Coefficients Revisited

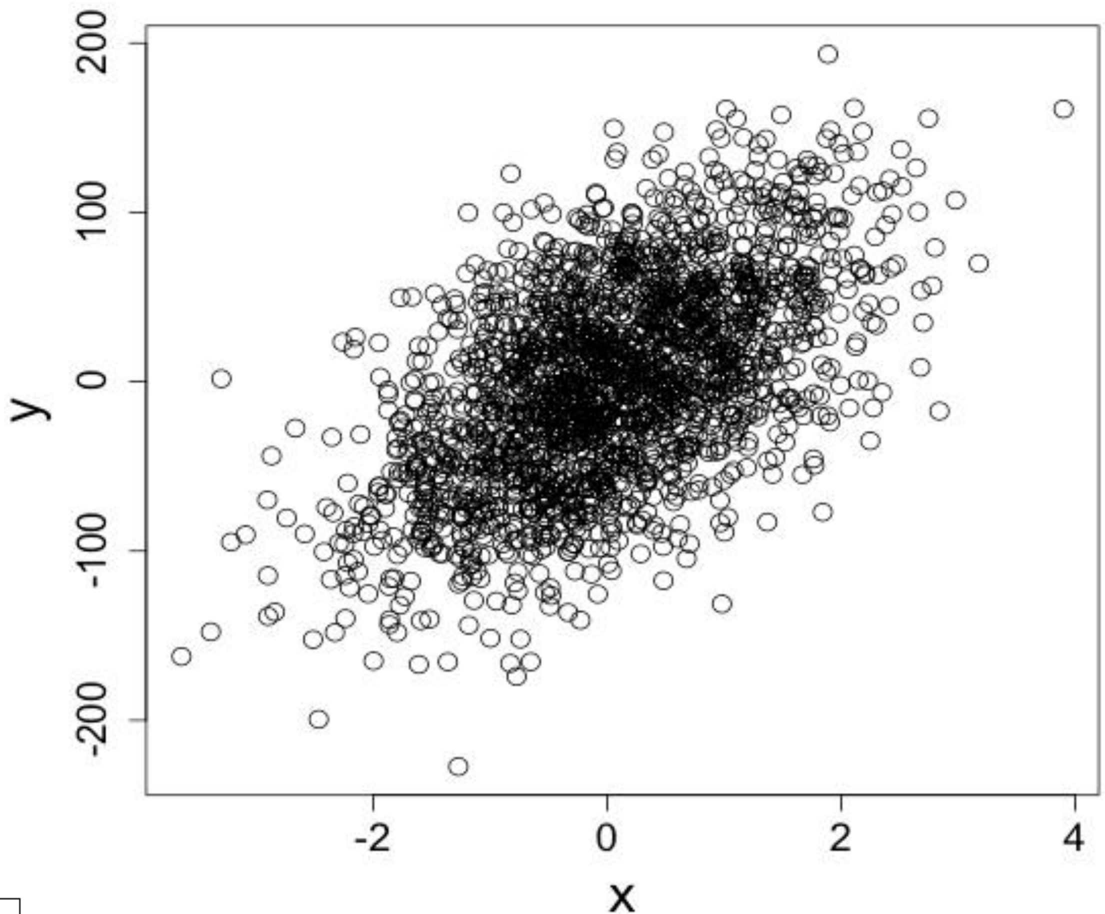
Overview

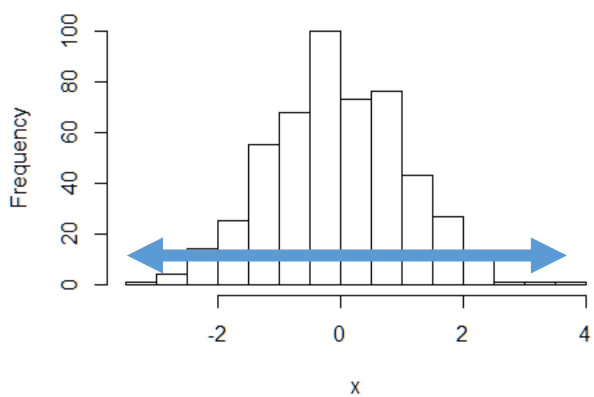
1. Rules of Coefficients
2. Range Standardization
3. GLM (Logistic Regression)
4. GLM (Poisson Regression)

