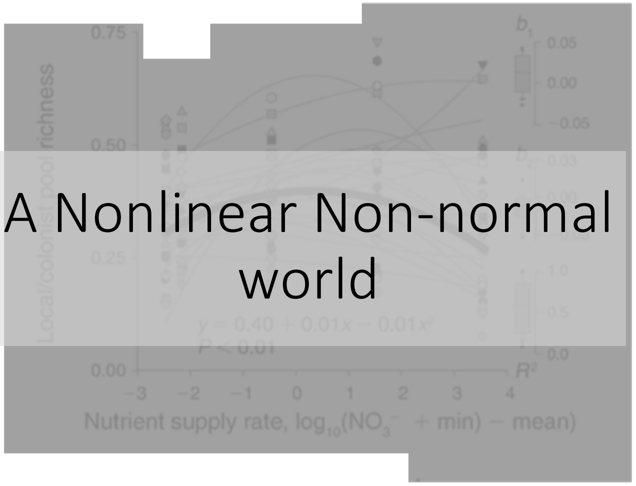
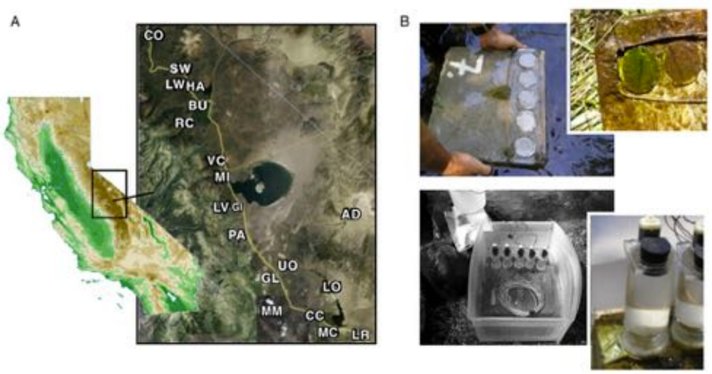


A Nonlinear Non-normal world

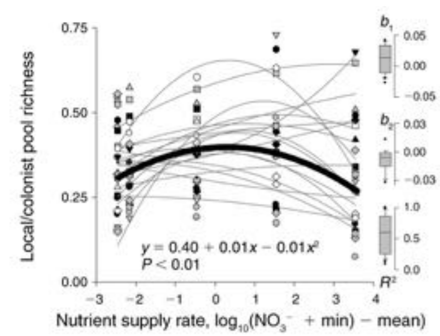
Non-linearities and More

1. Non-linearities in models
   - Polynomial Terms
   - Interactions
2. Generalized Linear Models
3. Special Considerations for GLMs and Dsep
4. Standardized Nonlinear Coefficients

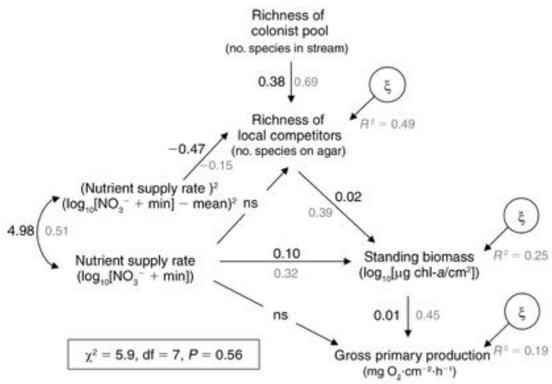
Cardinale et al. (2009) - Does diversity drive productivity or vice versa?



Non-linearities. Cardinale (2009)



Non-linearities. Cardinale (2009)

