Introduction to Local Estimation

Overview

1. Traditional vs. piecewise SEM
2. Tests of directed separation
3. Log-likelihood assessment
4. Introduction to *piecewiseSEM*
5. A Warning

1.1 Traditional vs. Piecewise SEM

1.1 Comparison. Traditional vs. piecewise SEM

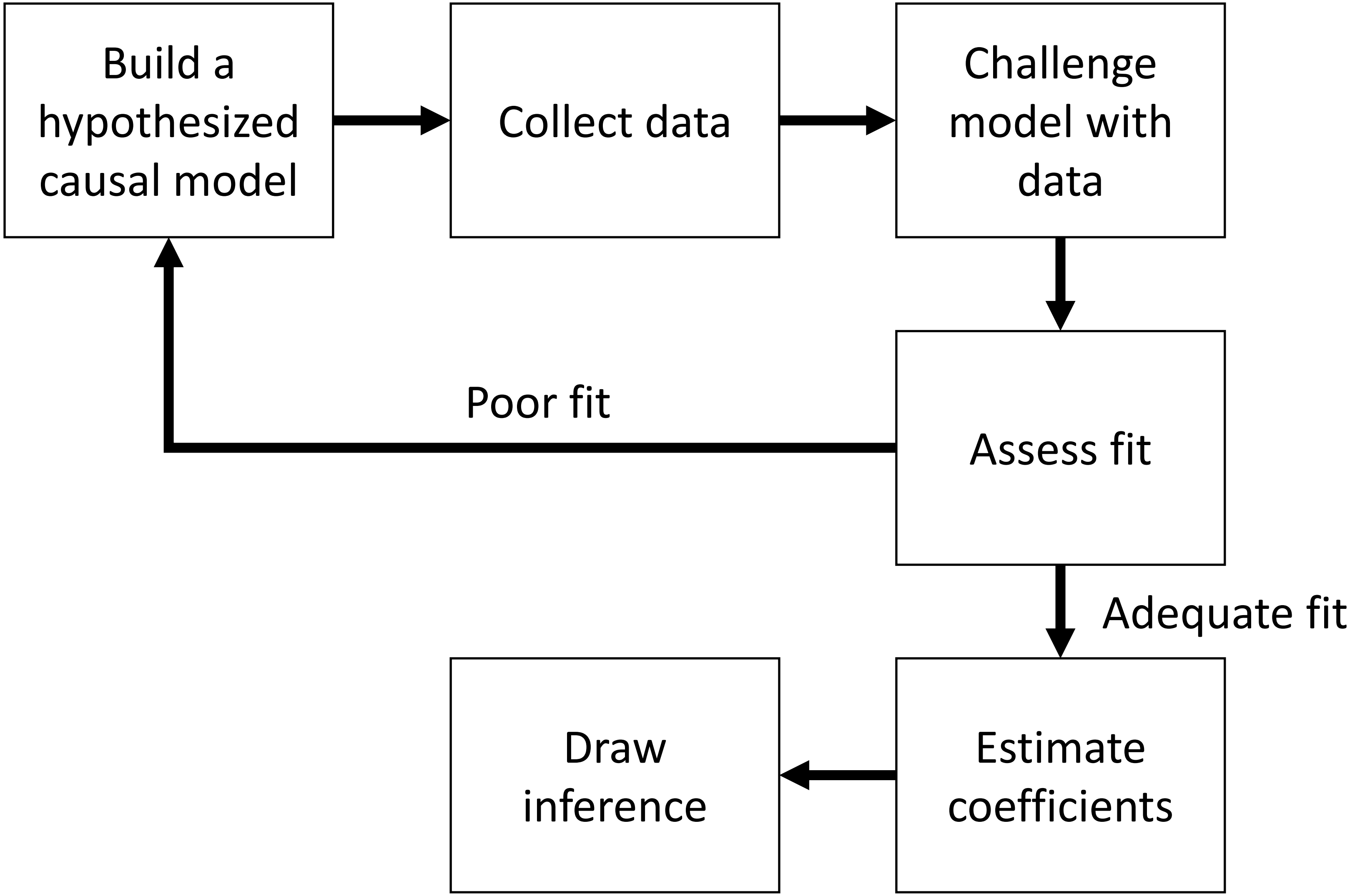
Variance-covariance	Piecewise
Single (global) variance-covariance matrix estimated	Variance-covariance matrices estimated for each endogenous variable
Simultaneous solution (computationally intensive)	Multiple solutions (modularized)
Fit to normal distribution	Incorporates various distributions (Poisson,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

1.1 Comparison. Traditional vs. piecewise SEM

Traditional SEM

Piecewise SEM

1.1 Piecewise SEM.



Build a hypothesized causal model

Collect data

Challenge model with data

Assess fit

Poor fit

Adequate fit

Estimate coefficients

Draw inference

1.2 Tests of Directed Separation

1.2 Directed Separation. Model fit

Does the model fit the data?

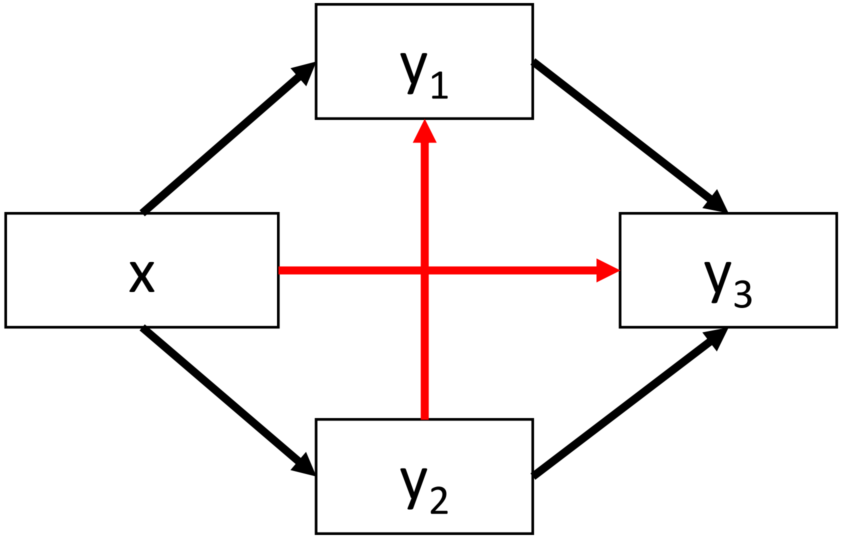
=

Does the model represent the data well?

=

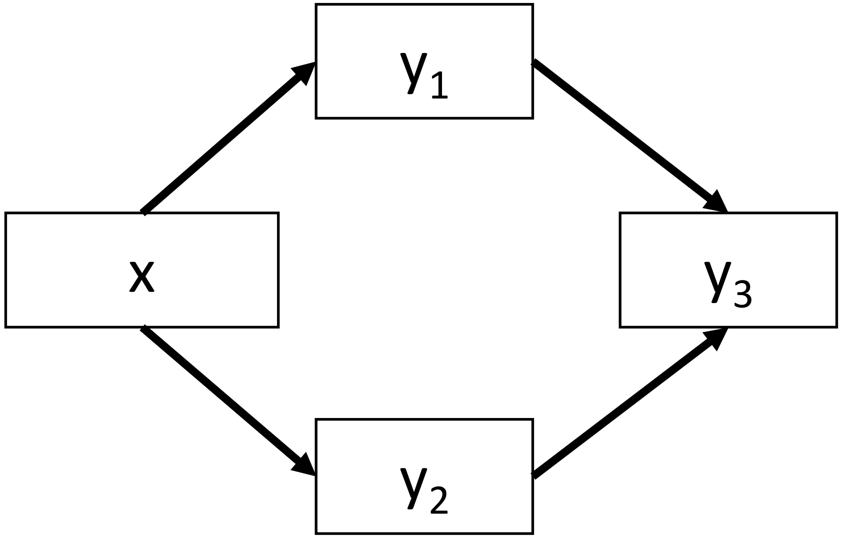
Are we missing important information?

1.2 Directed Separation. Model fit



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

1.2 Directed Separation. D-separation



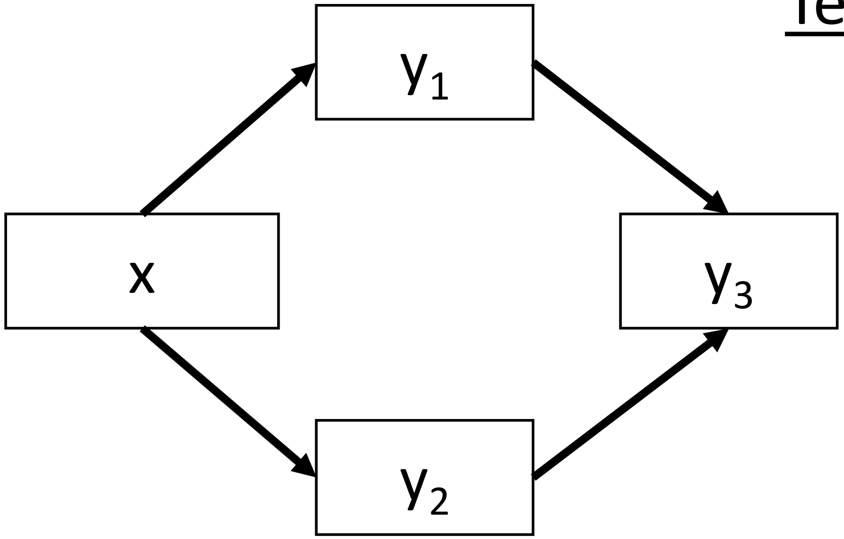
- Concept from *Graph Theory*
- Two nodes are d-separated if they are *conditionally independent* e.g., the effect of *x* on y_3 is zero conditioning on the influences of y_1 and y_2

1.2 Directed Separation. Independence claims

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

1. *Directly* controlling for causal connections via *immediate* parents
2. *Indirectly* controlling for common ancestors that could generate correlations between the pair (as reflected in immediate parents)
3. Not controlling for common descendent variables (the effect of the cause)

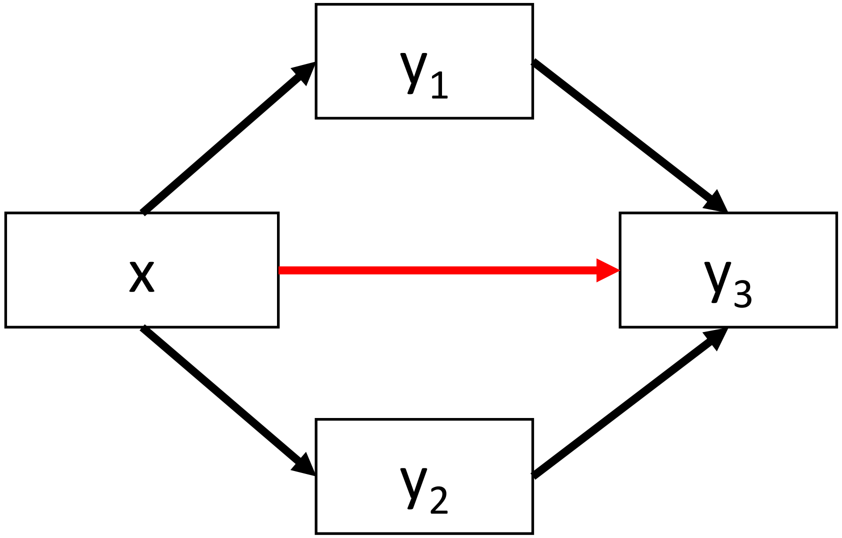
1.2 Directed Separation. D-separation



Test of directed separation:

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

1.2 Directed Separation. Independence claims



1. **Identify all independence claims**

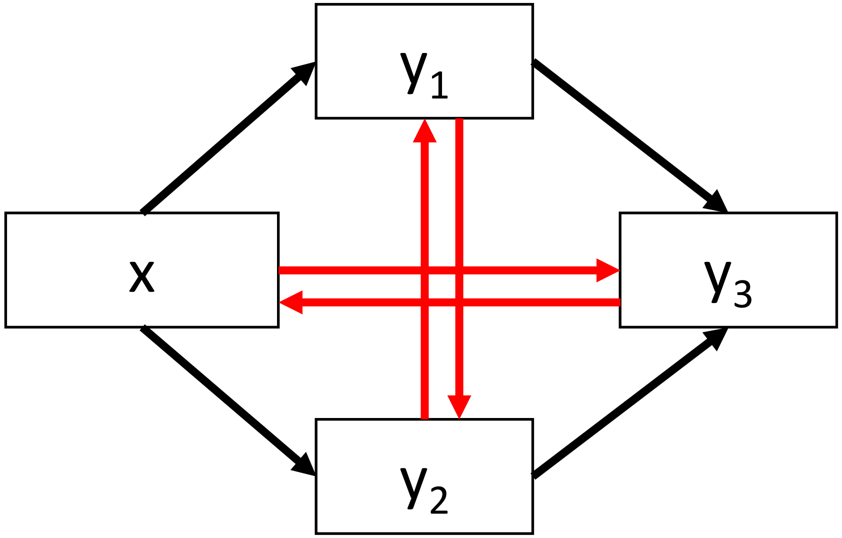
Basis set = the *smallest* possible set of independence claims from a graph

Notation:

$x \perp y_3 \mid (y_1, y_2)$

Predictor involved in claim Response Conditioning variables

1.2 Directed Separation. Independence claims



1. **Identify all independence claims**

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

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

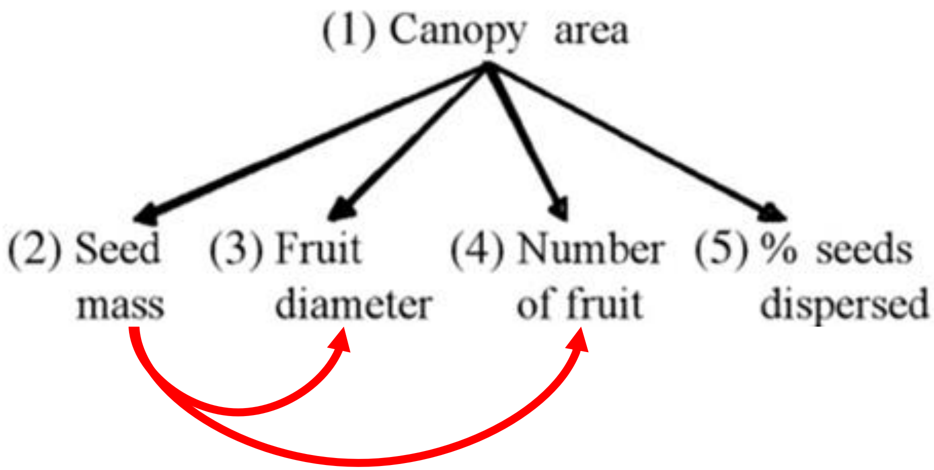
3. $y_1 \perp y_2 \mid (x)$

~~4. $y_2 \perp y_1 \mid (x)$~~

"Minimum set" = reciprocal relationships are not part of the basis set (*P*-values are identical in either direction, EXCEPT…)

1.2 Directed Separation. Deriving the basis set

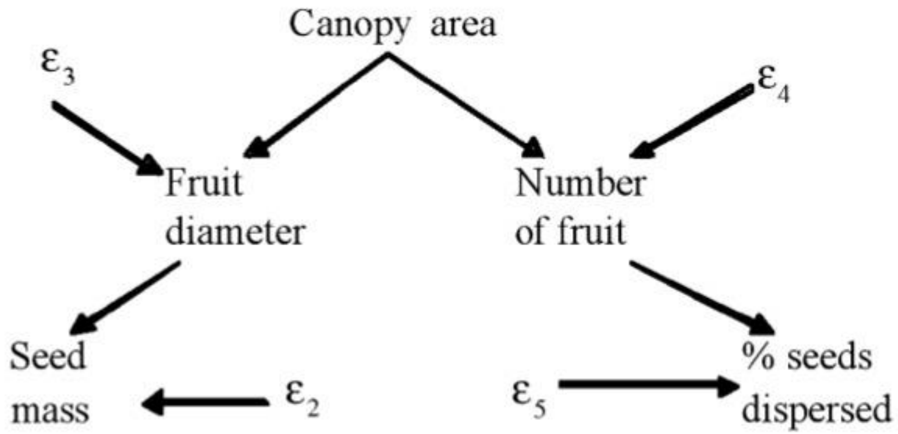
What is the basis set?



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

1.2 Directed Separation. Deriving the basis set

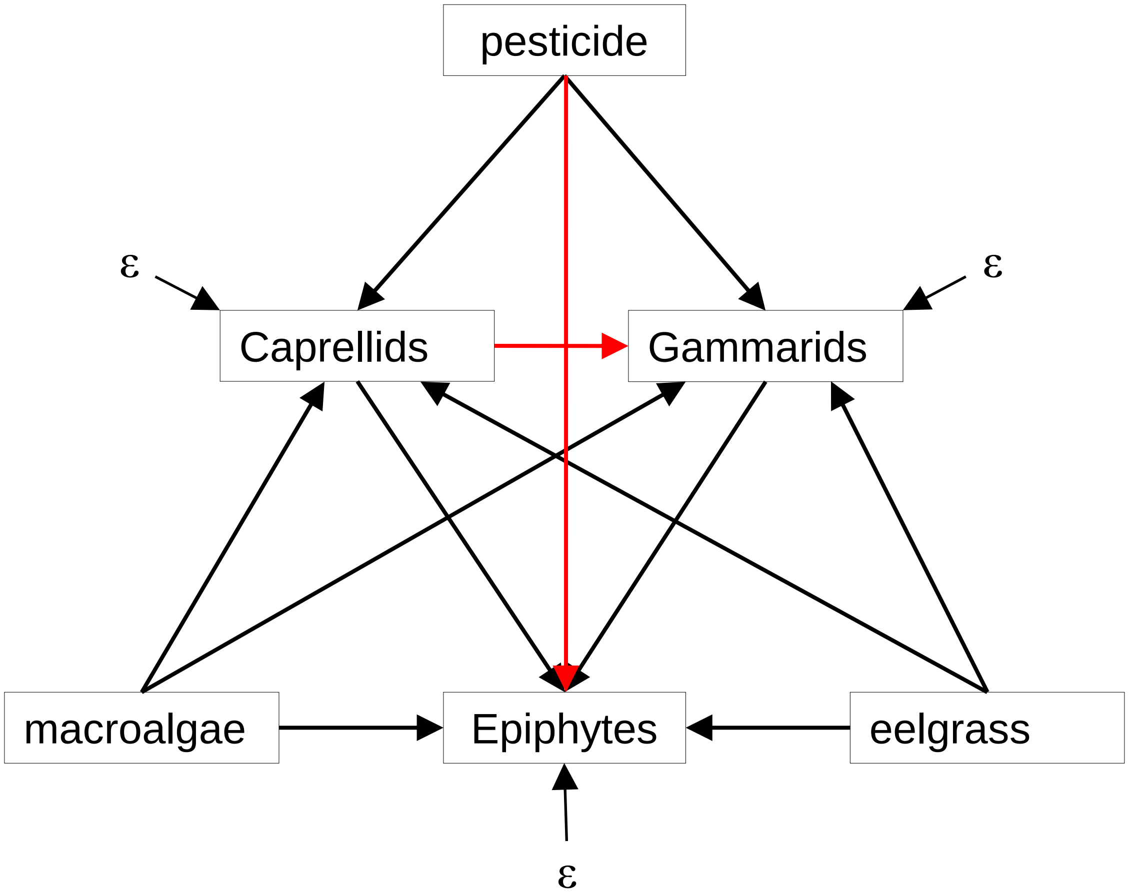
What is the basis set?