1.1 Coefficients. Covariance and correlation

$$VAR_y = \frac{\sum(y_i - \bar{y})^2}{n-1}$$

$$SD_y = \sqrt{VAR_y}$$



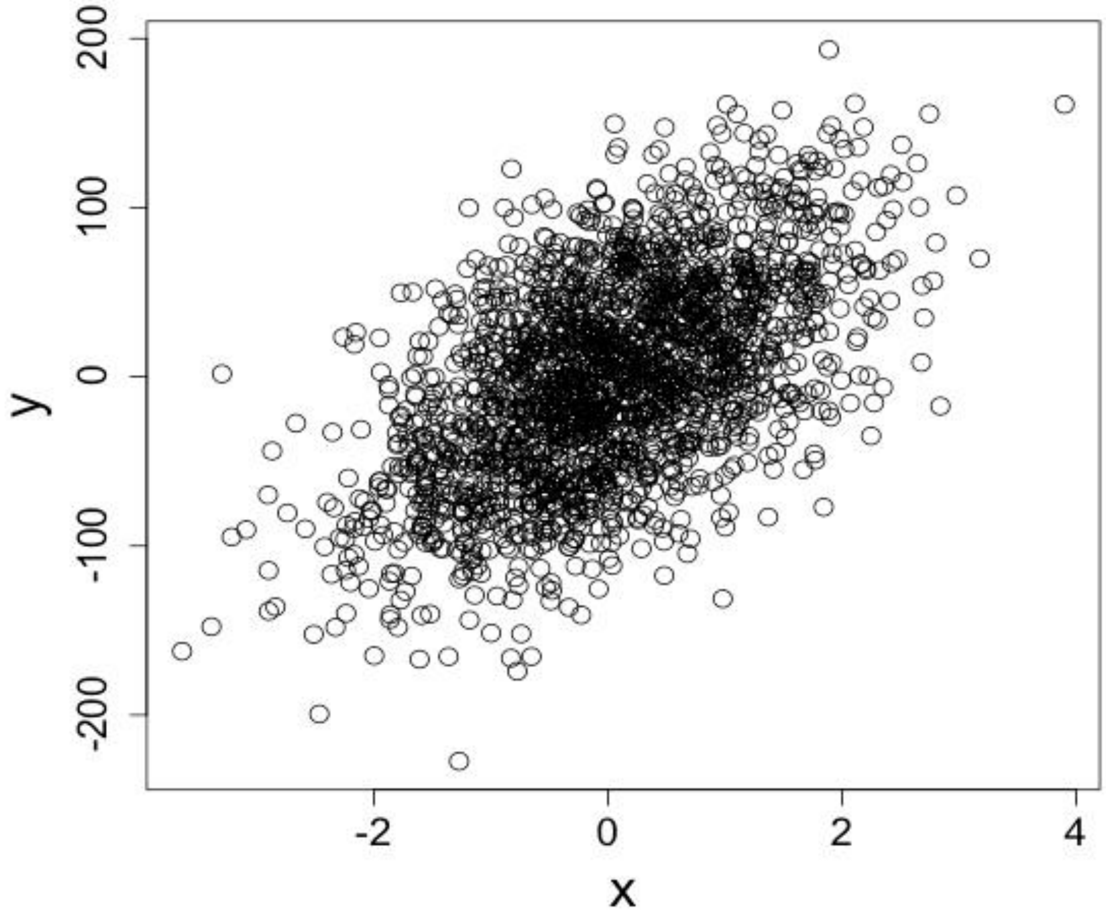


$$VAR_x = \frac{\sum(x_i - \bar{x})^2}{n-1}$$

1.1 Coefficients. Covariance and correlation

$$COV_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

$$r_{xy} = \frac{COV_{xy}}{(SD_x \times SD_y)}$$



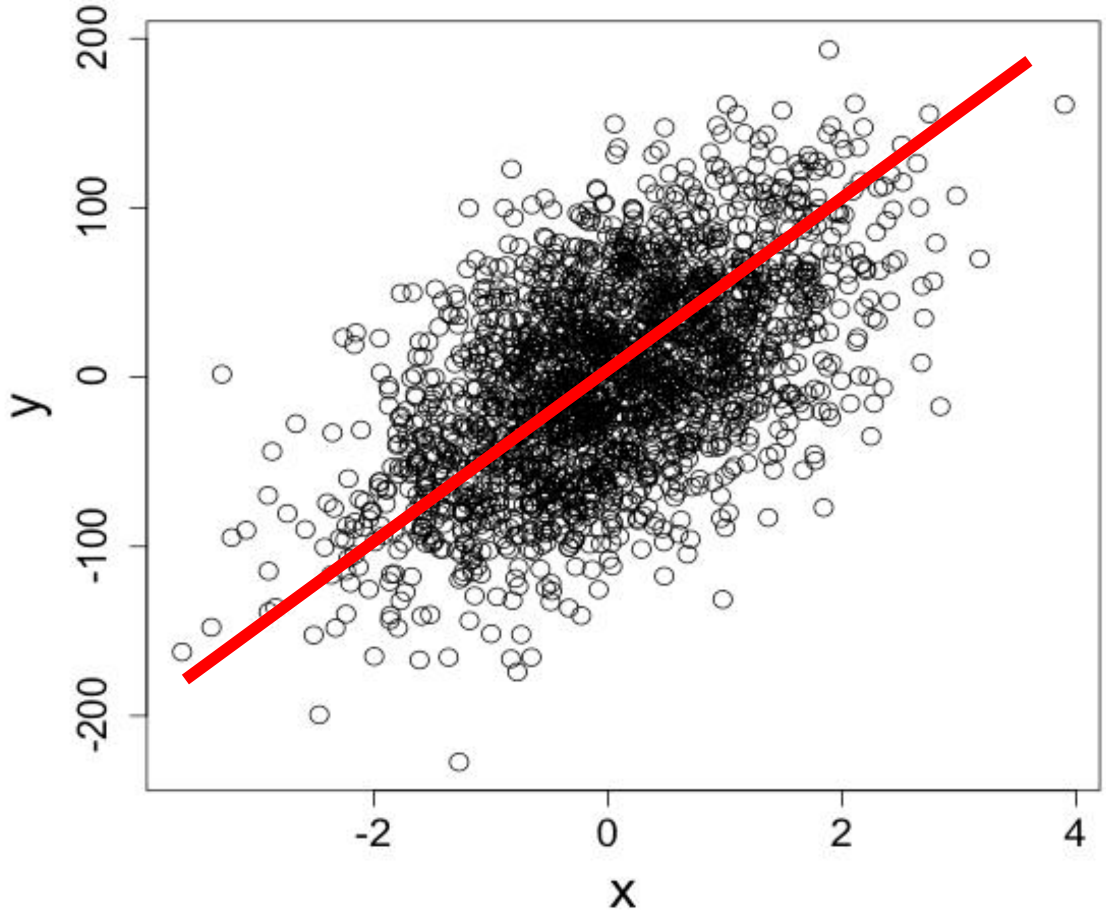
Covariances *are* correlations when variables are standardized
(Z-transformed: subtract the mean and divide by the SD)

1.1 Coefficients. Covariance and correlation

“rise”

$$\beta_{xy} = \frac{COV_{xy}}{VAR_x}$$

“run”



- *Unstandardized coefficient* = absolute strength of the pathway
  - “ An 1 unit change in *X* results in some unit change in *Y* ”

1.1 Coefficients. Standardization

- *Standardized coefficient* = relative strength of the pathway (correlation)
  - " A 1 standard deviation change in *X* results in some standard deviation change in *Y* "