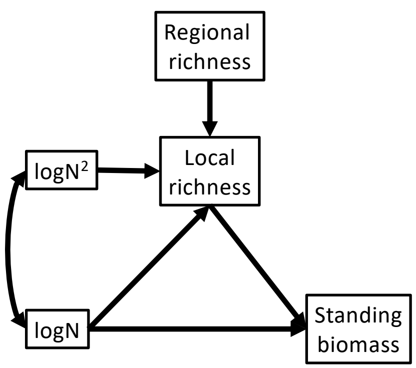


Cardinale et al. 2009

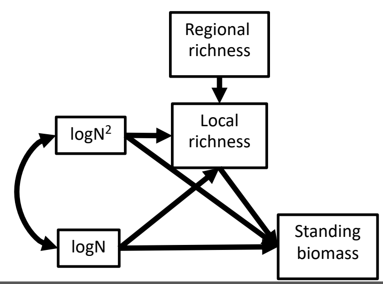
Non-linearities

- Code as independent linear and squared, cubed, etc. terms
- Currently squirrely if transformed inside model formula, e.g., `lm(y ~ poly(x, 2))`, `I(X^2)`

Non-linearities. Cardinale (2009)

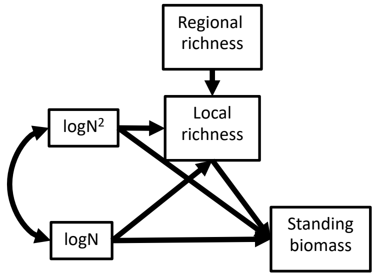


Non-linearities. Cardinale (2009)



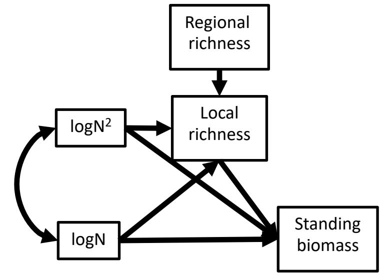

```
# Load data
cardinale <- read.csv("../data/cardinale.csv")

# Take log of N and N^2
cardinale$logN <- log10(cardinale$N + 1e-6)

cardinale$logN2 <- cardinale$logN ^ 2

# Take log of chl (standing biomass)
cardinale$logChl <- log10(cardinale$Chl)
```

Non-linearities. Cardinale (2009)



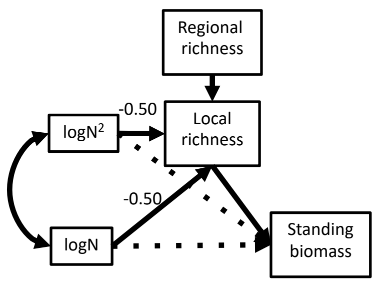

```
# Create SEM
cardinale.sem <- psem(
  lm(SA ~ logN + logN2 + SR, data = cardinale),
  lm(logChl ~ SA + logN + logN2, data =  cardinale),
  cardinale
)
```

Non-linearities. Cardinale (2009)



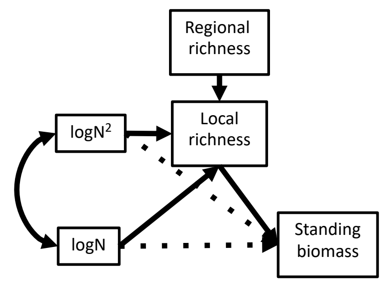

```
> coefs(cardinale.sem)
  Response Predictor Estimate Std.Error  DF Crit.Value P.Value Std.Estimate
1       SA      logN  -2.9944    1.5375 123    -1.9476  0.0537      -0.5044
2       SA     logN2  -0.4742    0.2424 123    -1.9568  0.0526      -0.5067
3       SA        SR   0.3838    0.0359 123    10.6844  0.0000       0.6893 ***
4   logChl        SA   0.0201    0.0040 123     5.0327  0.0000       0.3946 ***
5   logChl      logN   0.1168    0.0953 123     1.2258  0.2226       0.3858
6   logChl     logN2   0.0032    0.0150 123     0.2108  0.8334       0.0664
```

Non-linearities. Cardinale (2009)