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

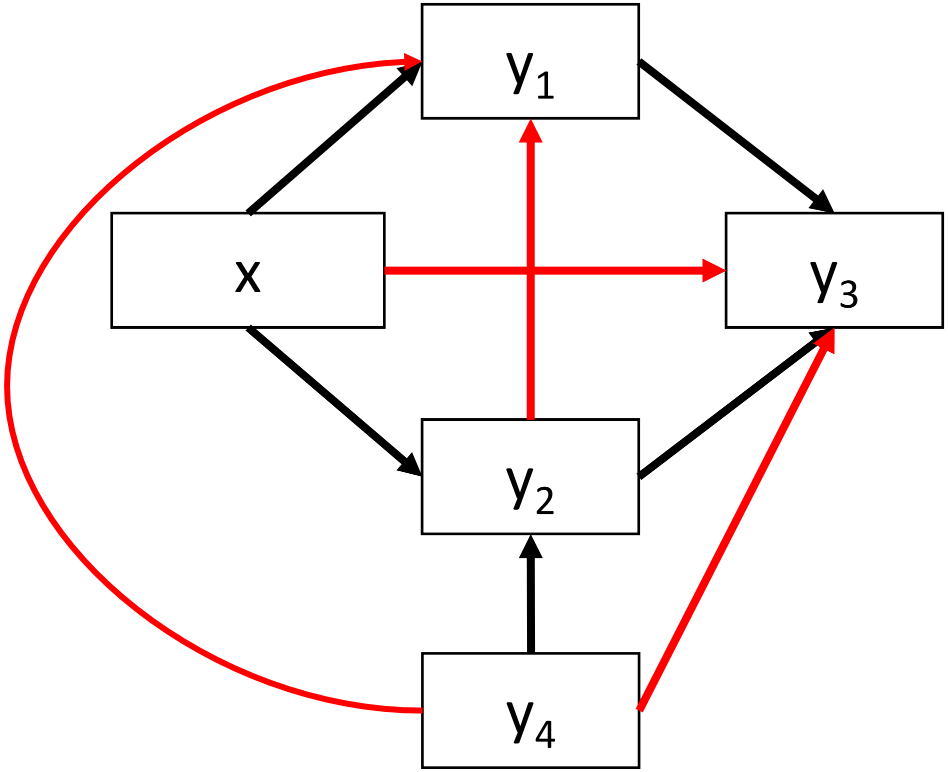
1.2 Directed Separation. Deriving the basis set

What is the basis set?



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

1.2 Directed Separation. Independence claims



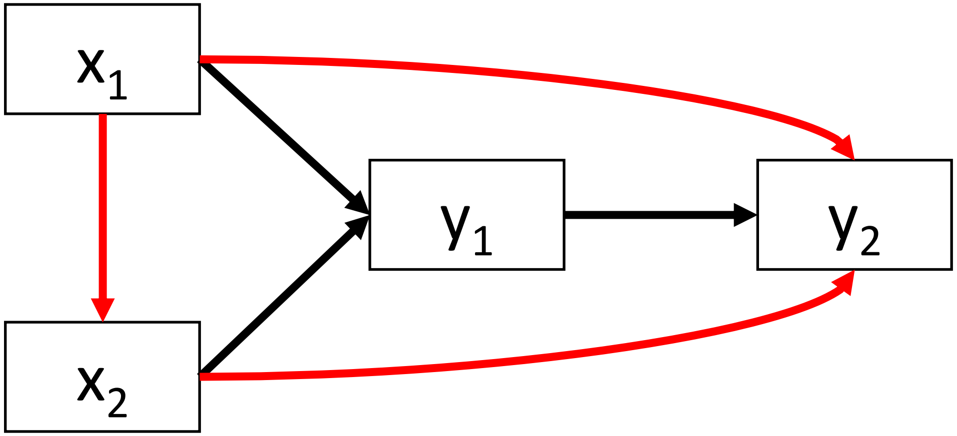
1. **Identify all independence claims**

   1. $x \perp y_3 \mid (y_1, y_2)$
   2. $y_1 \perp y_2 \mid (x)$
   3. $y_4 \perp y_1 \mid (x)$
   4. $y_4 \perp y_3 \mid (y_1, y_2)$
   5. $y_4 \perp x$???

Not conditioning on y4, since its >1 node away

1.2 Directed Separation. A note

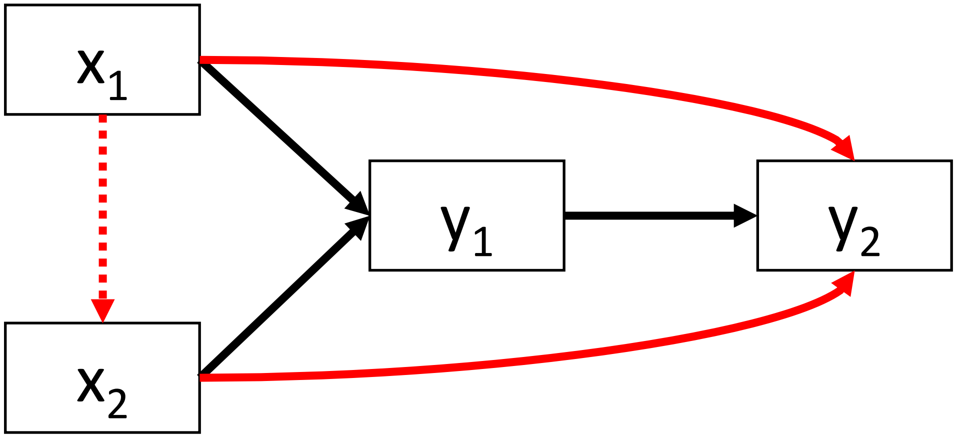
- Basis set excludes relationships among exogenous variables



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

1.2 Directed Separation. A note

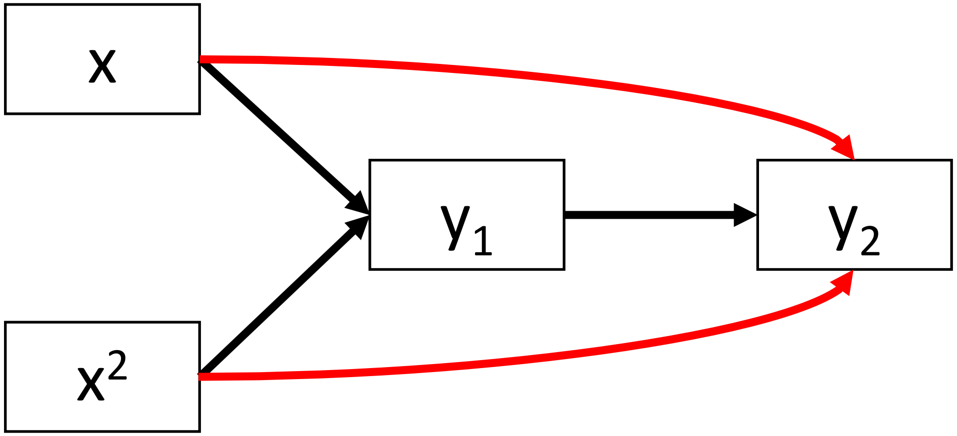
- Unclear as to the direction of the relationship (x_1 -> x_2 or x_2 -> x_1)
- Unclear whether variables could even be plausibly causally linked (e.g., ocean basin and latitude)
- Distributional assumptions, etc. not defined



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

1.2 Directed Separation. A note

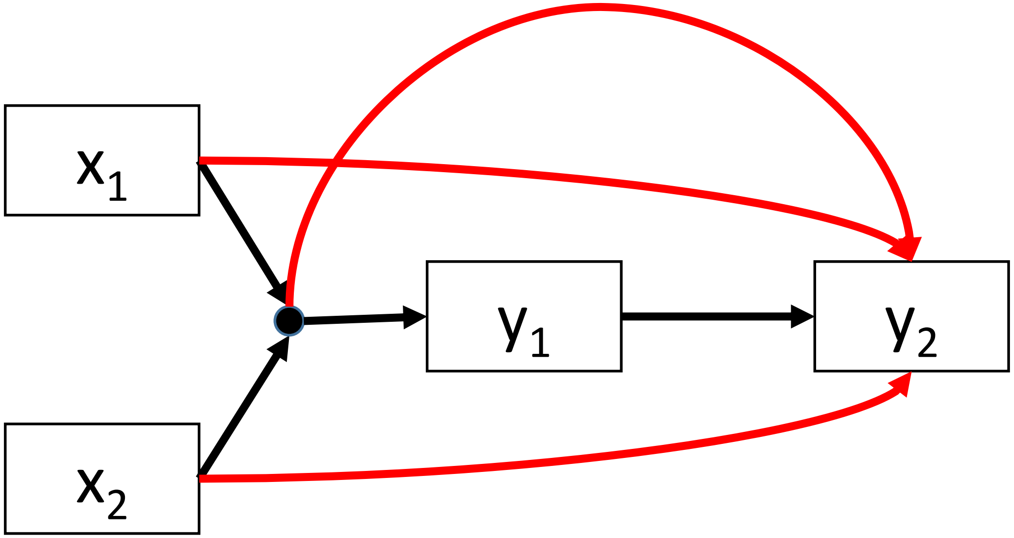
- Basis set generally excludes non-linear components (polynomials)



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

1.2 Directed Separation. A note

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



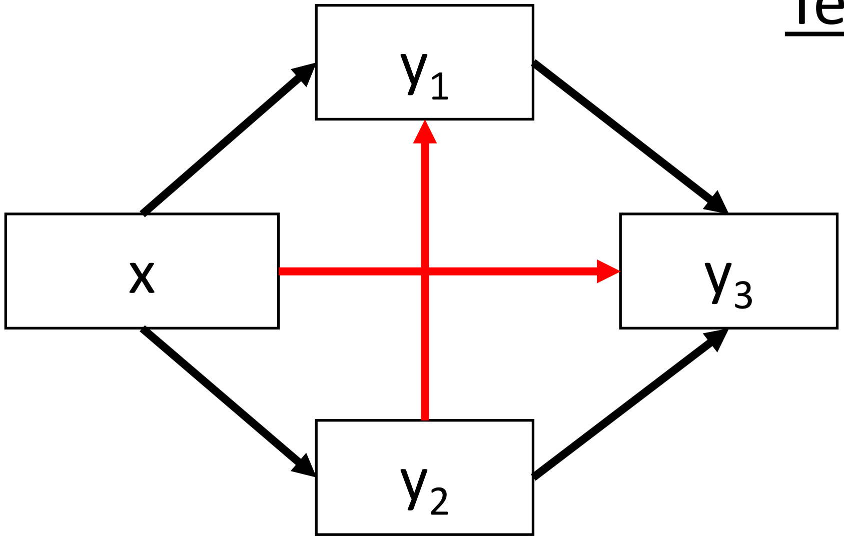
1. $x_1 \perp y_2 \mid (y_1)$
2. $x_2 \perp y_2 \mid (y_1)$

ACTIVITY

- Take your causal diagram from Day 1

- Derive the basis set

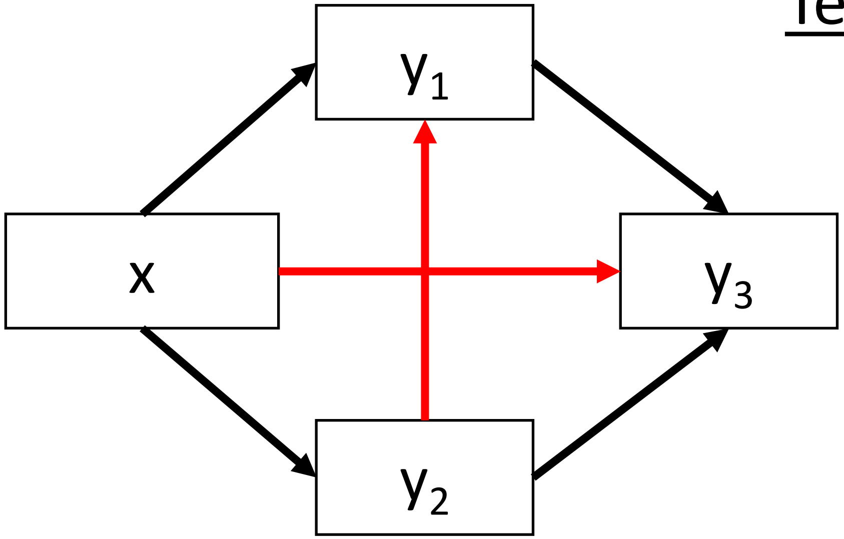
1.2 Directed Separation. D-separation



Test of directed separation:

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

1.2 Directed Separation. D-separation



Test of directed separation:

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

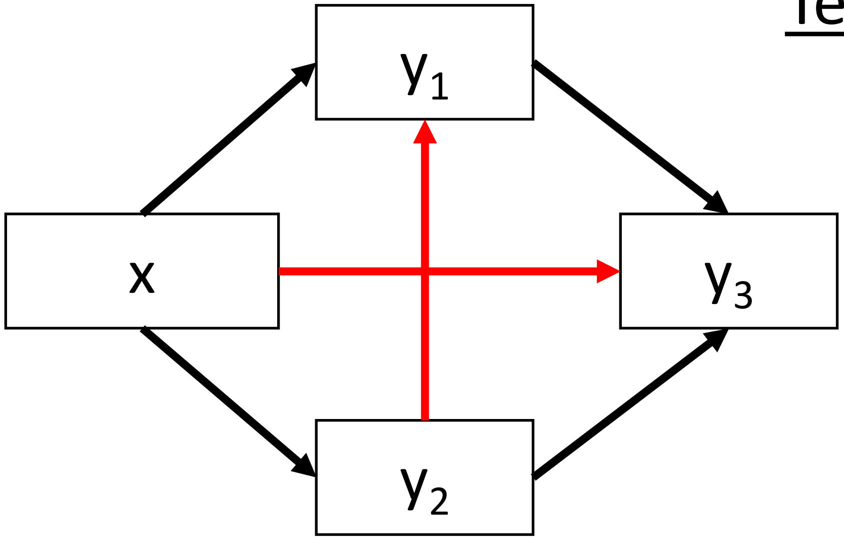
$y_3 \sim y_1 + y_2 + x$

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

$y_2 \sim x + y_1$

- Fit models (using same parameters as originally specified) and extract null significance statistic: *P*-value)
- A non-significant *P*-value suggests the claim is conditionally independent (i.e., relationship is no different than 0)

1.2 Directed Separation. D-separation



<u>Test of directed separation:</u>

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

1.2 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

1.2 Directed Separation. Fisher's *C*

<u>What is $p < 0.05$?</u>

- 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, <u>**$p \geq 0.05$ is GOOD**</u>

1.2 Directed Separation. Model selection

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

$$\mathrm{AIC} = C + 2K$$

- K = # of likelihood parameters estimated (not to be confused with k)

- Can be extended to small sample size:

$$\mathrm{AIC}c = C + 2K(n / (n - K - 1))$$

1.2 Directed Separation. Complexity and sample size

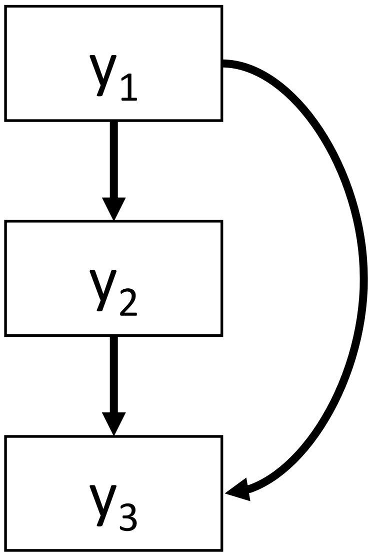
- Shipley suggests need only enough individual degrees of freedom to fit each component model

- Or, *d*-rule (Grace et al 2015):
  - *d* = # of samples / # of pathways
  - *d* ≥ 5

- 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*

1.2 Directed Separation. Saturated models

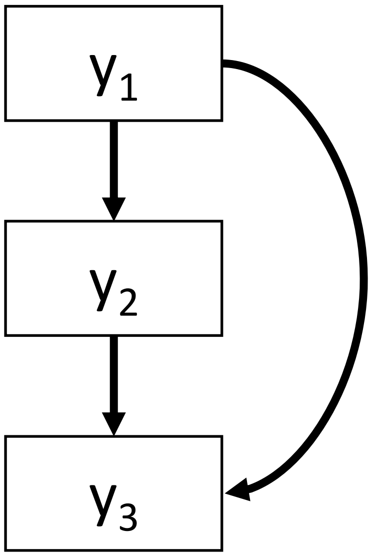
What is the basis set?



- There is no basis set for a saturated model (all paths are represented)
- No d-sep tests and therefore no *C* statistic can be constructed for this model (same as global estimation)

1.2 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., path significance, R^2)

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

1.3 Log-likelihood assessment

1.3. Model fit

Does the model fit the data?