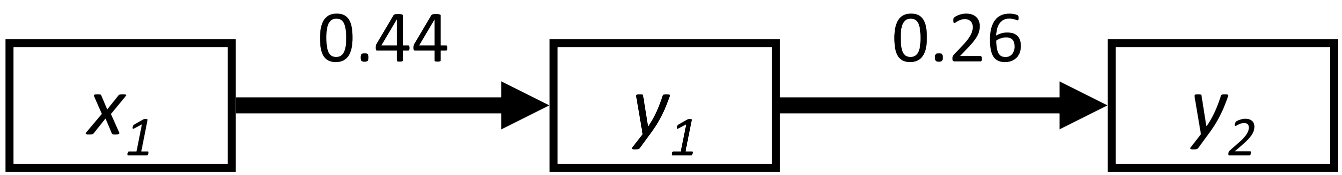
$$b_{xy} = \beta_{xy} * \frac{SD_x}{SD_y} = \frac{COV_{xy}}{(SD_x \times SD_y)} * \frac{SD_x}{SD_y} = \frac{COV_{xy}}{SD_y} = r_{xy}$$

1.1 Coefficients. Standardization

Unstandardized	Standardized
Good for prediction: coefficients are in raw units	Good for ranking: coefficients are in equivalent units
Has direct real world meaning	Less clear real world meaning
Can be compared across pathways or models that have identical units	Can be compared across all pathways in the same model and across model when population variances are not different (otherwise scaling is not equivalent)

1.1 Coefficients. Rule #3 of path coefficients

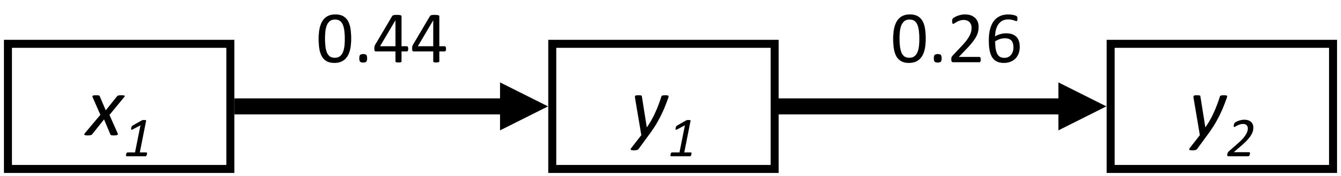
Third Rule of Path Coefficients: strength of a compound path is the product of the (standardized) coefficients along the path.



If the indirect path from x_1 to y_2 equals the correlation between x_1 and y_2, we say x_1 and y_2 are *conditionally independent*.

1.1 Coefficients. Rule #3 of path coefficients

What does it mean when two separated variables are *not* conditionally independent?