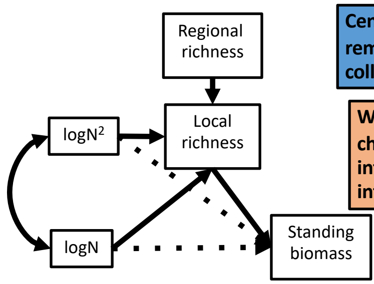

```
> coefs(cardinale.sem)
  Response Predictor Estimate Std.Error  DF Crit.Value P.Value Std.Estimate
1       SA      logN  -2.9944    1.5375 123    -1.9476  0.0537      -0.5044
2       SA     logN2  -0.4742    0.2424 123    -1.9568  0.0526      -0.5067
3       SA        SR   0.3838    0.0359 123    10.6844  0.0000       0.6893 ***
4   logChl        SA   0.0201    0.0040 123     5.0327  0.0000       0.3946 ***
5   logChl      logN   0.1168    0.0953 123     1.2258  0.2226       0.3858
6   logChl     logN2   0.0032    0.0150 123     0.2108  0.8334       0.0664
```

Non-linearities. Cardinale (2009)




```
> with(cardinale, cor(logN, logN2))

[1] -0.9684737
```

Centering Reduces Correlation

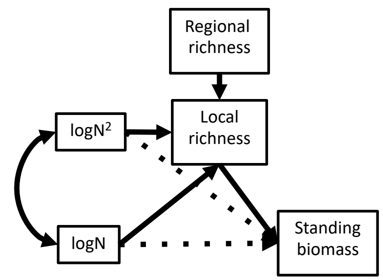


Centering helps remove collinearities.

Warning: it does change interpretation of interaction effects.

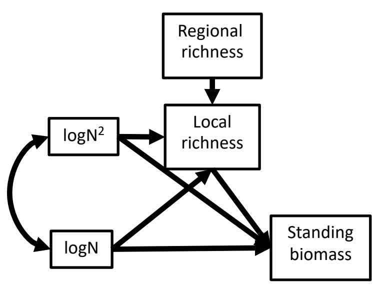
```
# Center polynomial to reduce collinearity
cardinale$logN.cen = scale(cardinale$logN, scale = F)

cardinale$logN2.cen = scale(cardinale$logN, scale = F) ^ 2
```

Centering Reduces Correlation



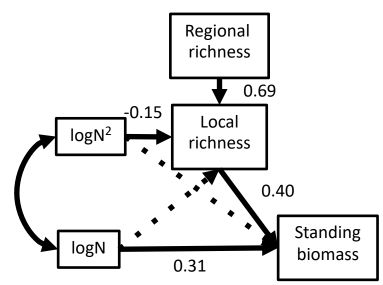

```
> cor(cardinale$logN.cen, cardinale$logN2.cen)
          [,1]
[1,] 0.5126311
```

Centering Reduces Correlation



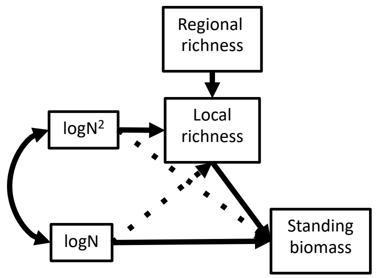

```
# Re-fit SEM using centered predictors
cardinale.sem2 <- psem(
  lm(SA ~ logN.cen + logN2.cen + SR, data = cardinale),
  lm(logChl ~ SA + logN.cen + logN2.cen, data = cardinale),
  cardinale
)
```

Centering Reduces Correlation



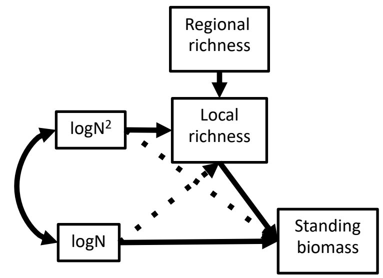

```
> coefs(cardinale.sem2)
  Response Predictor Estimate Std.Error  DF Crit.Value P.Value Std.Estimate
1       SA  logN.cen   0.3668    0.4460 123     0.8223  0.4125       0.0618
2       SA logN2.cen  -0.4742    0.2424 123    -1.9568  0.0526      -0.1470
3       SA        SR   0.3838    0.0359 123    10.6844  0.0000       0.6893 ***
4   logChl        SA   0.0201    0.0040 123     5.0327  0.0000       0.3946 ***
5   logChl  logN.cen   0.0944    0.0275 123     3.4320  0.0008       0.3116 ***
6   logChl logN2.cen   0.0032    0.0150 123     0.2108  0.8334       0.0193
```