=

Does the model represent the data well?

=

Are there more *likely* configurations?

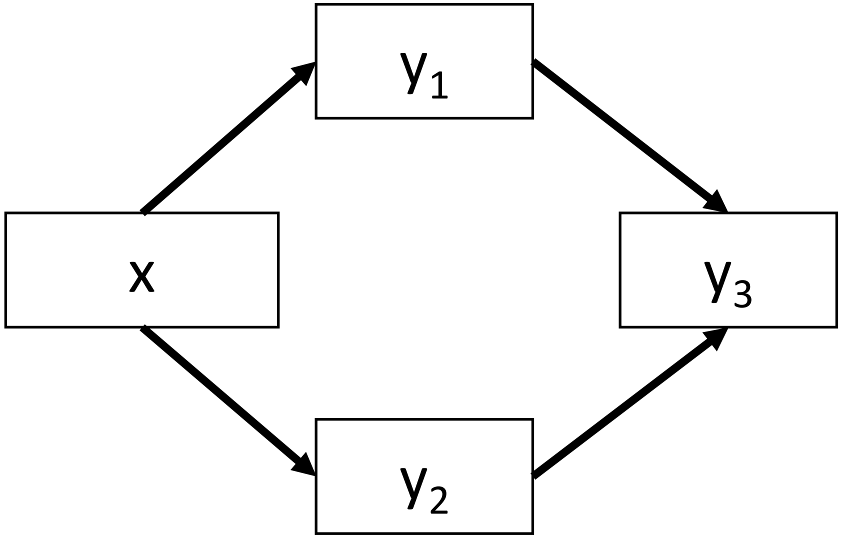
1.3. Directed separation. The trouble with *P*-values

- *P*-values are not always returned by default (see: *lme4*)
- D-sep tests only reflect changes in *topology* (whether paths or variables are missing), but there are lots of components to tweak (distributions, transformations, link functions, etc.) that don’t affect the topology

1.3. Log-likelihood

- *Recall*: Maximum likelihood estimation = find the parameters (coefficients) that maximize the probability of observing the data
- Likelihood = value of the maximum likelihood fitting function with the optimal parameters
- Log-likelihood (L-L) = the log-transformation of the likelihood

1.3. Log-likelihood



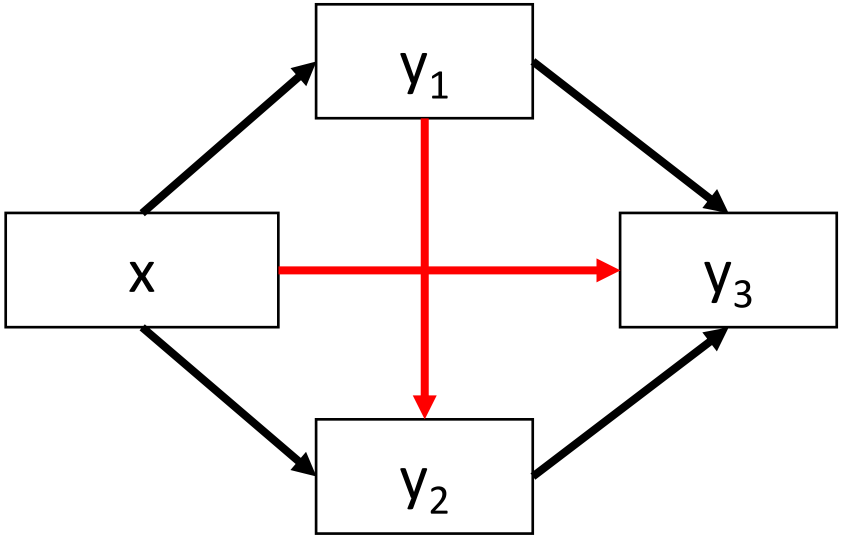
Can compute L-L for each model in our SEM:

$y_1 \sim \gamma_1 x$

$y_2 \sim \gamma_2 x$

$y_3 \sim \beta_1 y_1 + \beta_2 y_1$

1.3. Log-likelihood

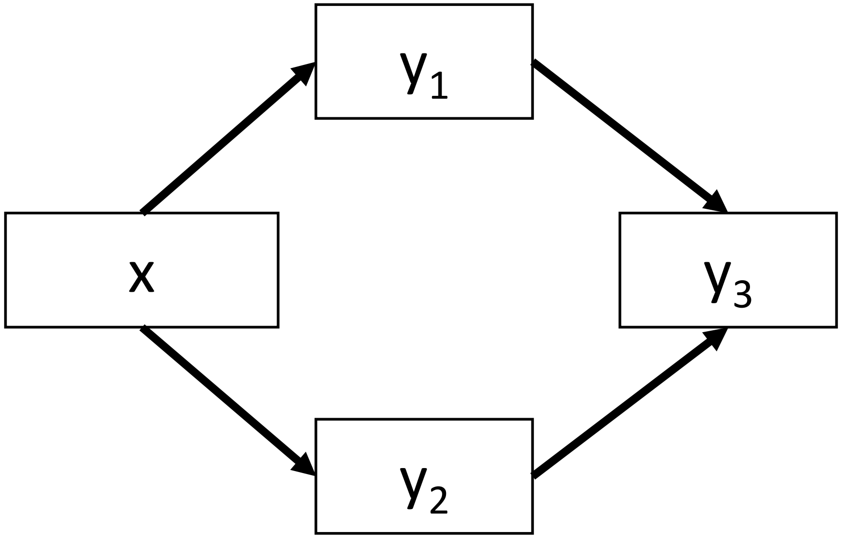


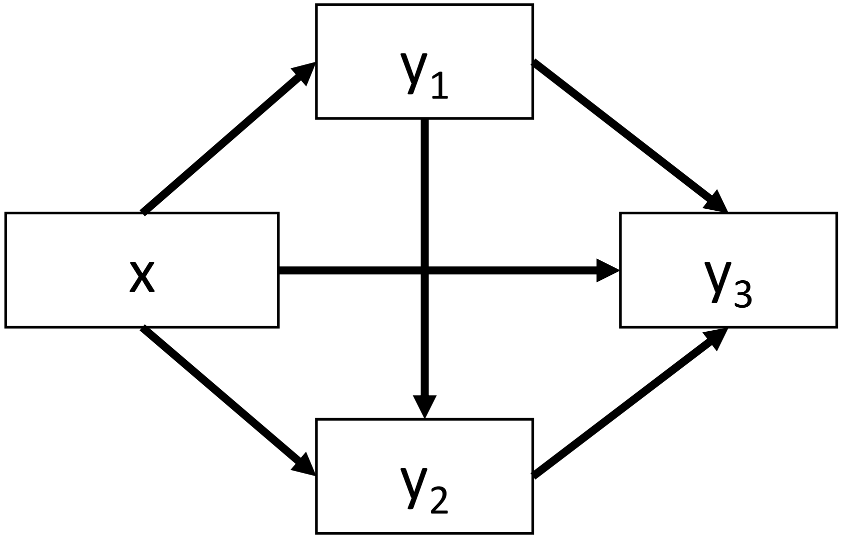
- Recall the goal of d-sep tests are to evaluate missing paths = same as asking if the paths are *no different than 0*

- What is the alternate hypothesis? That these paths *are* different from zero

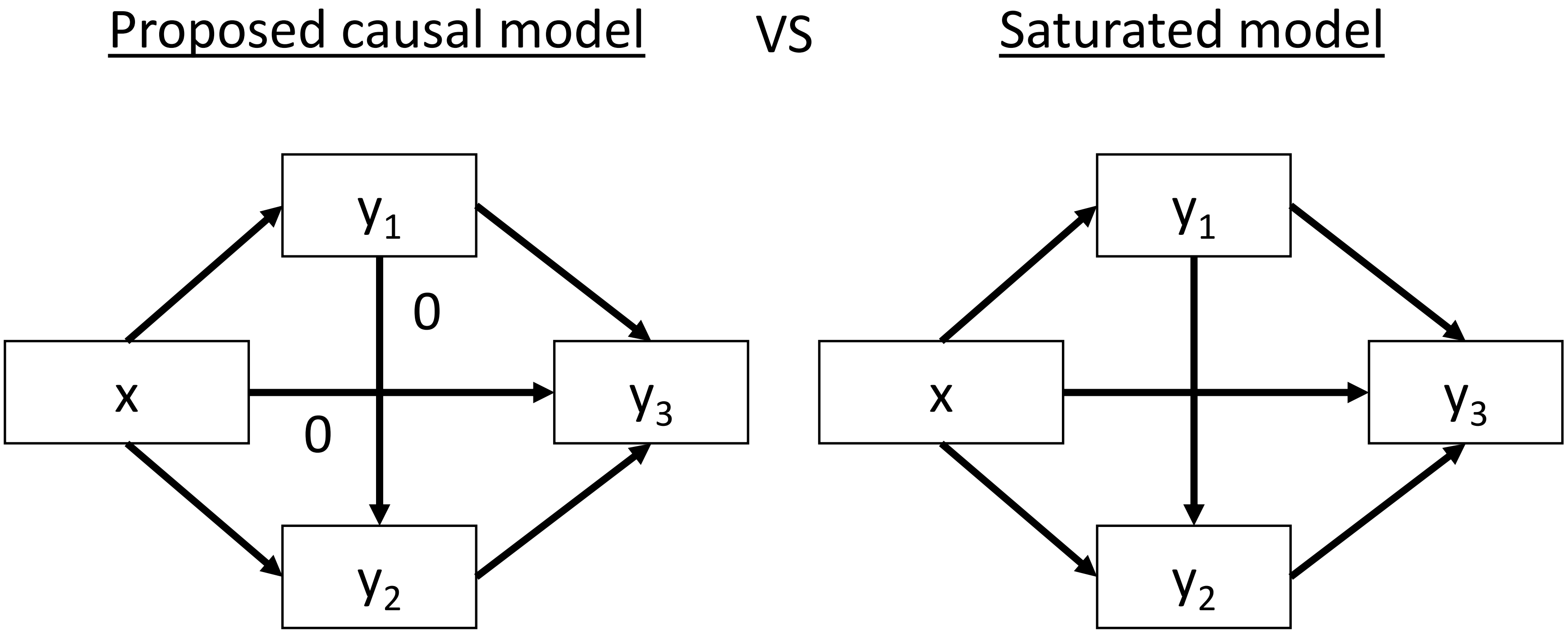
1.3. Log-likelihood. Competing models

Proposed causal model VS Saturated model



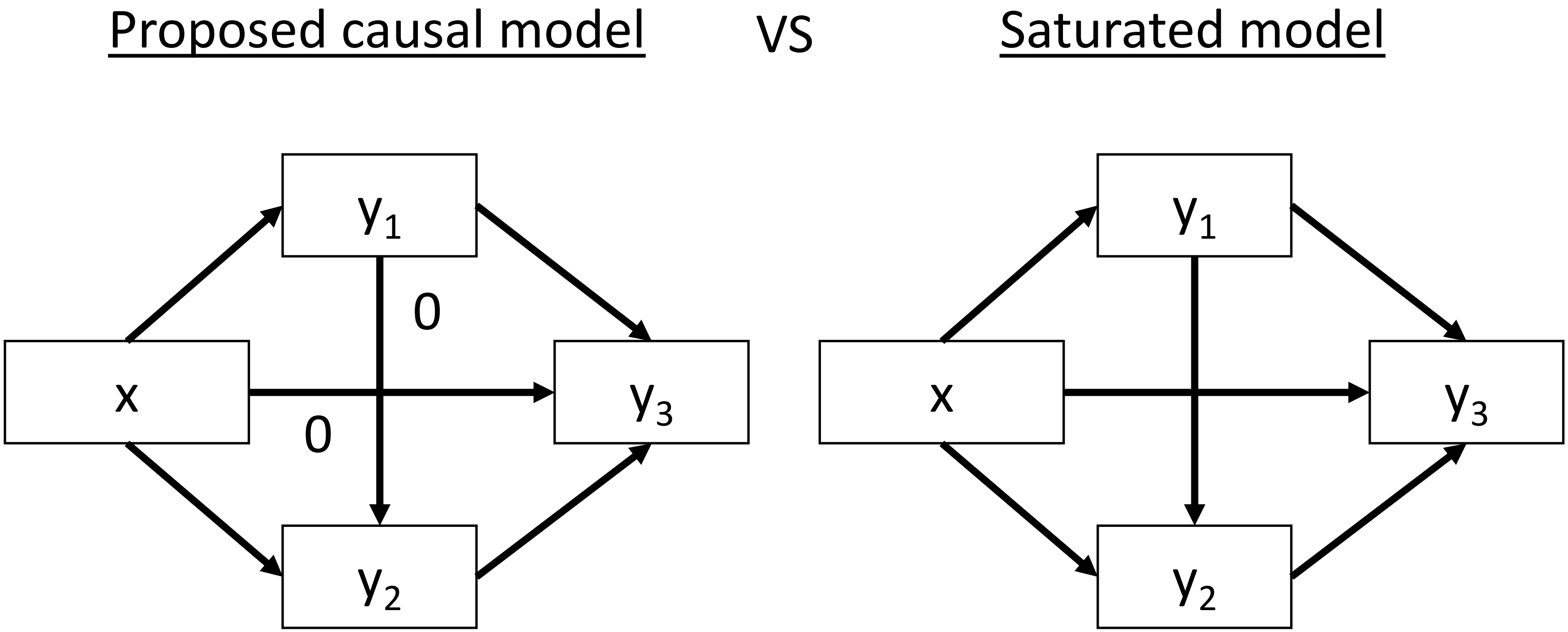


1.3. Log-likelihood. Competing models



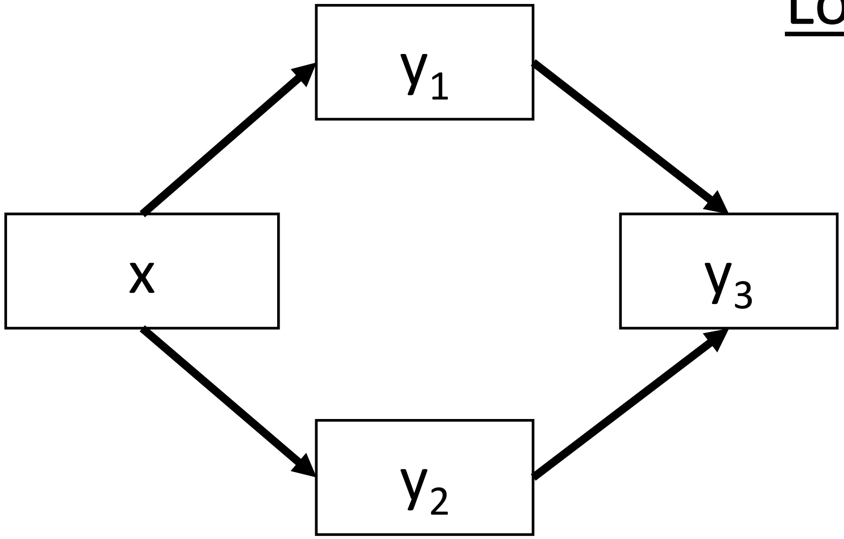
(this is also what we test in global estimation… is the difference in the estimated and observed covariances zero?)

1.3. Log-likelihood. Competing models



*How much more **likely** is the model in which all paths are free to vary (saturated model) then the proposed model?*

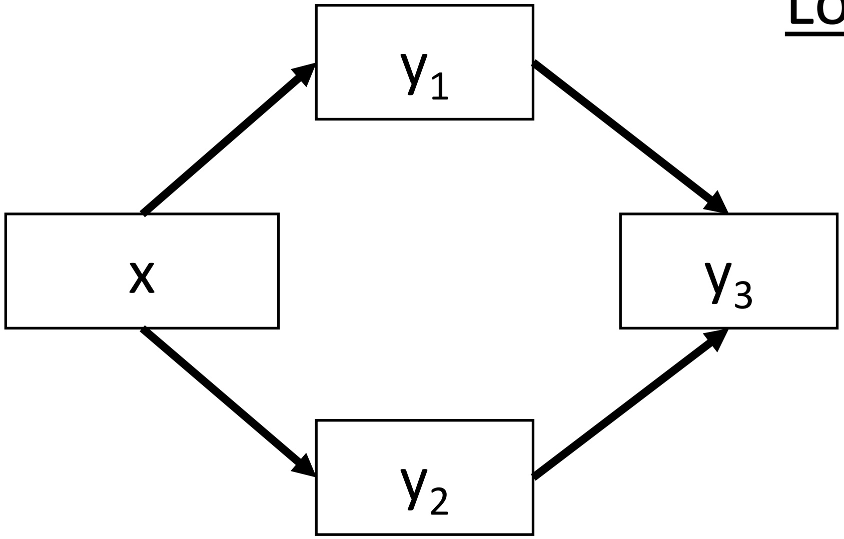
1.3. Log-likelihood. LRT



Log-likelihood test:

1. Fit proposed model and summarize log-likelihoods
2. Fit saturated model and summarize log-likelihoods
3. Test whether they are different

1.3. Log-likelihood. LRT



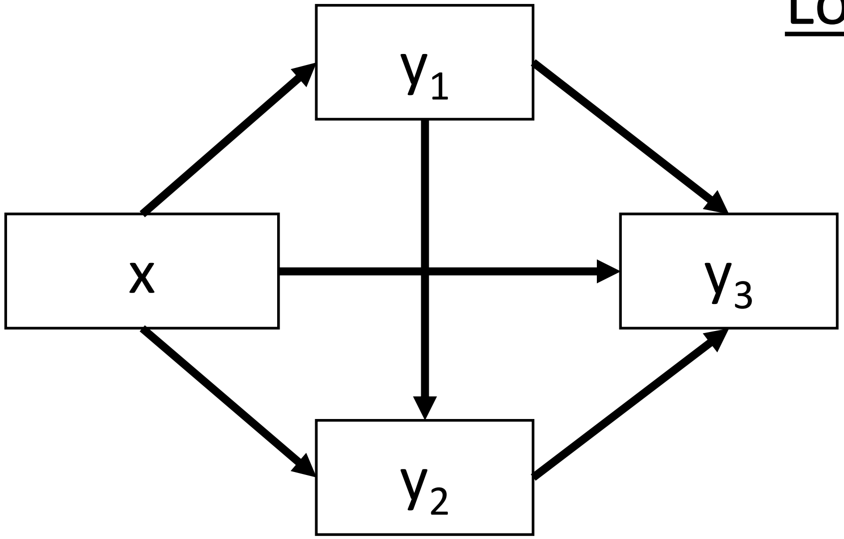
<u>Log-likelihood test:</u>

1. **Fit proposed model and summarize log-likelihoods**

Log-likelihood of SEM is the sum of the individual log-likelihoods

$$\log \mathcal{L}_M(\theta_M|X) = \sum_{i=1}^{v} \log(\mathcal{L}_i(\theta_i|X))$$

