

How Can I Predict New Richness Values?

The figure shows a path diagram identical to the one in the first slide, but with the path coefficients removed. Below the diagram is R code for using piecewise SEM to predict richness values.

```
library(piecewiseSEM)
source("./predict_piecewiseSEM.R")

modList <- list(
  lm(rich ~ distance + abiotic + hetero, data=keeley),
  lm(abiotic ~ distance, data=keeley),
  lm(hetero ~ distance, data=keeley)
)
```

Model Predicted Values from Fits

The figure shows the same path diagram as above. Below it is the R code to extract predicted values from the fitted model.

```
> head(fitted_piecewiseSEM(modList))
  abiotic  hetero  rich
1 50.90807 0.7021004 60.35224
2 44.36228 0.6284415 30.66832
3 51.02268 0.7033901 58.38871
4 51.02268 0.7033901 58.57802
5 50.32866 0.6955804 45.03430
6 50.32866 0.6955804 45.03546
```

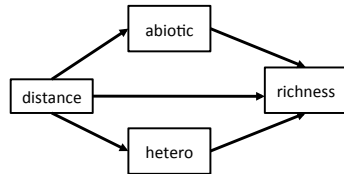
New Predicted Values

The figure shows the same path diagram as above. Below it is the R code to predict new richness values for specific distance values.

```
> predict_piecewiseSEM(modList,
  newdata=data.frame(distance=c(30, 50)))

  abiotic  hetero  rich
1 41.54852 0.5967785 29.99998
2 49.54506 0.6867626 49.99870
```

Or with Model Specific CIs



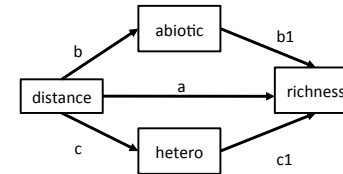
```
> predict_piecewiseSEM(modList, newdata=data.frame(distance=c(30, 50)),
interval="confidence")
```

```

abiotic_fit abiotic_lwr abiotic_upr hetero_fit.1 hetero_lwr.1 ...
1 41.54852 38.08894 45.00810 0.5967785 0.5421282
2 49.54506 48.10297 50.98715 0.6867626 0.6639822
  
```

But... Error is Not Propogating Through the Model

In Lavaan We Can Calculate Compound Coefficients Allowing Error to Propagate



$$\text{Richness} \sim a \cdot \text{distance} + b1 \cdot b \cdot \text{distance} + c1 \cdot c \cdot \text{distance}$$

But what about intercepts?

$$\text{Intercept} = ia + ib \cdot b1 + ic \cdot c1$$

Estimating Direct and Indirect Effects with Named Coefficients

```

totDistModel <- '
rich ~ a*distance + b*abiotic + c*hetero
abiotic ~ b1*distance
hetero ~ c1*distance
  
```

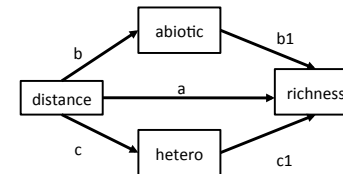
```

rich ~ ia*1
abiotic ~ ib*1
hetero ~ ic*1
  
```

```

direct:= a
indirect:= b1*b + c1*c
total:= b1*b + c1*c + a
totalIntercept := il + ib*b1 + ic*c1
  
```

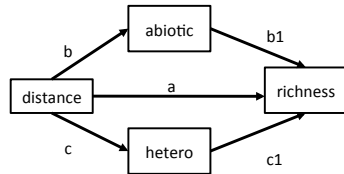
Estimates of Compounded Effects



Defined Parameters:

| | Estimate | Std.Err | Z-value | P(> z) |
|----------------|----------|---------|---------|---------|
| direct | 0.640 | 0.156 | 4.117 | 0.000 |
| indirect | 0.360 | 0.103 | 3.488 | 0.000 |
| total | 1.000 | 0.145 | 6.918 | 0.000 |
| totalIntercept | -19.070 | 9.683 | -1.969 | 0.049 |

Predictions with those Coefficients!



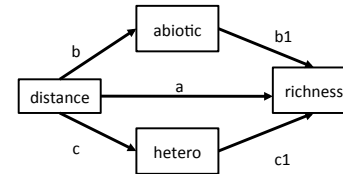
```

> #Get new predictions
> newDistance <- 57

> #naive prediction
> newDistance*1 + -19.070

[1] 37.93
  
