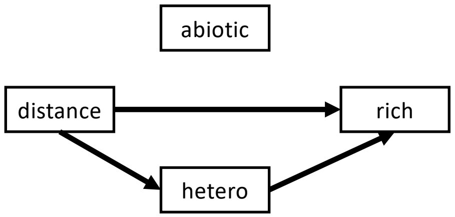
```
keeley.sem4 <- psem(
  lm(hetero ~ distance, data = keeley),
  lm(rich ~ distance + hetero, data = keeley),
  abiotic ~ 1,
)
```

piecewiseSEM. Compare with LRT



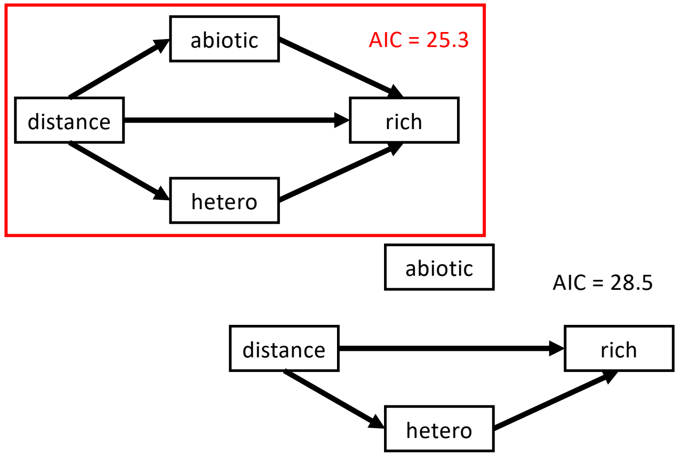
```
> anova(keeley.sem2, keeley.sem4)
    Chi square difference test

        AIC     BIC Fisher.C Fisher.C.Diff DF.diff P.value
1    25.352 52.850    3.352
vs 2 28.522 46.021   14.522         11.17       2  0.0038 **
```

piecewiseSEM. Compare with AIC



```
AIC(keeley.sem2)[1]
[1] 25.352

AIC(keeley.sem4)[1]
[1] 28.522
```

piecewiseSEM. AIC comparisons



piecewiseSEM. Understanding Relationships

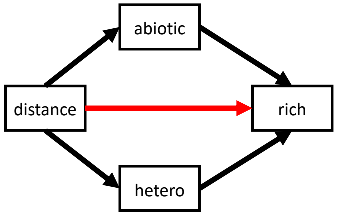
3 Ways to View the independent effect of distance on richness:

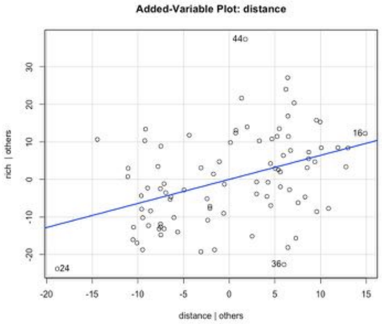
1. Partial Regression Plots
   - residual of richness + distance effect ~ residual of distance after accounting for abiotic + hetero
   - VERY VERY conservative, but, accurate
2. Component Residual Plots
   - residual of richness + distance effect ~ distance
3. Fit of richness ~ distance holding other variables at median
   - Preferred, as in sensible units

piecewiseSEM. Digging into psem

```
#psem objects are lists of models
> keeley.sem2[[3]]

Call:
lm(formula = rich ~ abiotic + hetero +
distance, data = keeley)

Coefficients:
(Intercept)      abiotic       hetero
distance
   -30.8880       0.5233      33.4010
0.6404
```

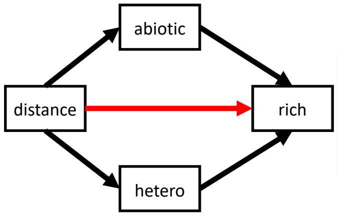
piecewiseSEM. Partial correlations

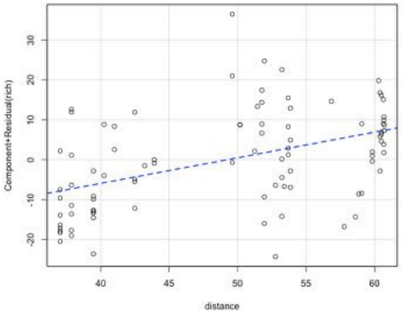




```
#Added Variable Plots
library(car)

avPlot(keeley.sem2[[3]], variable = "distance")
```