1.3. Log-likelihood. LRT

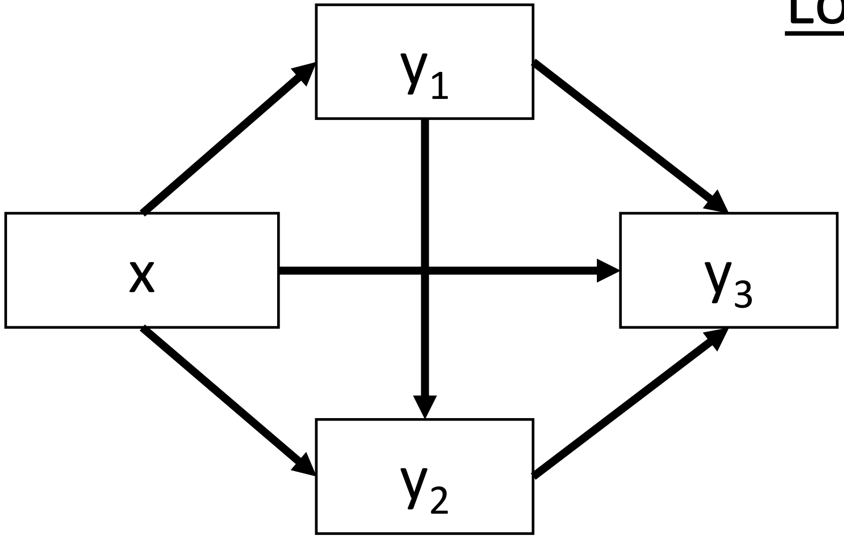


$$\log \mathcal{L}_M(\theta_M|X) = \sum_{i=1}^{v} \log(\mathcal{L}_i(\theta_i|X))$$

Log-likelihood test:

1. Fit proposed model and summarize log-likelihoods
2. **Fit saturated model and summarize log-likelihoods**
3. Test whether they are different

1.3. Log-likelihood. LRT



Log-likelihood test:

1. Fit proposed model and summarize log-likelihoods
2. Fit saturated model and summarize log-likelihoods
3. **Test whether they are different** = χ^2 likelihood ratio test

$$\chi^2 = -2(\log(\mathcal{L}(M_1)) - \log(\mathcal{L}(M_2)))$$

1.3. Log-likelihood. Goodness-of-fit

- χ^2 statistic is the *same as we get from global estimation* if we assume multivariate normality

- LRT requires that proposed model be nested within the saturated model

- Allows extensions of techniques from global estimation (e.g., modification indices = how much does χ^2 likelihood change with additional removal of paths?)

- Can be extended to any model that uses ML estimation (e.g., GAMs) that were previously prohibited

- *Cannot* be used with techniques that are not estimated using maximum likelihood (e.g., quasi-likelihood)

1.3. Log-likelihood. Model comparison

- Can extend likelihood summing concept to compute model-wide AIC from submodel AICs:

$$AIC_M = \sum_{i=1}^{v} AIC_i$$

- Can be extended to small sample size correction
- Solves issue with lack of AIC from d-sep tests for saturated models

1.3. Log-likelihood. Issues

- If model does not converge or random effects are close to 0, then can produce wonky log-likelihood estimates

- This can lead to the impossible situation where $\chi^2 < 0$
  - In this case , you will get an NA for χ^2 statistic

- What to do?
  - Re-fit model and tweak optimization parameters to encourage convergence
  - Drop random effects whose variance components are very small from the model
  - Revert to d-sep tests

1.4 Introduction to *piecewiseSEM*

`Local_Estimation.R`

1.4 piecewiseSEM.

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

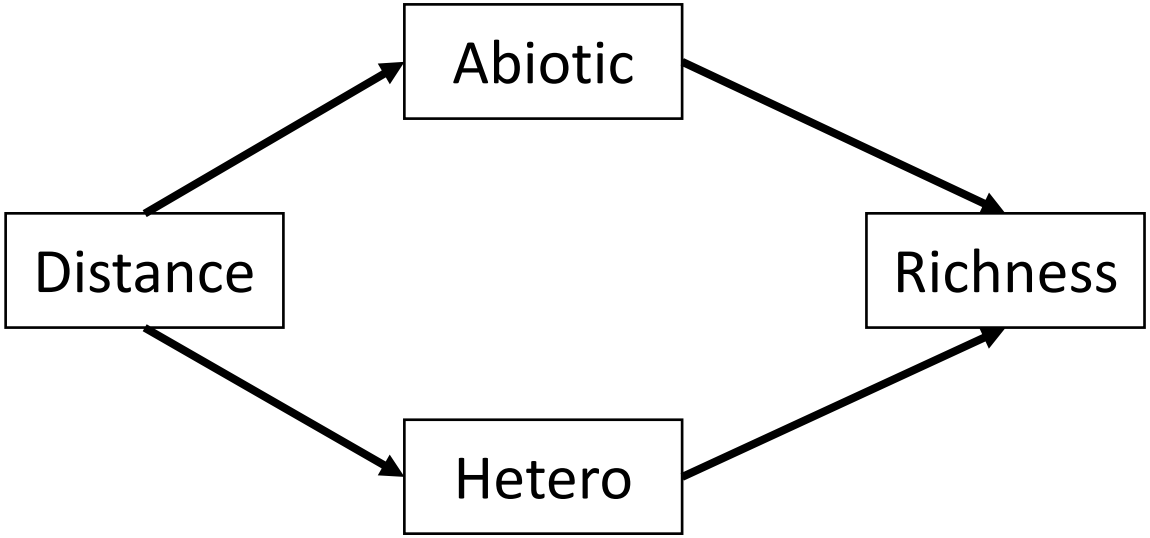
```
install.packages("devtools")
library(devtools)
install_github("jslefche/piecewiseSEM@devel")
```

When you see this, time to code along!

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)

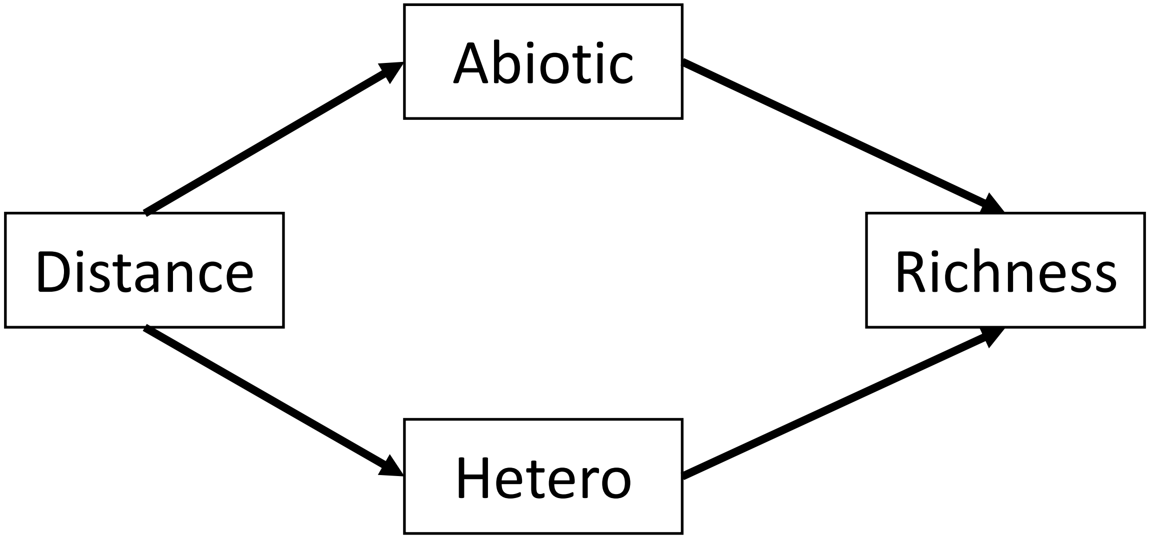
1.4 piecewiseSEM. Keeley example



1. Create list of structured equations
2. Conduct d-sep tests (evaluate fit)
3. Construct χ^2 from log-likelihoods
4. Extract coefficients

1.4 piecewiseSEM. Keeley example

Break this model up into component models



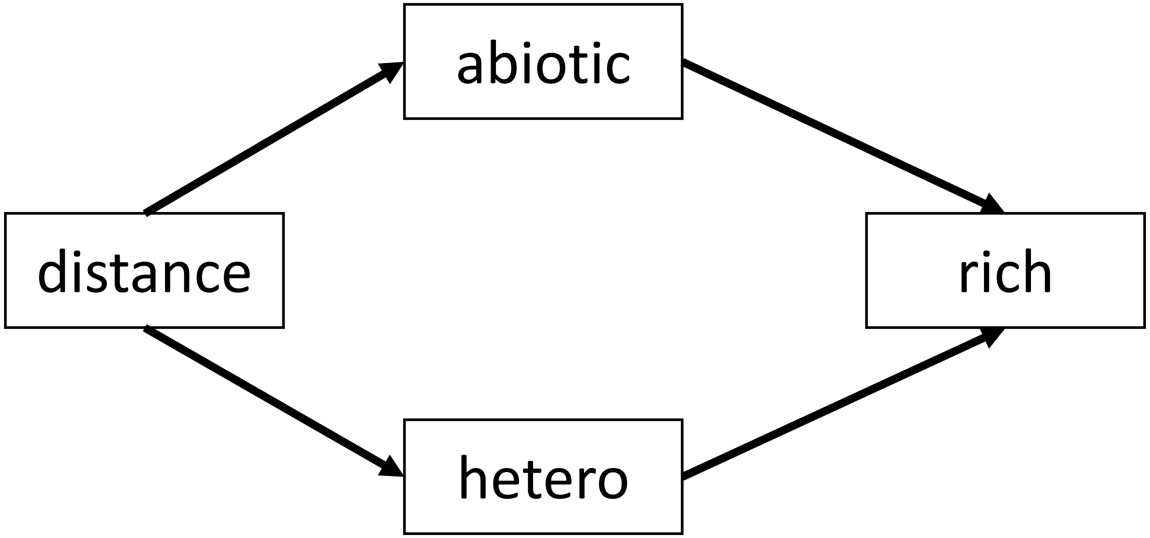
Abiotic

Distance

Richness

Hetero

1.4 piecewiseSEM. Store list of equations

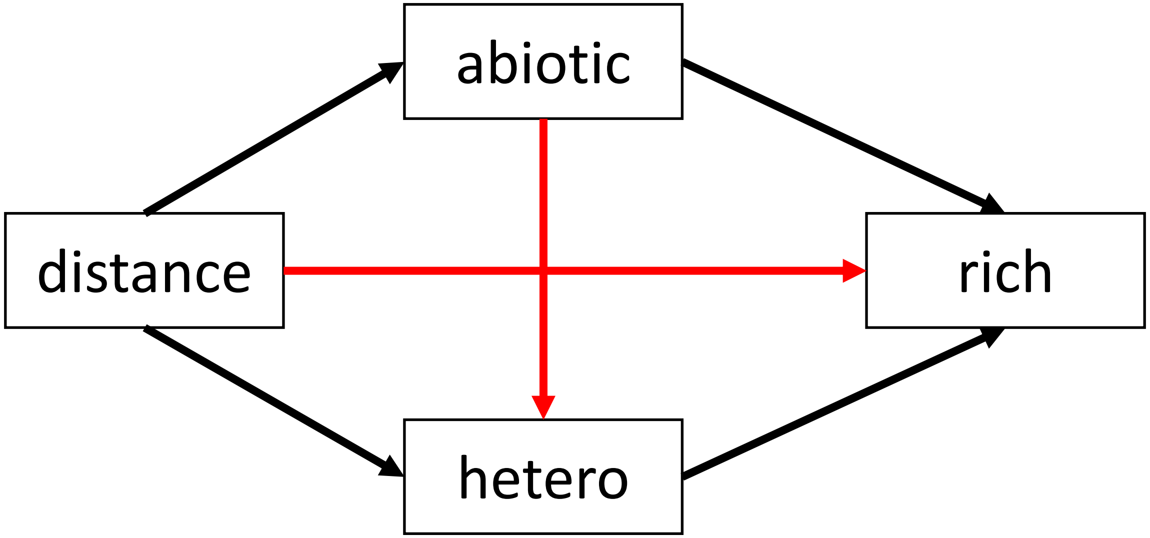


```
# Read in data
data(keeley)

# Create list of structured equations
keeley.sem <- psem(
  lm(abiotic ~ distance, data = keeley),
  lm(hetero ~ distance, data = keeley),
  lm(rich ~ abiotic + hetero, data = keeley),
  data = keeley
)
```

1.4 piecewiseSEM. Store list of equations

```
keeley.sem

Structural Equations of x :
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)"
```

1.4 piecewiseSEM. D-sep tests



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

$`1`
[1] "distance | rich ( abiotic, hetero )"

$`2`
[1] "abiotic | hetero ( distance )"
```

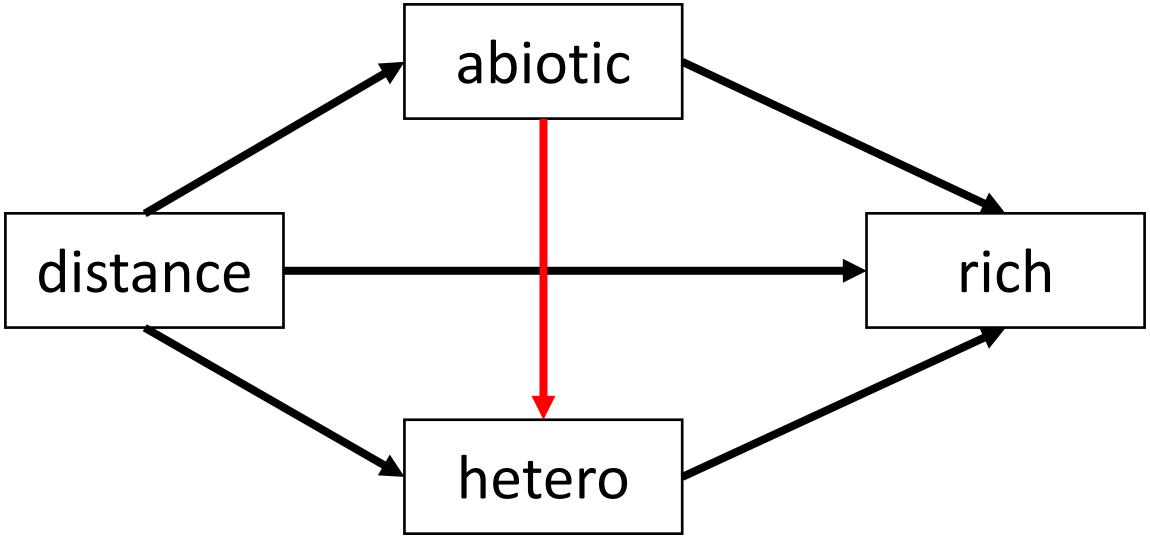
1.4 piecewiseSEM. D-sep tests

```
# Conduct 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     1.0933  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  **

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     1.4296  0.1871       0.1491
2   hetero  distance   0.0036    0.0015 87     2.4742  0.0153       0.2774 *

# 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
```

1.4 piecewiseSEM. D-sep tests

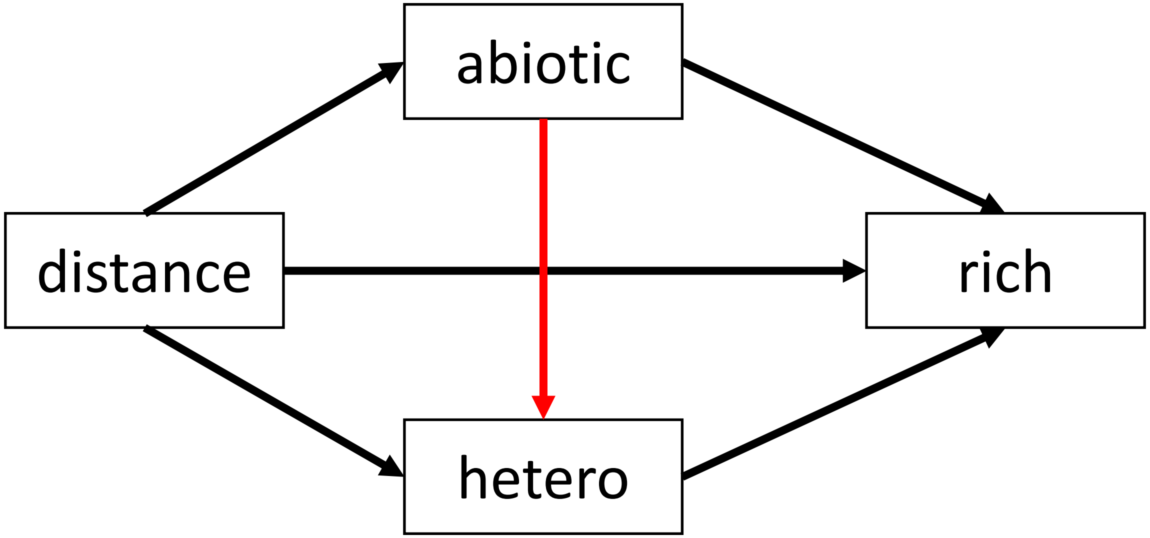
```
# Magically conduct d-sep tests
(keeley.dsep <- dSep(keeley.sem))