Centering Reduces Correlation



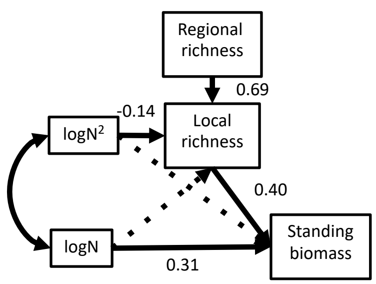

```
> dSep(cardinale.sem2)

      Independ.Claim    Estimate   Std.Error  DF Crit.Value   P.Value
1 logChl  ~  SR + ... 0.002044185 0.003078878 122  0.6639384 0.5079826
```

With lavaan



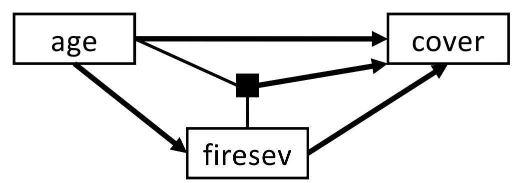

```
cardinale_mod <-'
SA ~ logN.cen + logN2.cen + SR
logChl ~ SA + logN.cen + logN2.cen
logN.cen ~~ logN2.cen
'

cardinale_fit <- sem(cardinale_mod, data=cardinale, fixed.x=F)
```

With lavaan…



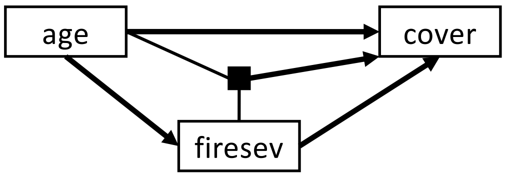

```
Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  SA ~
    logN.cen          0.367    0.439    0.836    0.403    0.367    0.062
    logN2.cen        -0.474    0.238   -1.989    0.047   -0.474   -0.147
    SR                0.384    0.035   10.858    0.000    0.384    0.028
  logChl ~
    SA                0.020    0.004    5.114    0.000    0.020    0.395
    logN.cen          0.094    0.027    3.486    0.000    0.094    0.312
    logN2.cen         0.003    0.015    0.214    0.831    0.003    0.019
```

Interaction Effects



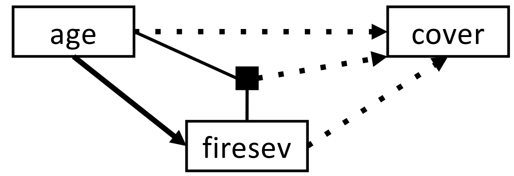

```
keeley_int <- psem(

  lm(cover ~ age*firesev, data = keeley),

  lm(firesev ~ age, data = keeley),

  data = keeley

)
```

Model is Saturated



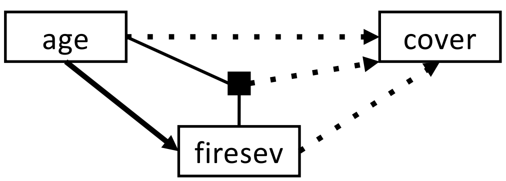

```
> fisherC(keeley_int)
  Fisher.C df P.Value
1        0  0       1
```

Where did the firesev effect go?



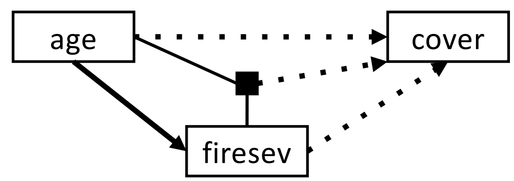

```
> coefs(keeley_int)
  Response   Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate
1    cover         age   0.0045    0.0067 86     0.6786  0.4992       0.1800
2    cover     firesev  -0.0149    0.0398 86    -0.3729  0.7101      -0.0774
3    cover age:firesev  -0.0021    0.0014 86    -1.5263  0.1306      -0.5700
4  firesev         age   0.0597    0.0125 88     4.7781  0.0000       0.4539 ***
```

Is this a problem?