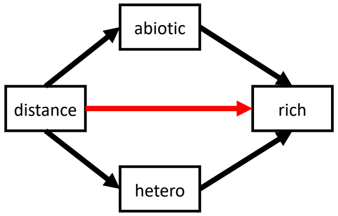
piecewiseSEM. Component + Residual

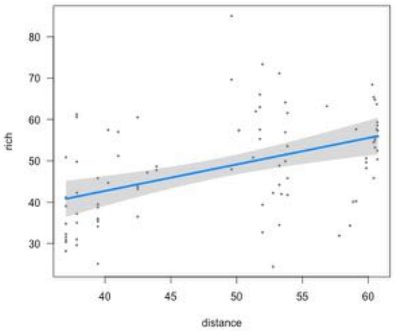




```
#CR Plot

crPlot(keeley.sem2[[3]], variable = "distance")
```

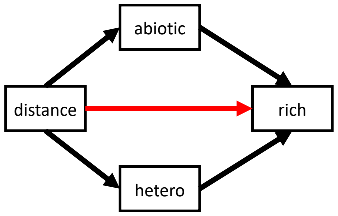
piecewiseSEM. Distance Effect When All Others are At Median





```
#visreg
library(visreg)

visreg(keeley.sem2[[3]],
       xvar = "distance")
```

piecewiseSEM. Working with Controlled Data



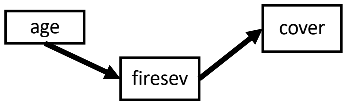
```
> vr <- visreg(keeley.sem2[[3]],
               xvar = "distance")

> head(vr$res)
   abiotic   hetero distance rich visregRes visregPos
1 48.03605 0.684314 53.40900   50  41.95826     FALSE
2 48.03605 0.684314 37.03745   50  41.15731      TRUE
3 48.03605 0.684314 53.69565   50  64.10536      TRUE
4 48.03605 0.684314 53.69565   50  56.91605      TRUE
5 48.03605 0.684314 51.95985   50  73.34811      TRUE
6 48.03605 0.684314 51.95985   50  39.34695     FALSE
```

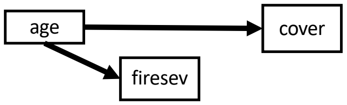
Visualizing While Controlling Covariates

- Useful for displaying trends, particularly with complex models where bivariate correlations are messy
- Can be used for any multiple regression (single model or list)
- Not applicable to simple regression (*Y* ~ *X*)

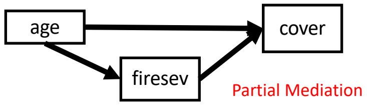
piecewiseSEM. Exercise

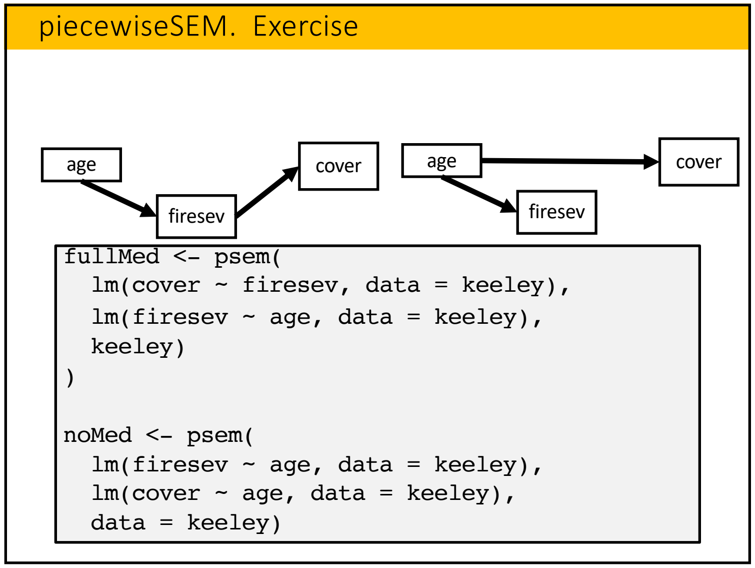
1. Fit and compare these models!