# By default the conditioning variables are hidden, but we can show
them
dSep(keeley.sem, conditioning = TRUE)

 Independ.Claim                        Test.Type DF Crit.Value      P.Value
1 rich ~ distance + abiotic + hetero        coef 86   4.093329 9.564005e-05 ***
2        hetero ~ abiotic + distance        coef 87   1.329585 1.871306e-01
```

1.4 piecewiseSEM. Re-assess fit



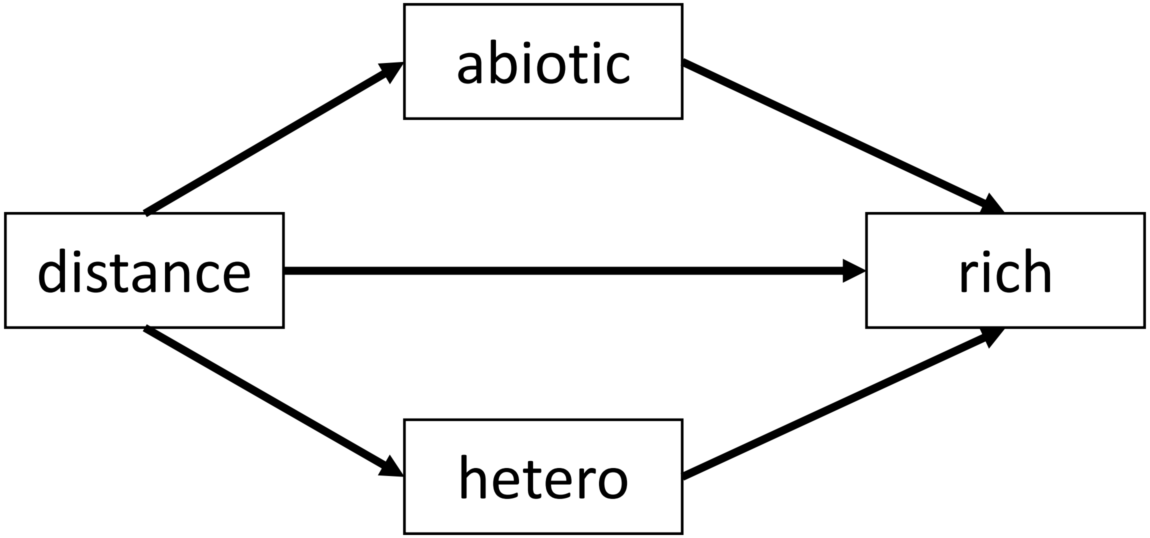
```
# Add significant path back into model
keeley.sem2 <- update(keeley.sem, rich ~ abiotic + hetero + distance)

dSep(keeley.sem2)

fisherC(keeley.sem2)
```

1.4 piecewiseSEM. Re-assess fit



```
            Independ.Claim    Estimate   Std.Error DF Crit.Value   P.Value
1 hetero  ~  abiotic + ... 0.002229248 0.001676649 87   1.429585 0.1871306

  Fisher.C df P.Value
1    3.352  2   0.187
```

1.4 piecewiseSEM. Log-likelihood fit


abiotic

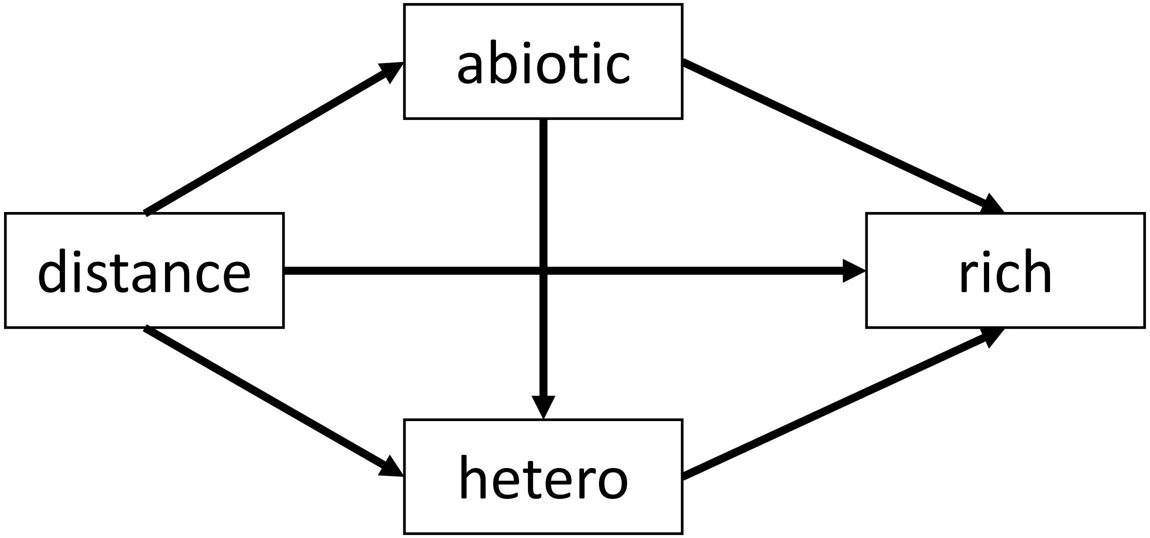
distance

rich

hetero

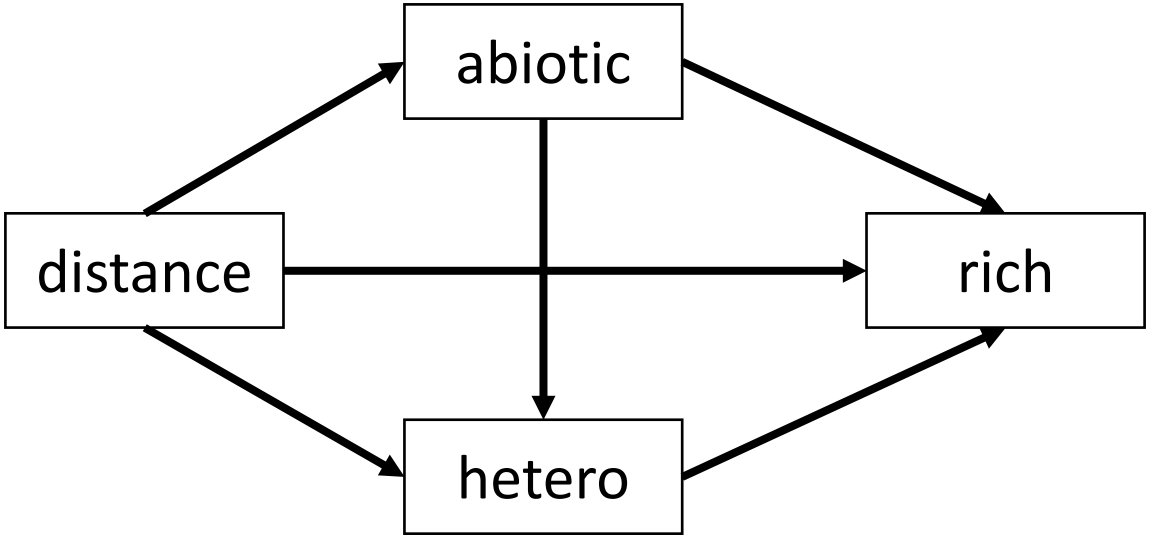
```
# Get log-likelihoods from original model
(M1 <- sapply(keeley.sem2, function(x) ifelse(class(x) == "data.frame", NA, logLik(x))))
                                   data
-299.9828   73.3324 -342.9844        NA

# Sum L-Ls
(M1 <- sum(M1, na.rm = TRUE))
[1] -569.6348
```

1.4 piecewiseSEM. Log-likelihood fit



```
# Fit saturated model (add all missing paths)
Keeley.sem3 <- update(keeley.sem2, hetero ~ abiotic + distance)
```

1.4 piecewiseSEM. Log-likelihood fit



abiotic

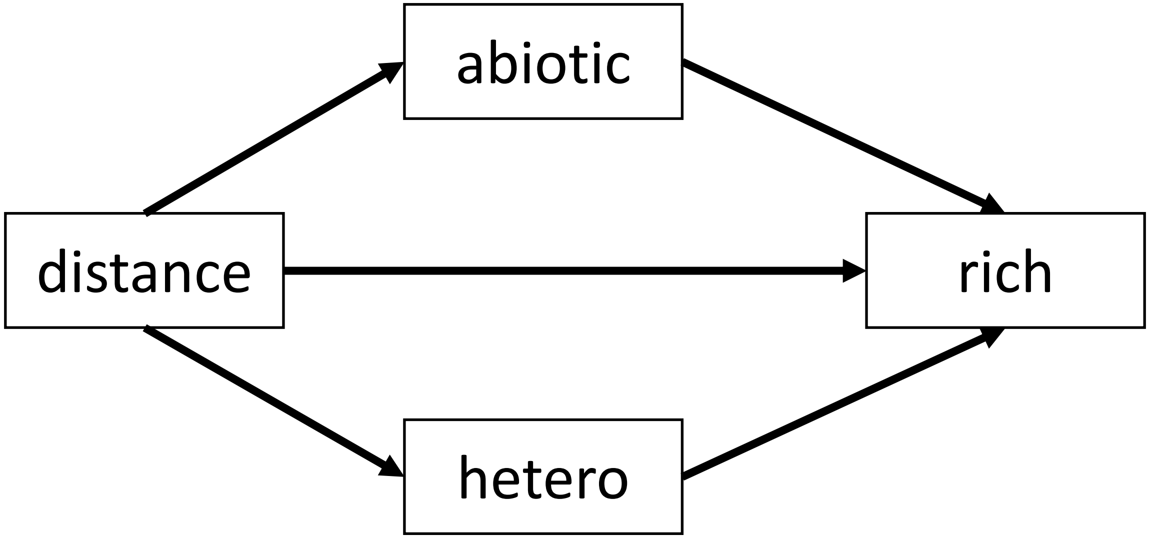
distance

rich

hetero

```
# Get log-likelihoods from saturated model
(M2 <- sapply(keeley.sem3, function(x) ifelse(class(x) == "data.frame",
NA, logLik(x))))
                                       data
-299.98277   74.23761 -342.98438         NA

# Sum L-Ls
(M2 <- sum(M2, na.rm = TRUE))
[1] -568.7295
```

1.4 piecewiseSEM. Log-likelihood fit



abiotic

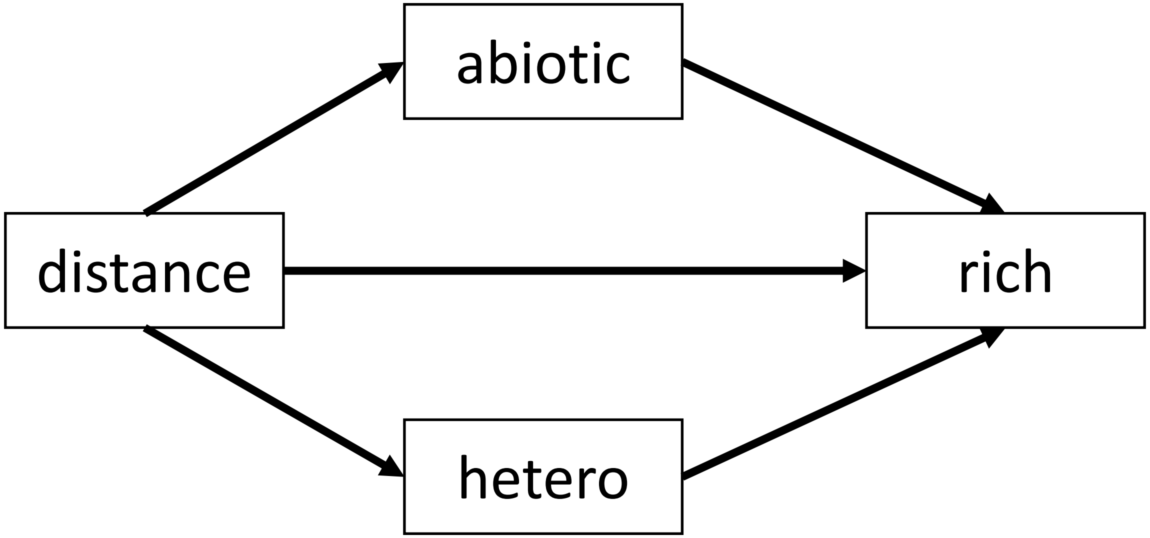
distance

rich

hetero

```
# Compute chi-squared statistic
Chi.sq <- -2*(M1 - M2)

# Compare to chi-squared distribution with 1 d.f. (one additional
estimated parameter in saturated model)
1 - pchisq(Chi.sq, 1)

[1] 0.1784574
```

1.4 piecewiseSEM. Log-likelihood fit



abiotic

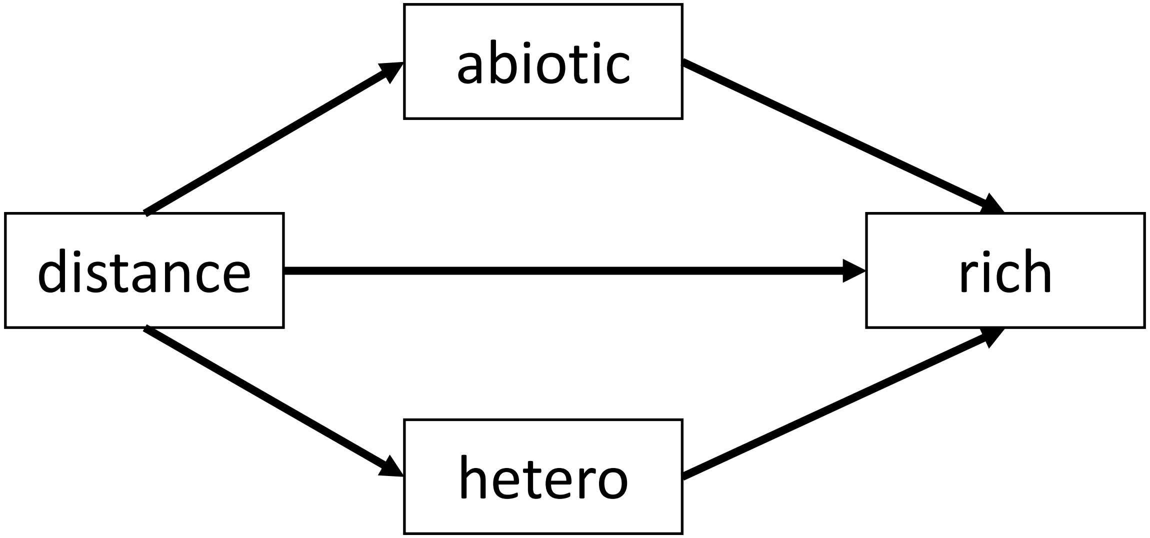
distance

rich

hetero

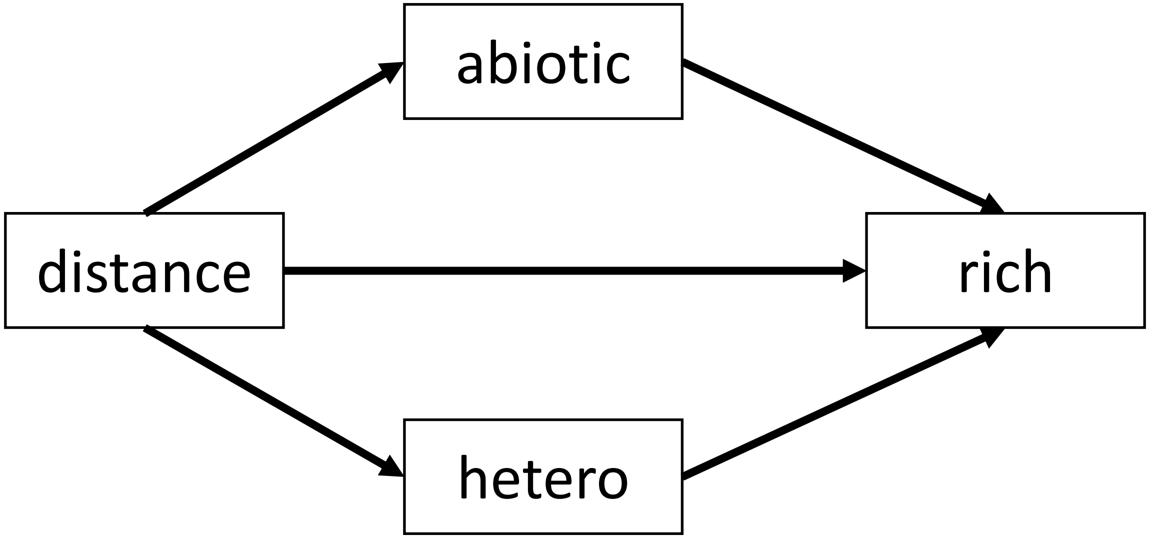
```
# Auto-magic calculation!
LLchisq(keeley.sem2)

  Chisq df P.Value
1  1.81  1   0.178
```

1.4 piecewiseSEM. Log-likelihood fit



```
# Same P-value as from lavaan (chi-squared value too!)
model <- '
abiotic ~ distance
hetero ~ distance
rich ~ abiotic + hetero + distance
'

lavaan::lavInspect(lavaan::sem(model, keeley), "fit")["pvalue"]

   pvalue
0.1784574
```

1.4 piecewiseSEM. Get coefficients



abiotic

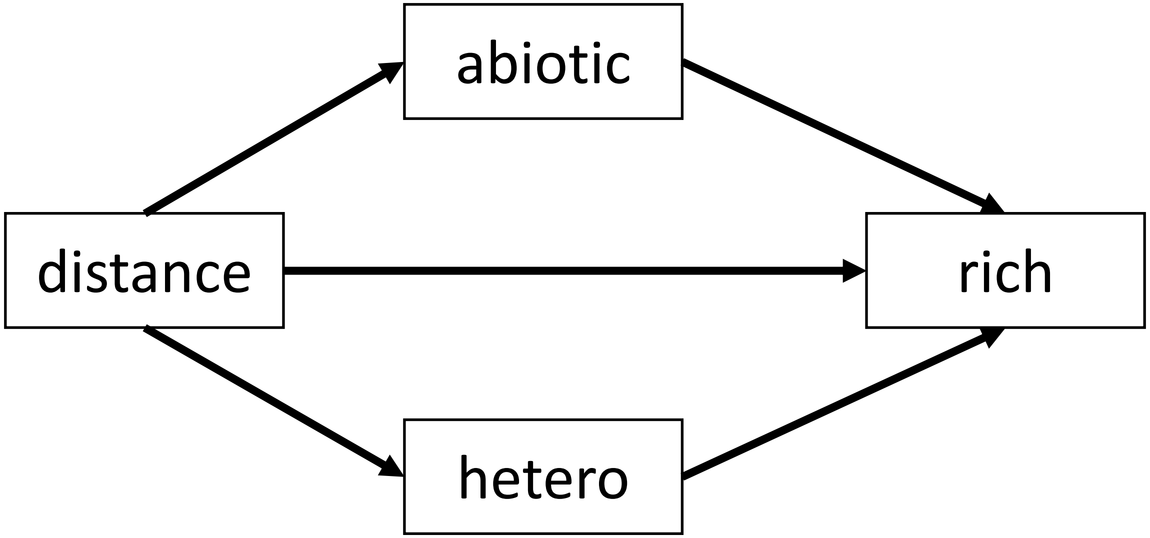
distance

rich

hetero