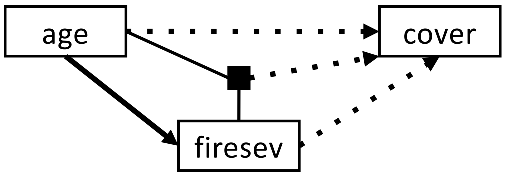

```
> with(keeley, cor(age, age*firesev))
[1] 0.8687952
```

Centering for Interactions



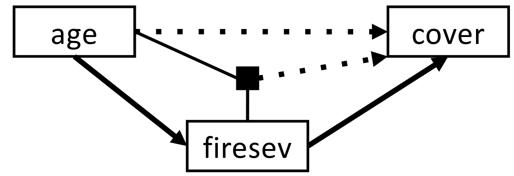

```
keeley$fire_cent <- scale(keeley$firesev, scale=FALSE)
keeley$age_cent <- scale(keeley$age, scale=FALSE)
```

Centering for Interactions



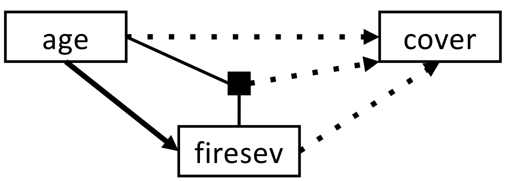

```
keeley_int_cent <- psem(

  lm(cover ~ age_cent *fire_cent, data = keeley),

  lm(fire_cent ~ age_cent, data = keeley),


  data = keeley

)
```

Additive Effect Recovered



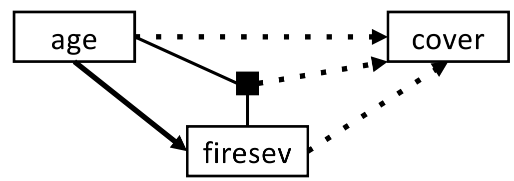

```
> coefs(keeley_int_cent)
   Response           Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate
1     cover            age_cent  -0.0050    0.0027 86    -1.8810  0.0634      -0.1985
2     cover           fire_cent  -0.0684    0.0203 86    -3.3752  0.0011      -0.3561  **
3     cover age_cent:fire_cent  -0.0021    0.0014 86    -1.5263  0.1306      -0.1438
4 fire_cent            age_cent   0.0597    0.0125 88     4.7781  0.0000       0.4539 ***
```

In Lavaan Same Problem



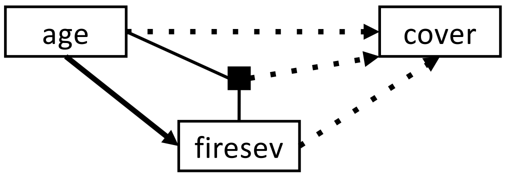

```
partialMedModel_int<-' firesev ~ age

      cover ~ firesev + age + firesev:age'
```

Model Also Does Not Fit



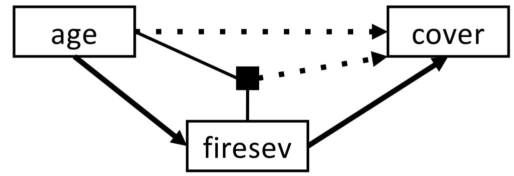

```
lavaan (0.5-23.1097) converged normally after  24 iterations

  Number of observations                            90

  Estimator                                         ML
  Minimum Function Test Statistic              121.892
  Degrees of freedom                                 1
  P-value (Chi-square)                           0.000
```

Collinearity Shows Up in Modification Indices



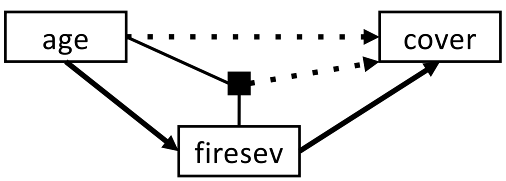

```
> modificationindices(partialMedFit_int)
        lhs     op         rhs     mi     epc sepc.lv sepc.all sepc.nox
...
11 firesev      ~       cover 66.770 -14.162 -14.162   -2.673   -2.673
12 firesev      ~ firesev:age 66.770   0.030   0.030    1.550    0.018
16 firesev:age  ~     firesev 53.016  19.871  19.871    0.380    0.380
```