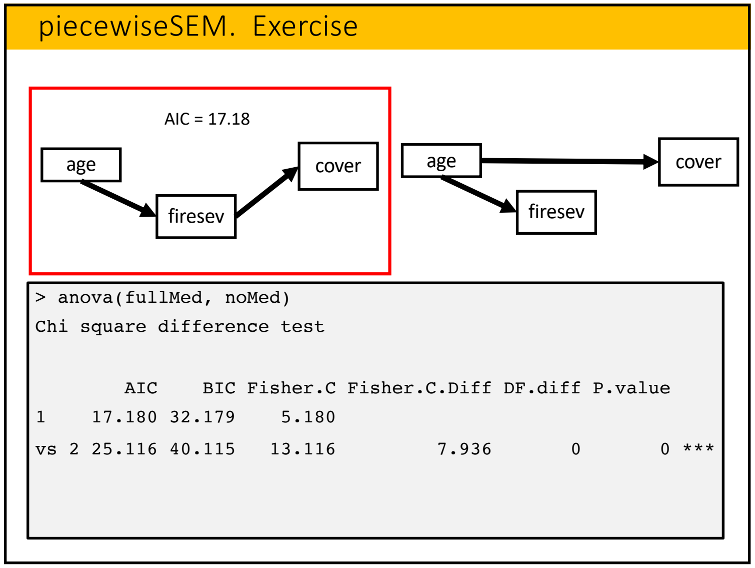
2. Visualize the effect of fire severity's effect on cover given this model structure





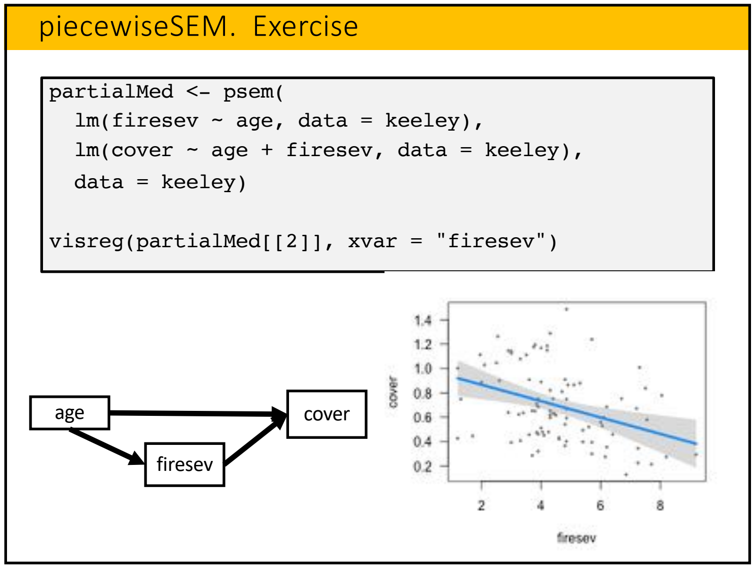
piecewiseSEM. Exercise

```
fullMed <- psem(
  lm(cover ~ firesev, data = keeley),
  lm(firesev ~ age, data = keeley),
  keeley)
)

noMed <- psem(
  lm(firesev ~ age, data = keeley),
  lm(cover ~ age, data = keeley),
  data = keeley)
```



piecewiseSEM. Exercise

```
> anova(fullMed, noMed)
Chi square difference test

        AIC     BIC Fisher.C Fisher.C.Diff DF.diff P.value
1    17.180  32.179    5.180
vs 2 25.116  40.115   13.116         7.936       0       0 ***
```



piecewiseSEM. Exercise

```
partialMed <- psem(
  lm(firesev ~ age, data = keeley),
  lm(cover ~ age + firesev, data = keeley),
  data = keeley)

visreg(partialMed[[2]], xvar = "firesev")
```