```
# 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     1.8562  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     1.0933  0.0001       0.3743 ***
```

1.4 piecewiseSEM. Get coefficients


abiotic

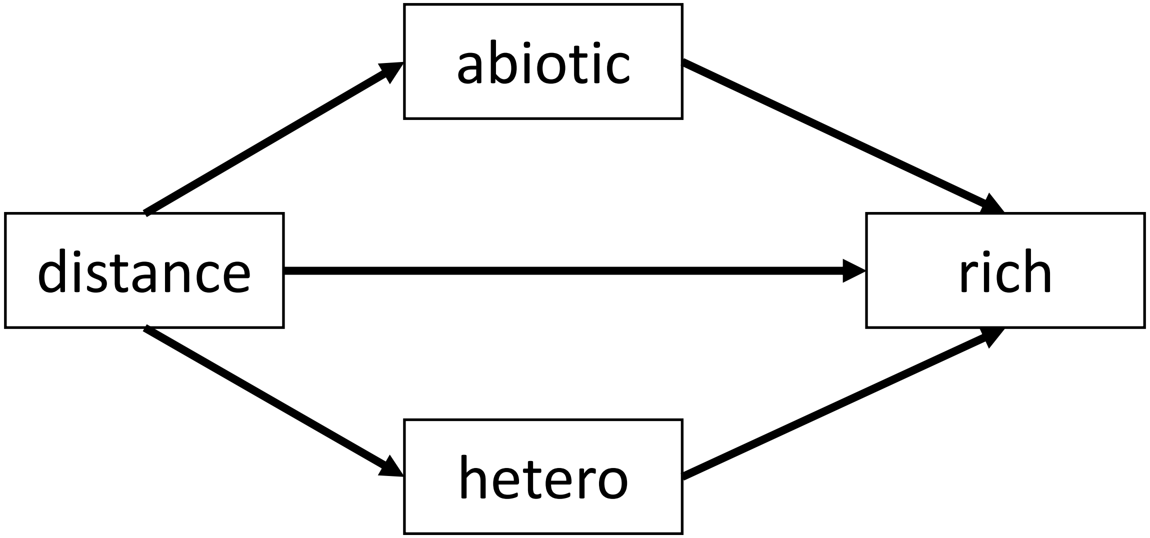
distance

rich

hetero

```
# 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       0.374  **
```

1.4 piecewiseSEM. Get coefficients


abiotic

distance

rich

hetero

```
# 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
```

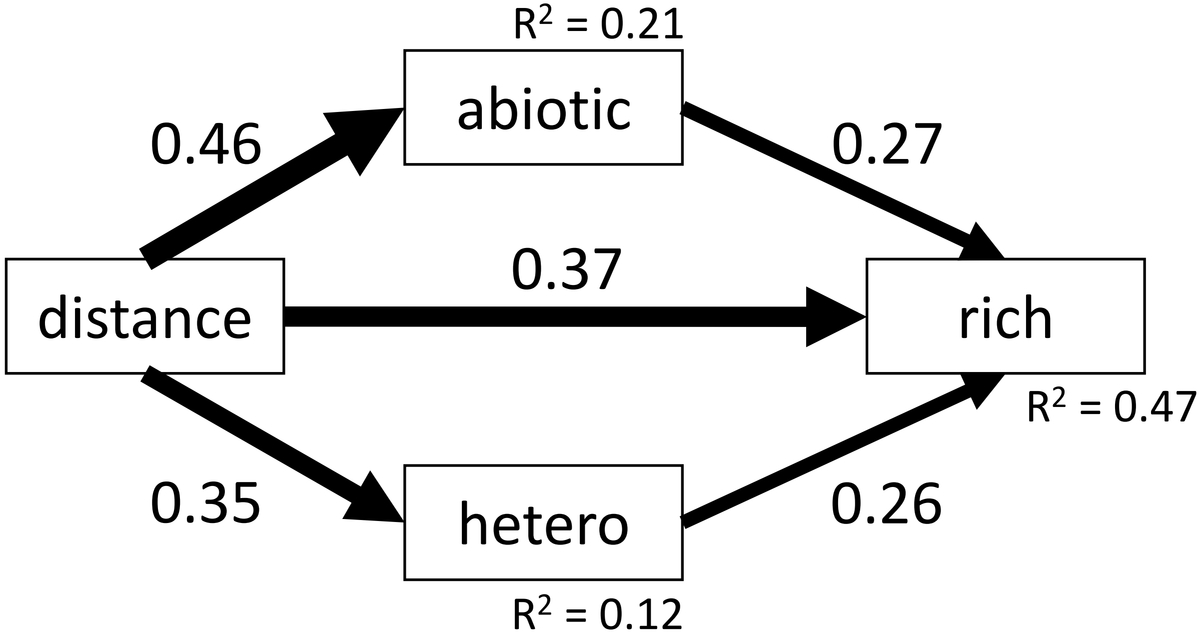
1.4 piecewiseSEM. Summary

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

Structural Equation Model of keeley.sem2

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

    AIC
 1161.270

---
Tests of directed separation:

          Independ.Claim Test.Type DF Crit.Value P.Value
  hetero ~ abiotic + ...      coef 87     1.3296  0.1871

--
Global goodness-of-fit:

Chi-Squared = 1.81 with P-value = 0.178 and on 1 degrees of freedom
Fisher's C = 3.352 with P-value = 0.187 and on 2 degrees of freedom
```

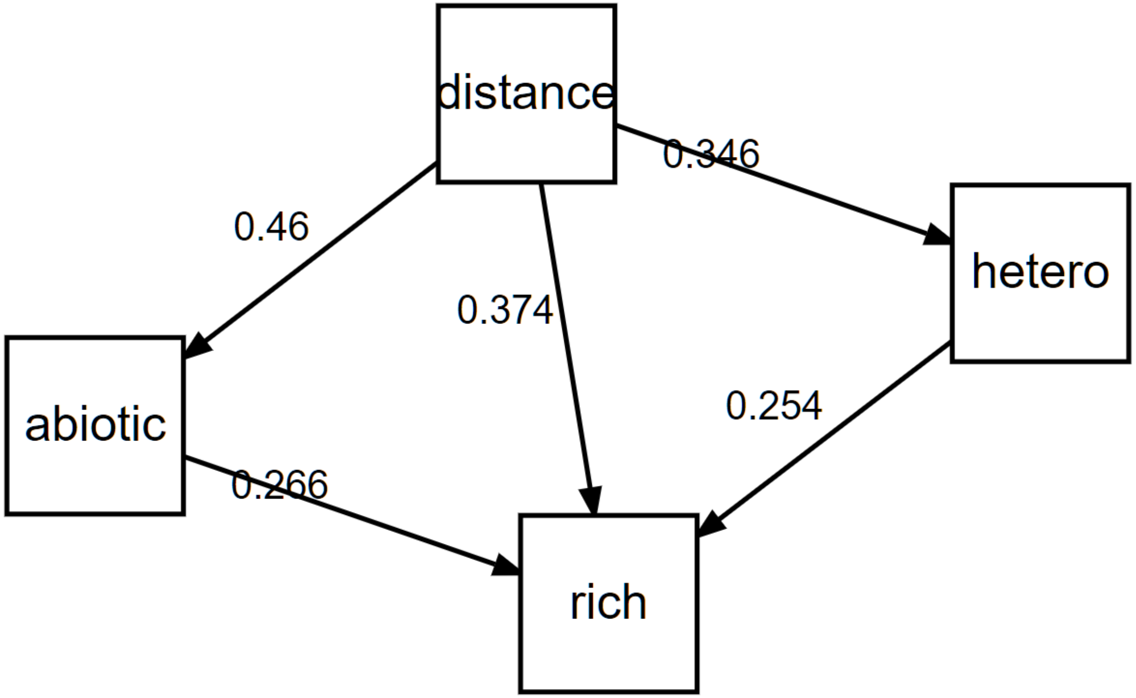
1.4 piecewiseSEM. Summary

```
---
Coefficients:

  Response Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate    
   abiotic  distance   0.3998    0.0823 88     4.8562  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     4.0933  0.0001       0.3743 ***

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

---
Individual R-squared:

  Response method R.squared
   abiotic   none      0.21
    hetero   none      0.12
      rich   none      0.47
```

1.4 piecewiseSEM. Summary

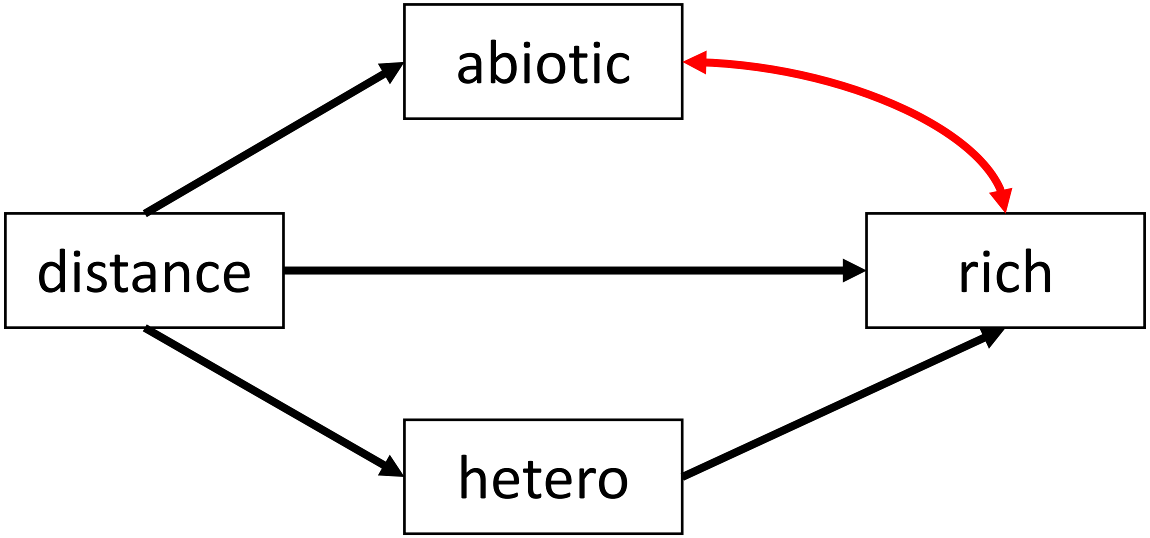


1.4 piecewiseSEM. Summary



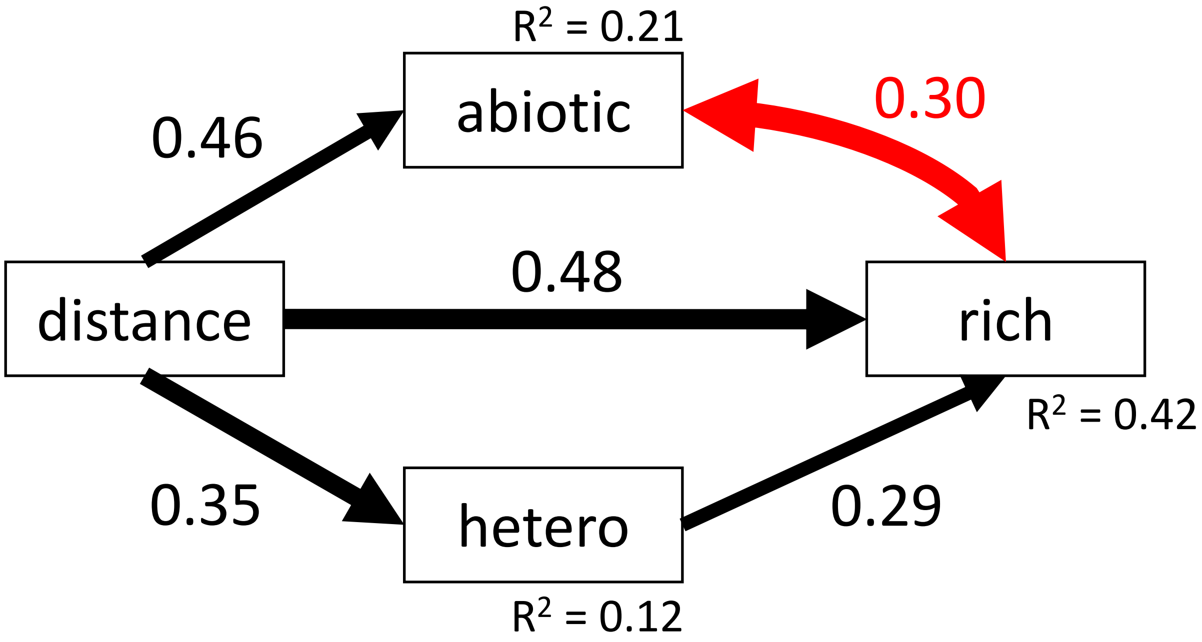
```
# Use built-in plotting function based on `diagrammeR`
plot(keeley.sem2)
```

1.4 piecewiseSEM. Correlated errors

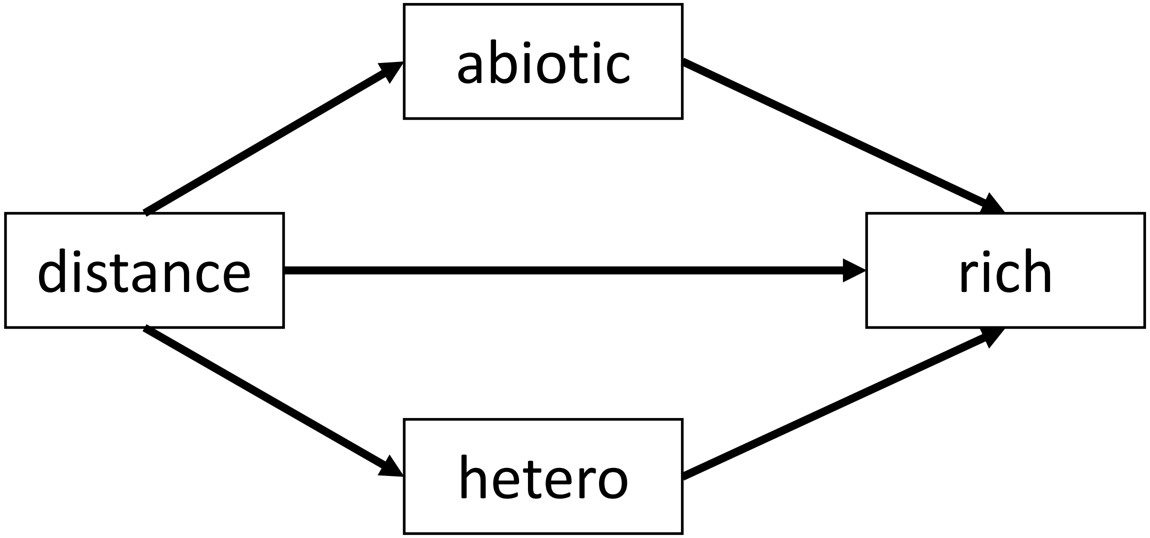


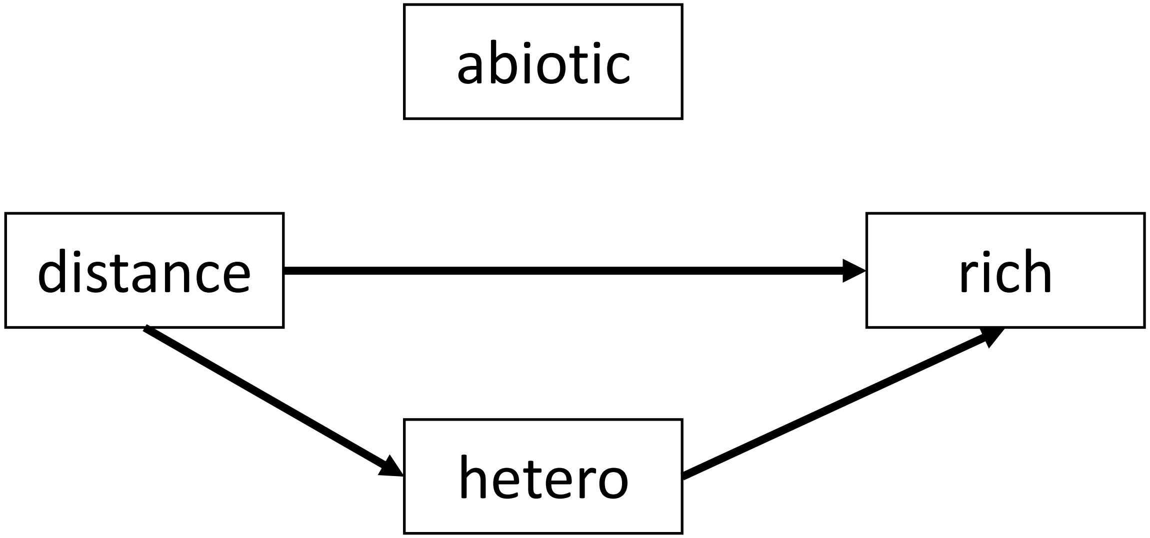
```
keeley.sem3 <- psem(
  lm(abiotic ~ distance, data = keeley),
  lm(hetero ~ distance, data = keeley),
  lm(rich ~ distance + hetero, data = keeley),
  rich %~~% abiotic # same syntax as lavaan
)

summary(keeley.sem3)
```