Same Solution




```
partialMedModel_int_scale<-'
  fire_cent ~ age_cent

  cover ~ fire_cent + age_cent + fire_cent:age_cent
'
```

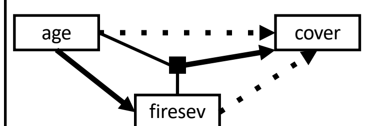
Same Solution



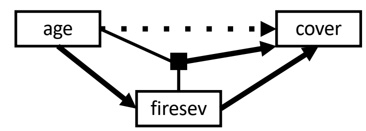

```
lavaan (0.5-23.1097) converged normally after  31 iterations

  Number of observations                            90

  Estimator                                         ML
  Minimum Function Test Statistic                0.121
  Degrees of freedom                                 1
  P-value (Chi-square)                           0.728
```

Final Notes on Centering

Uncentered



Centered



- In uncentered model, additive paths estimate the effect of one variable in the absence of the other.
- In centered model, additive paths estimate the effect of one variable at the average level of the other.
- E.g., fire severity has no effect when age = 0 **versus** fire severity has an effect at the mean level of age

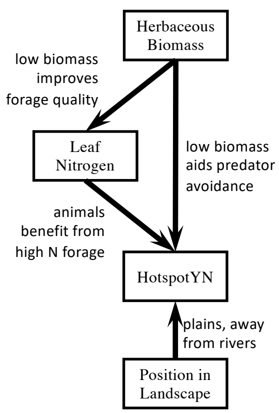
Non-linearities and More

1. Non-linearities in models
   - Polynomial Terms
   - Interactions
2. Generalized Linear Models
3. Special Considerations for GLMs
4. Standardized Nonlinear Coefficients

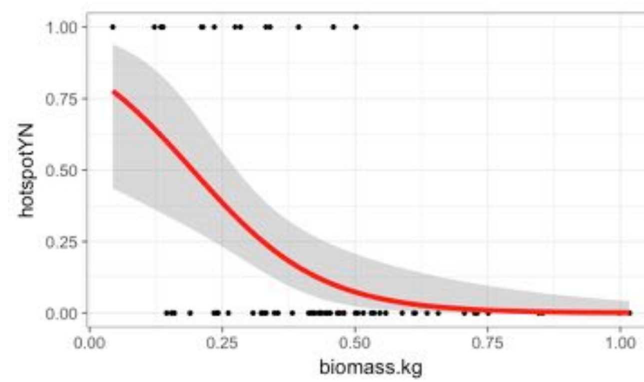
Landscape-scale analyses suggest both nutrient and antipredator advantages to Seregenti herbivore hotspots

133 sites surveyed from 2005-2007 & classified into 'hotspots' (grazers present 80% of time, grazing evident, dung present)

8.4 piecewiseSEM. GLM Example



8.4 piecewiseSEM. GLM Example

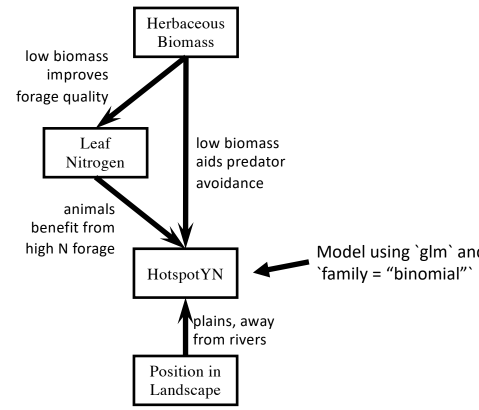


GLM. Components

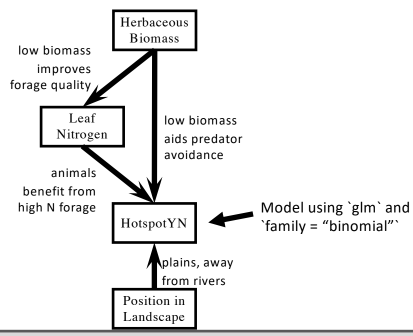
1. A likelihood: $y_i \sim \text{dbin}(\text{prob} = \mu_i, \text{size} = 1)$
   - Could also be poisson, Gamma, negative binomial, etc.
2. A link function: $g(\mu_i) = \eta_i$
   - Logit for binomial
   - Could be log, identity, inverse, etc. for others
3. A linear predictor: $\eta_i = \Sigma \beta X_i$