```

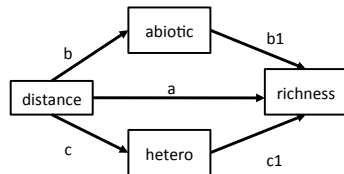
You have Compound Parameter Uncertainty



Defined Parameters:

| | Estimate | Std.Err | Z-value | P(> z) |
|----------------|----------|---------|---------|---------|
| direct | 0.640 | 0.156 | 4.117 | 0.000 |
| indirect | 0.360 | 0.103 | 3.488 | 0.000 |
| total | 1.000 | 0.145 | 6.918 | 0.000 |
| totalIntercept | -19.070 | 9.683 | -1.969 | 0.049 |

You have Compound Parameter Uncertainty

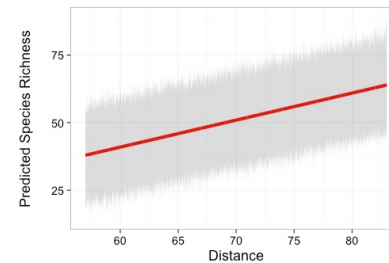


```

> predDist <- rnorm(100, 1, 0.145)*newDistance +
  rnorm(100, -19.070, 9.683)

> quantile(predDist)
  0%    25%   50%   75%   100%
10.58009 31.15239 39.55937 46.08567 66.19564
  
```

Creating Ranges of Predictions

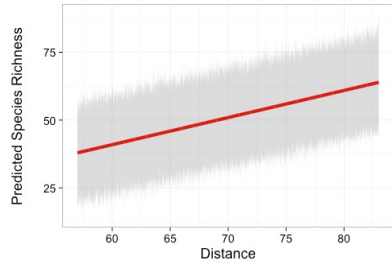


```

new_data <- data.frame(distance = seq(57,83,.01))
new_data$predicted_richness <- new_data$distance*1 + -19.070

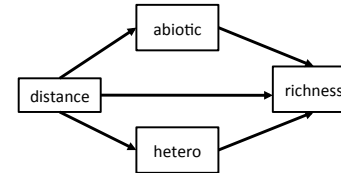
new_data <- cbind(new_data,
  t( sapply(new_data[,1], function(x)
    quantile( rnorm(100, 1, 0.145)*x +
      rnorm(100, -19.070, 9.683),
      probs=c(0.025, 0.975)))
  ))
  
```

Creating Ranges of Predictions



But... UGH – that’s a lot of heavy lifting, and the variance is...not so smooth.

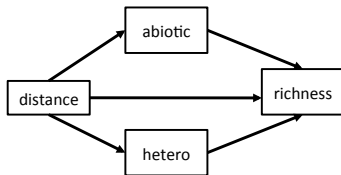
Lavaan Also Allows Easier Propagation of Error Due to Full Parameter Vcov Matrix



```

> vcov(totDistFit)
           rch-ds rch-bt rch-ht abtc-d htr-ds rch--r abtc-- htr--h
rich-distance  0.024
rich-abiotic   -0.012  0.029
rich-hetero    -0.521  0.000 115.778
abiotic-distance 0.000  0.000  0.000  0.007
hetero-distance  0.000  0.000  0.000  0.000  0.000
rich--rich      0.000  0.000  0.000  0.000  0.000 317.772
abiotic--abiotic 0.000  0.000  0.000  0.000  0.000  0.000 47.000
hetero--hetero  0.000  0.000  0.000  0.000  0.000  0.000  0.000 0.000
  
```

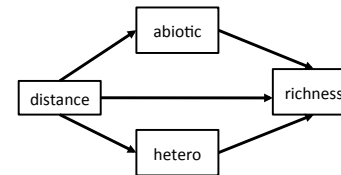
New Functions Order and Fit Relationships



```

> source("./fitted_lavaan.R")
> new_data <- data.frame(distance = seq(57,83,.01))
> pred_df <- predict_lavaan(totDistFit, newdata = new_data)
> head(pred_df )
  abiotic hetero rich
1 52.34385 0.7182571 56.99825
2 52.34785 0.7183021 57.00825
3 52.35184 0.7183471 57.01825
4 52.35584 0.7183920 57.02825
  
```

Error Propagation Via Simulation



```

> pred_df_ci <- predict_lavaan(totDistFit, newdata = new_data, simCI=TRUE,
                               nsims=100, ci=0.95)
> head(pred_df_ci)
  abiotic hetero rich abiotic_lower hetero_lower rich_lower abiotic_upper
1 52.34385 0.7182571 56.99825 50.73199 0.6929244 54.06795 54.40702
2 52.34785 0.7183021 57.00825 50.73473 0.6929432 54.07652 54.41241
3 52.35184 0.7183471 57.01825 50.73746 0.6929619 54.08510 54.41781
4 52.35584 0.7183920 57.02825 50.74019 0.6929807 54.09368 54.42320
5 52.35984 0.7184370 57.03825 50.74293 0.6929995 54.10225 54.42859
6 52.36384 0.7184820 57.04825 50.74566 0.6930183 54.11083 54.43399
  
```