1.4 piecewiseSEM. Correlated errors

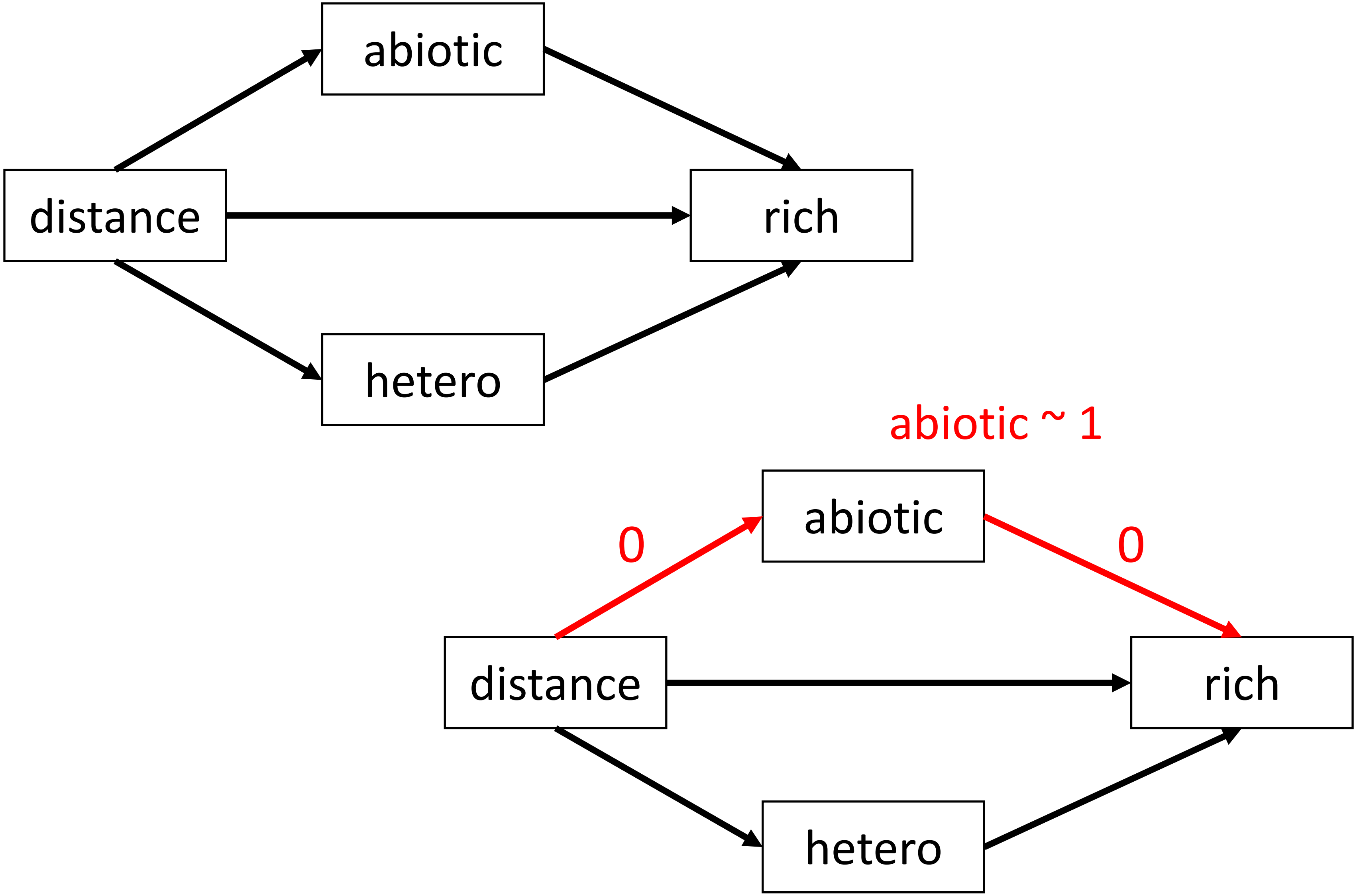


1.4 piecewiseSEM. AIC comparisons

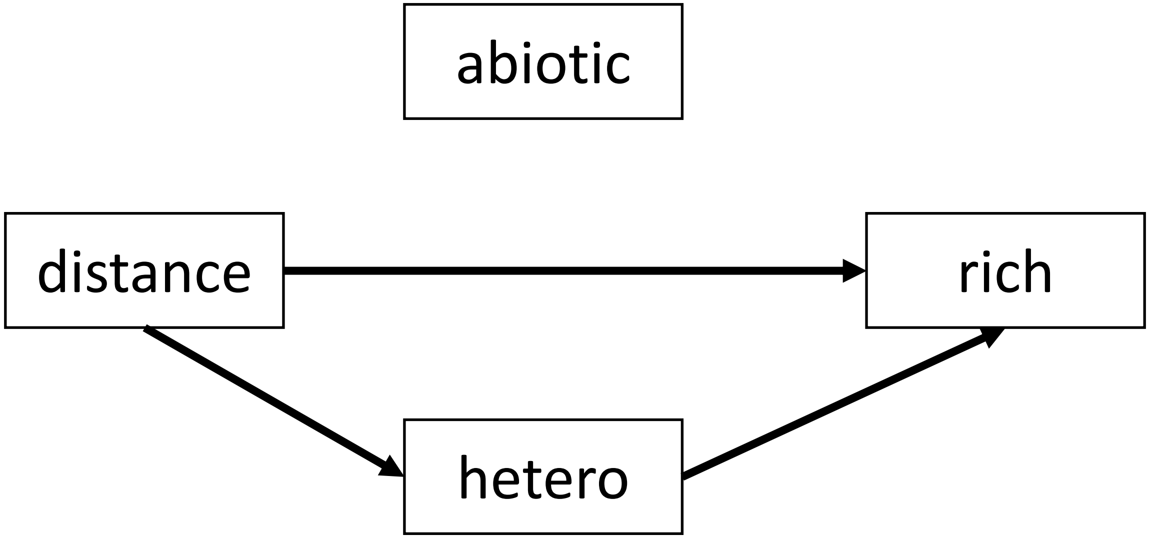




1.4 piecewiseSEM. AIC comparisons

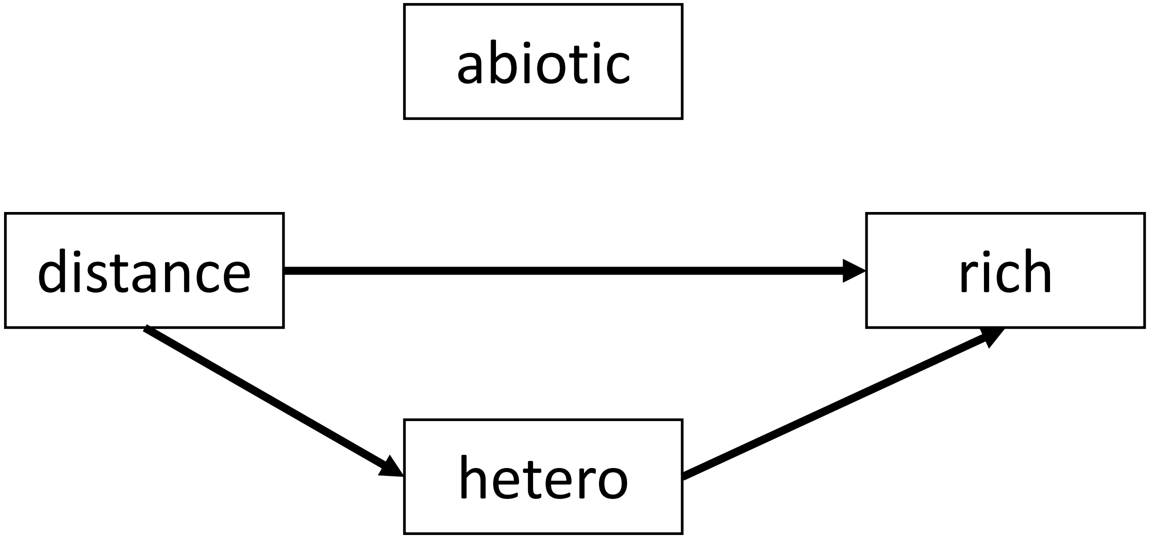


1.4 piecewiseSEM. Fit new model



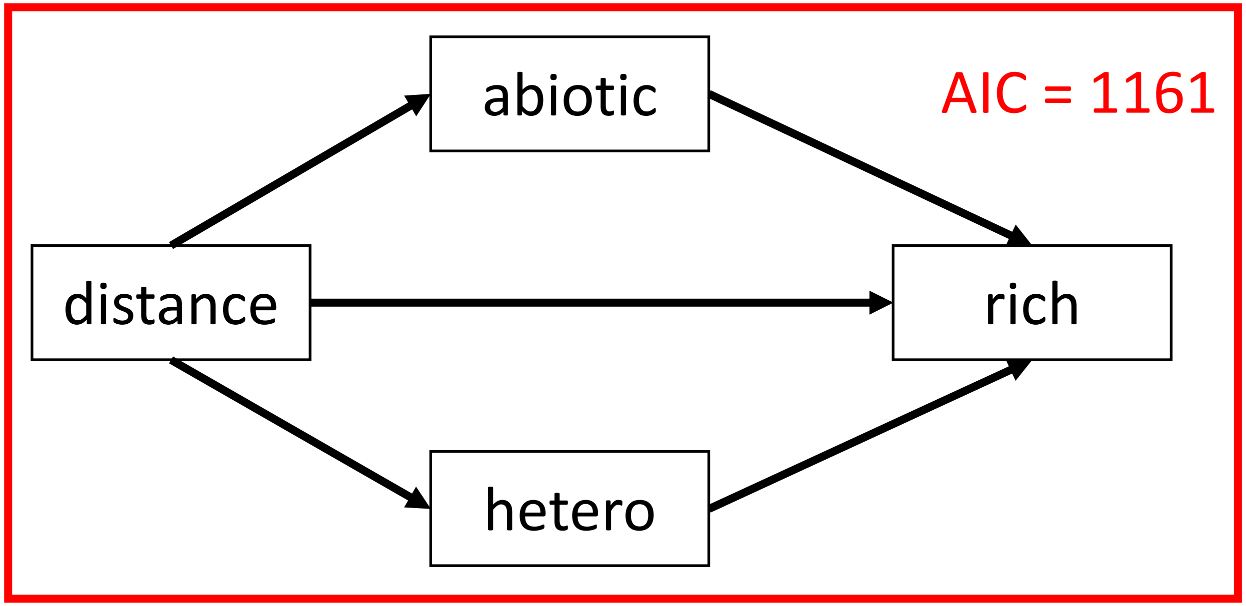
```
# Fit alternate model
keeley.sem4 <- psem(
  lm(hetero ~ distance, data = keeley),
  lm(rich ~ distance + hetero, data = keeley),
  lm(abiotic ~ 1, data = keeley)
)
```

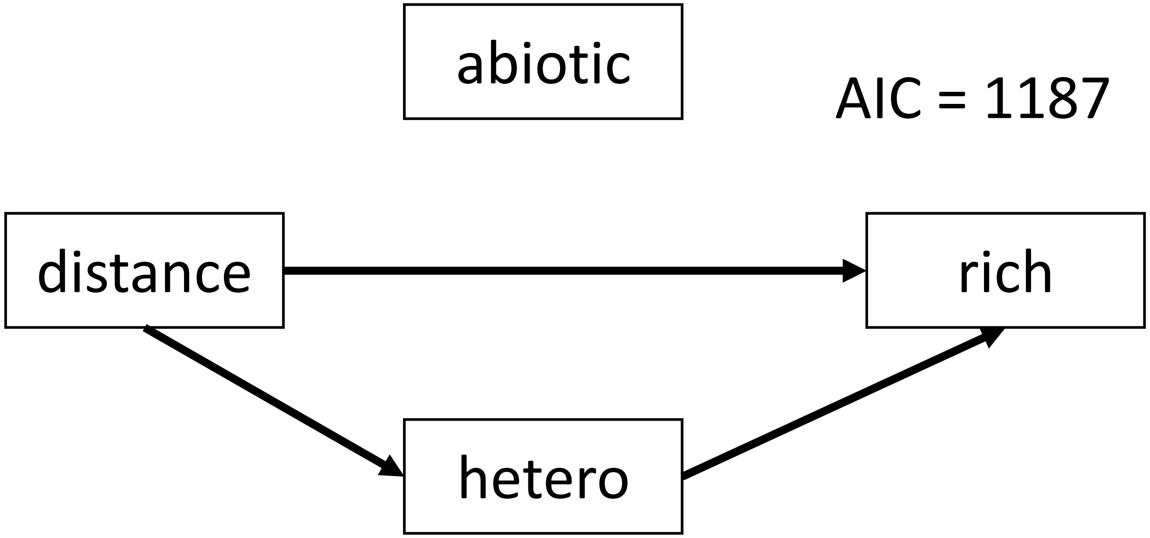
1.4 piecewiseSEM. Fit new model



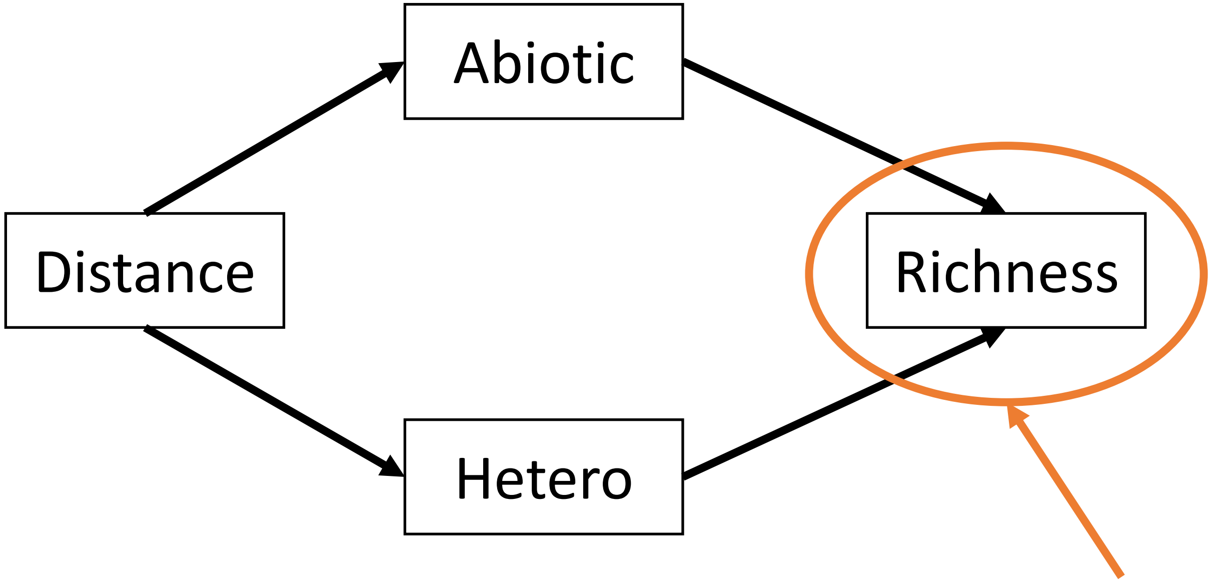
```
# Compare the two models using AIC
AIC(keeley.sem2, keeley.sem4)
  df      AIC
x 11 1161.270
y  9 1187.479
```

1.4 piecewiseSEM. AIC comparisons





1.4 SEM Examples. Refit Keeley using GLM

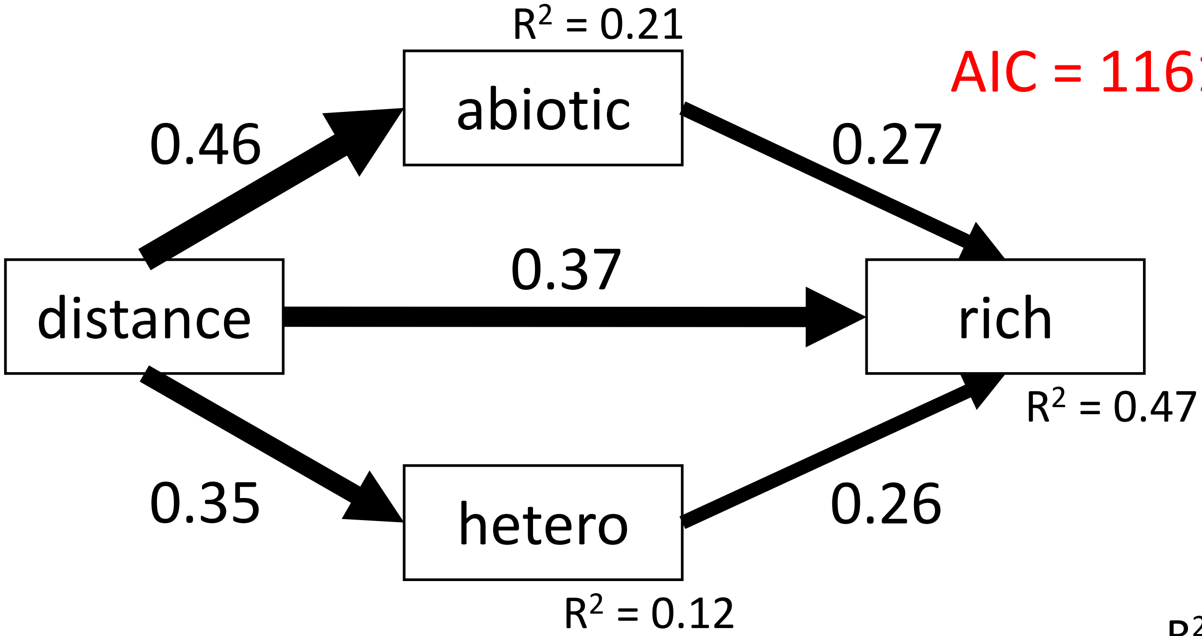


This is a Poisson-distributed variable!