Other distributions possible – we are exploring other packages

8.4 piecewiseSEM. GLM Example

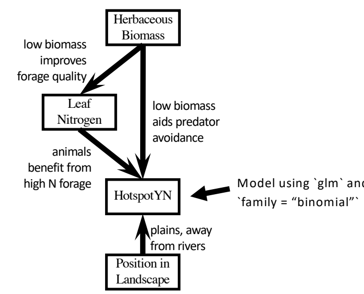


8.4 piecewiseSEM. GLM Example




```
# Load data and fit this model!
anderson <- read.csv("../data/anderson.csv")
```

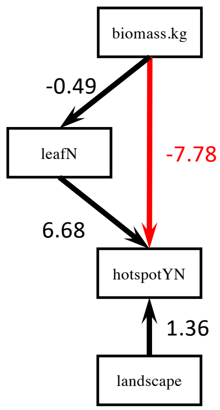
ACTIVITY. Anderson results




```
anderson.sem <- psem(

  lm(leafN ~ biomass.kg, anderson),

  glm(hotspotYN ~ leafN + biomass.kg + landscape,
                family = "binomial", data = anderson),

  data = anderson

)
```

ACTIVITY. Anderson results from coefs

UNSTANDARDIZED

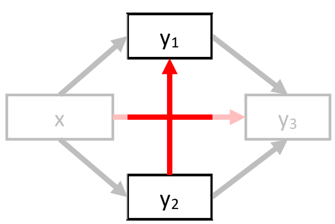


Non-linearities and More

1. Non-linearities in models
2. Generalized Linear Models
3. Special Considerations for GLMs and Dsep
4. Standardized Nonlinear Coefficients

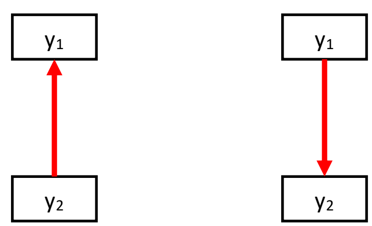
8.5 Directed Separation. A warning

- Intermediate non-normal endogenous variables pose a challenge



8.5 Directed Separation. A warning

- If normal, significance values are reciprocal

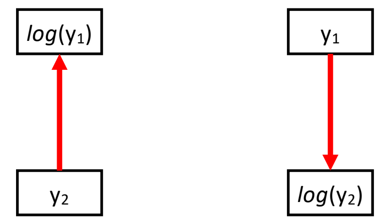


8.5 Directed Separation. A warning

```
# 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] 0.5152512

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

[1] 0.5152512
```

8.5 Directed Separation. A warning

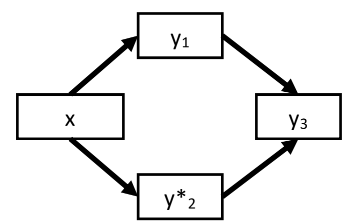
- If non-normal, significance values are *not* reciprocal because of transformation via link function



8.5 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] 0.5152512

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

[1] 0.4792175
```

8.5 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
)
```

8.5 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.
```

8.5 Directed Separation. Options

1. Specify directionality
   - You know a priori which direction the claim should flow
2. Remove path from the basis set
   - Eh……..
3. Calculate both p-value and choose the *lower* one to be conservative
   - Often the most honest choice
   - piecewiseSEM's default

8.5 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 + ... -0.2006020 0.1088444 96 -1.8430170 0.06841223
2 y2  ~  y1 + ... -0.2826168 0.3994236 97 -0.7075617 0.47921747

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

[1] 0.5152512

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

[1] 0.4792175
```

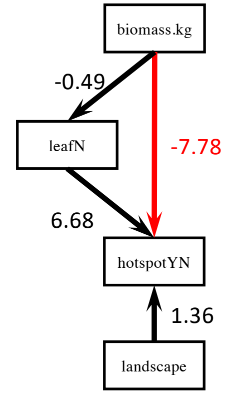
8.5 Directed Separation. A warning

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

 Independ.Claim    Estimate  Std.Error DF Crit.Value     P.Value
1  y3  ~  x + ... -0.20060205 0.10884438 96 -1.8430170 0.06841223
2 y1  ~  y2 + ... -0.01818537 0.02784565 97 -0.6530776 0.51525122

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

 Independ.Claim   Estimate Std.Error DF Crit.Value    P.Value
1  y3  ~  x + ... -0.2006020 0.1088444 96 -1.8430170 0.06841223
2 y2  ~  y1 + ... -0.2826168 0.3994236 97 -0.7075617 0.47921747
```

8.5 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 + ... -0.200602 0.1088444 96  -1.843017 0.06841223
```

ACTIVITY. Anderson results




```
> fisherC(anderson.sem, conserve=TRUE)
  Fisher.C df P.Value
1    2.491  2   0.288
```

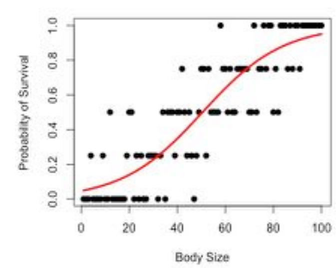
Non-linearities and More

1. Non-linearities in models
   - Polynomial Terms
   - Interactions
2. Generalized Linear Models
3. Special Considerations for GLMs and Dsep
4. Standardized Nonlinear Coefficients

GLM. The problem

PROBLEM: the relationship between *y* and *x* is non-linear; the estimated Betas of the effect on η are on a linear scale

The standard deviation of *y* is different than the standard deviation of η. *Must use sd of η to standardize Beta…how do we get it?*



Grace et al. In Review

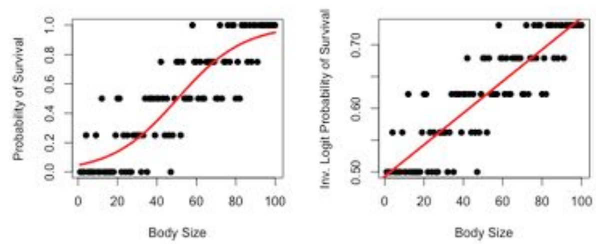
GLM Review

Expectation (some distribution is applied for likelihood):
E(**y**) = μ

Link Function (is the inverse)
g(μ) = **η**

x -> η-> μ -> y

Linear Relationship at the core
η = x**β**



A Latent Theoretic Approach

- Assume that y is related to η via some unobserved variable y*

x -> **η**-> **y*** -> y

- When **y* > some value, y=1, else y = 0**
- Variance in y* = variance in **η** + theoretical variance of the error distribution (sd_e) – every error distribution has one!
  - E.g. 1 for probit link, $\pi^2/3$ for logit

$$\beta_j = b_j \, sd_x / (sd_\eta + sd_\varepsilon)$$

Observed Variance Approach

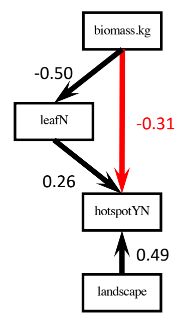
Thinking about R^2 (more on that later):

$R^2 = \sigma^2_{E(y)}/\sigma^2_y$

So, $sd_y = sd_{E(y)}/R$

$$\beta_j = b_j \, sd_x \; / \; (sd_{E(y)}/R)$$

Anderson results with OE formulation




```
> coefs(anderson.sem, standardize.type = "Menard.OE")
   Response  Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate
1     leafN biomass.kg  -0.4880    0.1050 65    -4.6486  0.0000      -0.4995 ***
2 hotspotYN      leafN   6.6867    2.7818 63     2.4037  0.0162       0.2638   *
3 hotspotYN biomass.kg  -7.7838    3.5694 63    -2.1807  0.0292      -0.3143   *
4 hotspotYN  landscape   1.3600    0.4955 63     2.7449  0.0061       0.4913  **
```

Latent Linear versus OE

Other links and error distributions coming soon…

`standardize.type = "latent.linear"`   `standardize.type = "Menard.OE"`