1.4 SEM Examples. Refit Keeley using GLM

```
# Re-run Keeley with GLM for richness
keeley.glm.sem <- psem(
  lm(abiotic ~ distance, data = keeley),
  lm(hetero ~ distance, data = keeley),
  glm(rich ~ abiotic + hetero + distance, family = "poisson", data = keeley),
  keeley
)

summary(keeley.glm.sem)
```

1.4 SEM Examples. Refit Keeley using GLM



$R^2 = 0.21$

abiotic

0.46

0.27

distance

0.37

rich

$R^2 = 0.47$

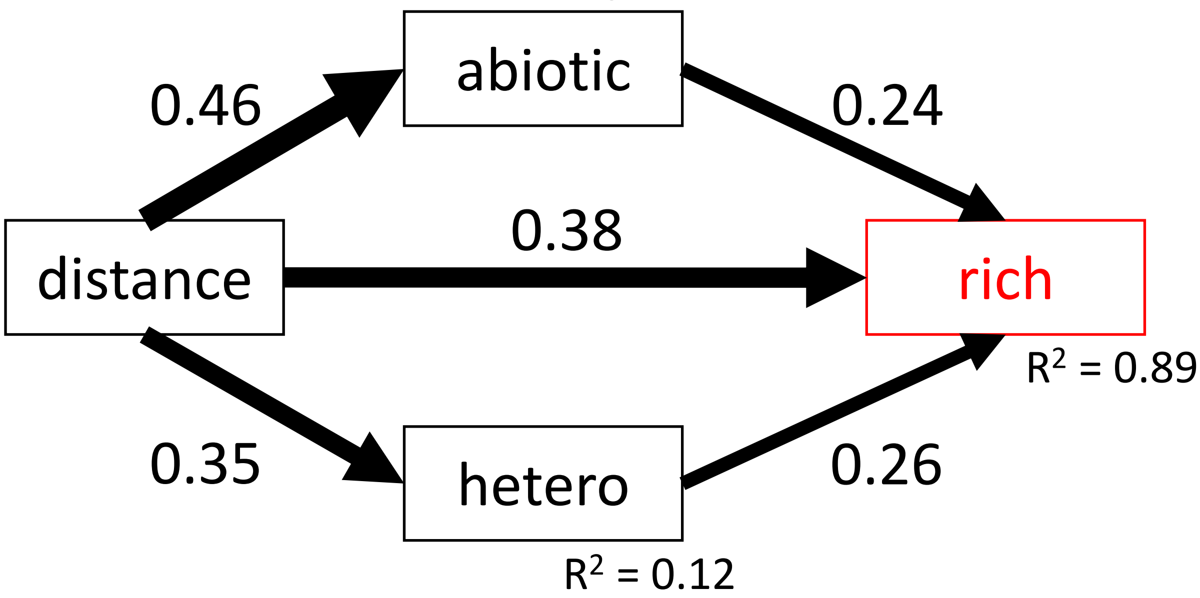
0.35

hetero

0.26

$R^2 = 0.12$

AIC = 1161.3



AIC = 1220.7

$R^2 = 0.21$

abiotic

0.46

0.24

distance

0.38

rich

$R^2 = 0.89$

0.35

hetero

0.26

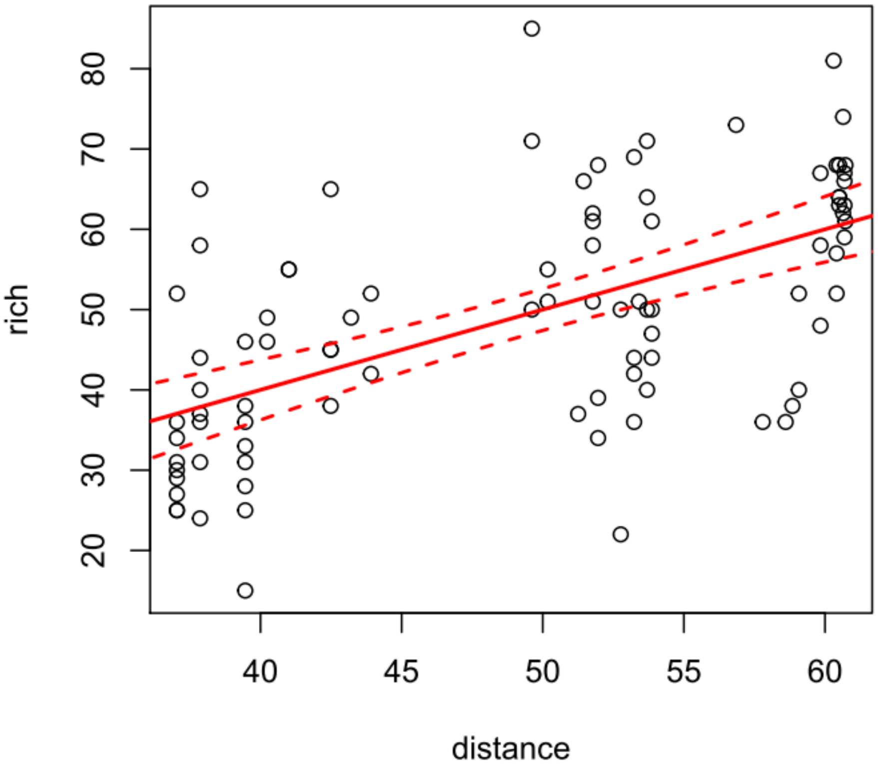
$R^2 = 0.12$

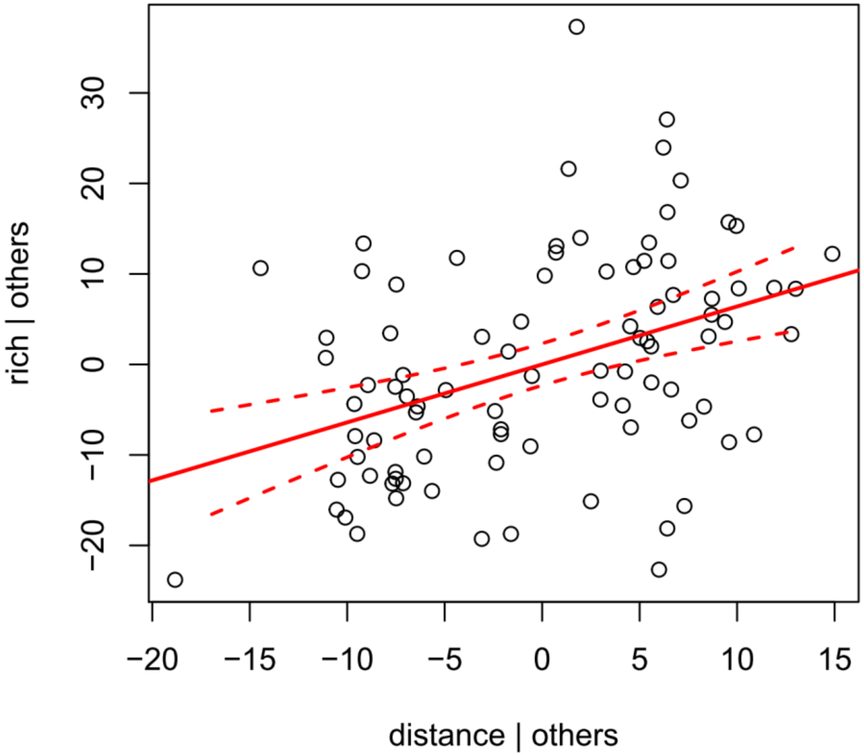
1.4 piecewiseSEM. Partial regression coefficient

Isolate the independent effect of distance on richness:

1. Regress abiotic and hetero against richness (removing distance)
2. Regress distance against abiotic and hetero (remove rich)
3. Regression residuals of 1 against 2 (having removed effects of abiotic and hetero from both)

1.4 piecewiseSEM. Partial regression coefficient





1.4 piecewiseSEM. Partial regression coefficient

- Useful for displaying trends, particularly with complex models where bivariate correlations are messy

- Can be used for any multiple regression (single model or list)

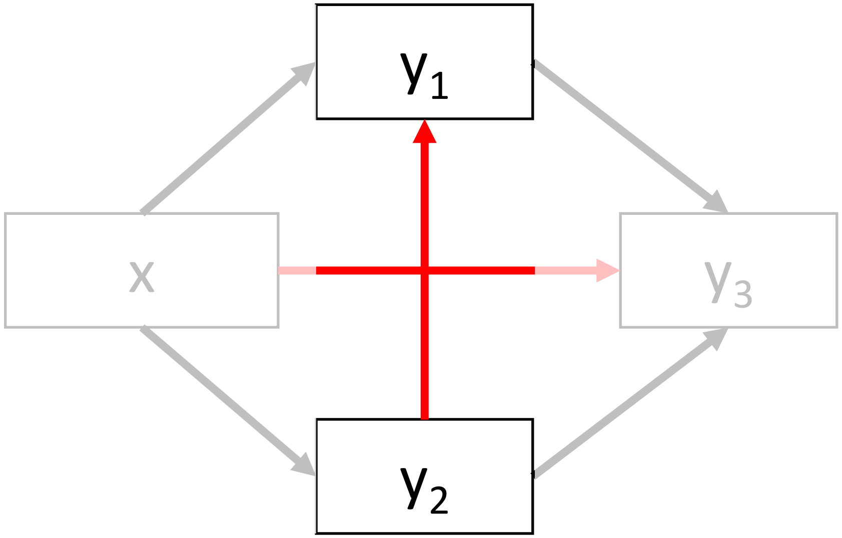
- Not applicable to simple regression (*Y* ~ *X*) for obvious reasons

- Well implemented in *emmeans* package

1.5 A Warning…

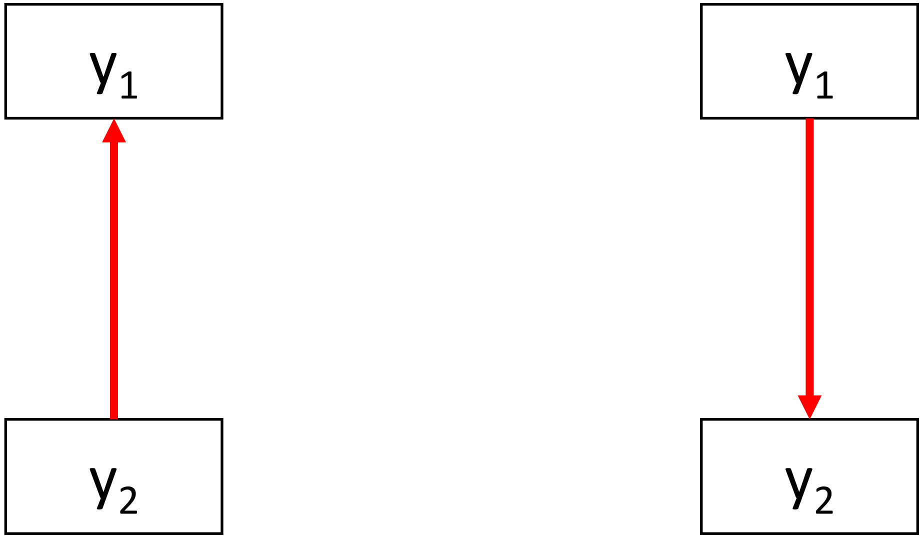
1.5 Directed Separation. A warning

- Intermediate non-normal endogenous variables pose a challenge



1.5 Directed Separation. A warning

- If normal, significance values are reciprocal



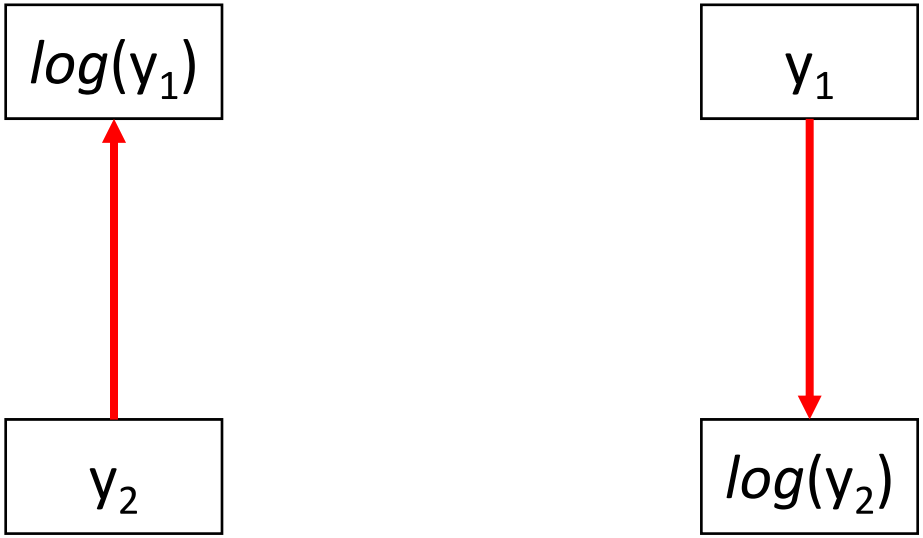
$$P_{2,1} \quad == \quad P_{1,2}$$

1.5 Directed Separation. A warning

```
set.seed(66)

data <- data.frame(x = rnorm(100), y1 = rnorm(100), y2 = rpois(100,
10), y3 = rnorm(100))

# Show that y2 ~ y1 is the same as y2 ~ y1 for LM
mody1.y2 <- lm(y1 ~ y2 + x, data)

mody2.y1 <- lm(y2 ~ y1 + x, data)

summary(mody1.y2)$coefficients[2, 4]

[1] 0.7429784

summary(mody2.y1)$coefficients[2, 4]

[1] 0.7429784
```

1.5 Directed Separation. A warning

- If non-normal, significance values are *not* reciprocal because of transformation via link function



$log(y_1)$

y_2

y_1

$log(y_2)$

$$P_{2,1} \neq P_{1,2}$$

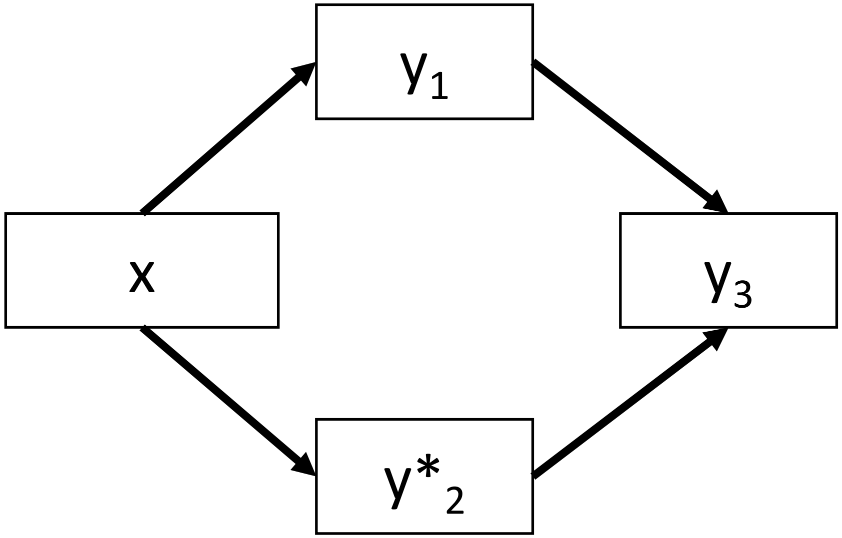
1.5 Directed Separation. A warning

```
# Show that y2 ~ y1 is not the same as y2 ~ y1 for GLM
mody1.y2 <- lm(y1 ~ y2 + x, data)

mody2.y1.glm <- glm(y2 ~ y1 + x, "poisson", data)

summary(mody1.y2)$coefficients[2, 4]

[1] 0.7429784

summary(mody2.y1.glm)$coefficients[2, 4]

[1] 0.8036267
```

1.5 Directed Separation. A warning

```
# Same is true for log-likelihoods
logLik(mody1.y2)
'log Lik.' -128.1663 (df=4)

logLik(mody2.y1.glm)
'log Lik.' -239.3152 (df=3)

# Because of differences in ML-fitting function for Gaussian vs.
Poisson GLM
```

1.5 Directed Separation. A warning



```
# Create SEM with GLM
modelList <- psem(
  lm(y1 ~ x, data),
  glm(y2 ~ x, "poisson", data),
  lm(y3 ~ y1 + y2, data),
  data
)
```

1.5 Directed Separation. A warning

```
# Run summary
summary(modelList)

 Error:
Non-linearities detected in the basis set where P-values are not symmetrical.
This can bias the outcome of the tests of directed separation.

Offending independence claims:
 y2 <- y1 *OR* y2 -> y1

Option 1: Specify directionality using argument 'direction = c()'.

Option 2: Remove path from the basis set by specifying as a correlated error using
'%~~%'.

Option 3: Use argument 'conserve = TRUE' to compute both tests, and return the
most conservative P-value.
```

1.5 Directed Separation. A warning

```
# Address conflict using conserve = T
summary(modelList, conserve = T)

dSep(modelList, conserve = T)

  Independ.Claim    Estimate  Std.Error DF Crit.Value   P.Value
1  y3 ~ x + ... -0.01414678 0.09749775 96 -0.1450985 0.8849373
3 y1 ~ y2 + ... -0.01161551 0.03532167 97 -0.3288495 0.7429784

# Check against
summary(mody1.y2)$coefficients[2, 4]

[1] 0.7429784

summary(mody2.y1.glm)$coefficients[2, 4]

[1] 0.8036267
```

1.5 Directed Separation. A warning

```
# Address conflict using direction = c()
dSep(modelList, direction = c("y2 <- y1"))

 Independ.Claim    Estimate  Std.Error DF Crit.Value   P.Value
1   y3 ~ x + ... -0.01414678 0.09749775 96 -0.1450985 0.8849373
2  y1 ~ y2 + ... -0.01161551 0.03532167 97 -0.3288495 0.7429784

dSep(modelList, direction = c("y1 <- y2"))

 Independ.Claim    Estimate  Std.Error DF Crit.Value   P.Value
1   y3 ~ x + ... -0.01414678 0.09749775 96 -0.1450985 0.8849373
2  y2 ~ y1 + ... -0.00872099 0.03507248 97 -0.2486562 0.8036267
```

1.5 Directed Separation. A warning

```
# Address conflict using correlated errors
modelList2 <- update(modelList, y2 %~~% y1)

dSep(modelList2)

 Independ.Claim    Estimate  Std.Error DF Crit.Value   P.Value
1   y3 ~ x + ... -0.01414678 0.09749775 96 -0.1450985 0.8849373
```


Emergency cat nap!
pokeme.com