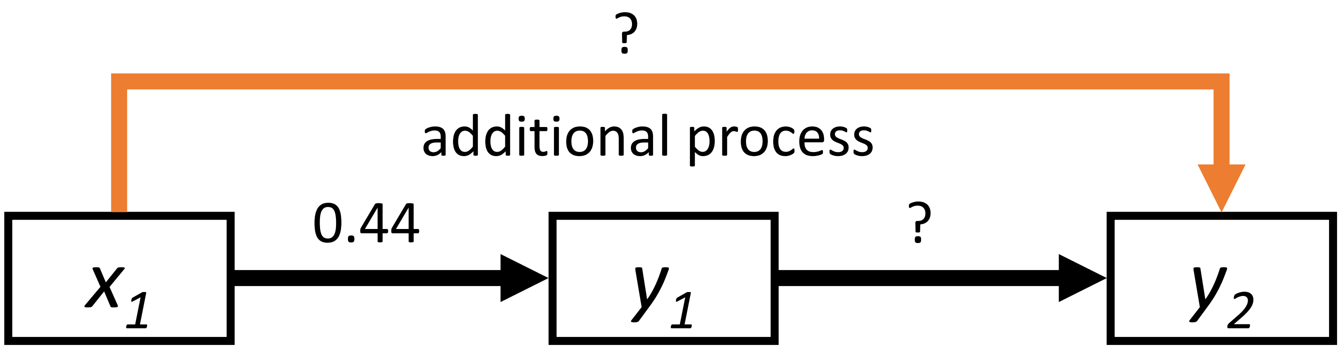


0.44 * 0.26 = 0.11, which is not equal to $r_{x,y2}$ = 0.31

	x_1	y_1	y_2
x_1	1.0		
y_1	0.44	1.0	
y_2	0.31	0.26	1.0

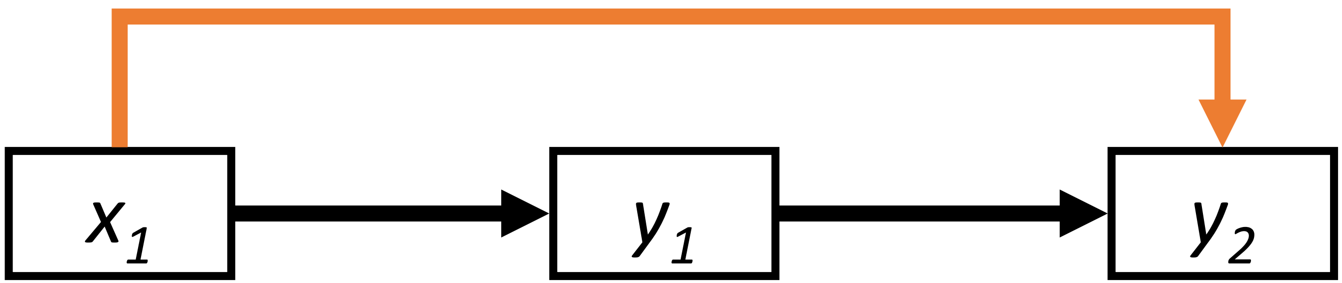
1.1 Coefficients. Rule #4 of path coefficients

The inequality implies that the true model is:



Fourth Rule of Path Coefficients: when variables are connected by more than one causal pathway, the path coefficients are "partial" regression coefficients.

1.1 Coefficients. What is a partial coefficient?



Direct correlation

Indirect correlations

$$\gamma_{21} = \frac{r_{x_1 y_2} - (r_{x_1 y_1} \times r_{y_1 y_2})}{1 - r^2_{x_1 y_1}}$$

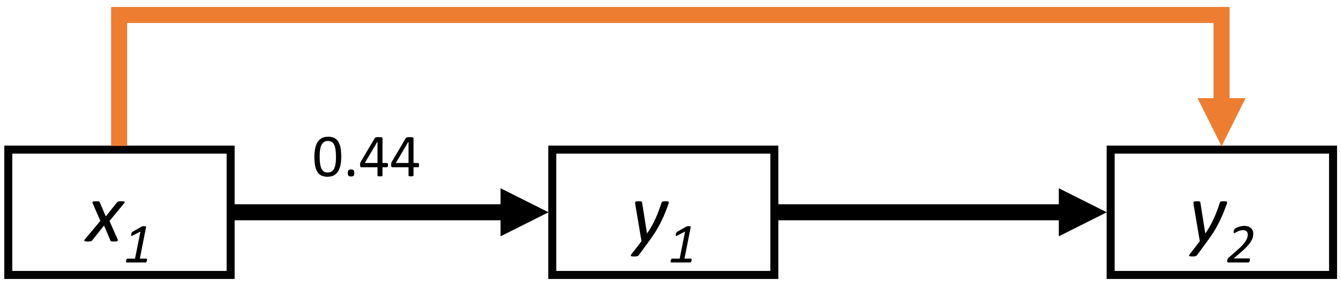
Shared variance between predictors

1.1 Coefficients. What is a partial coefficient?

$$\gamma_{21} = \frac{r_{x_1 y_2} - (r_{x_1 y_1} \times r_{y_1 y_2})}{1 - r^2_{x_1 y_1}}$$

	x_1	y_1	y_2
x_1	1.0		
y_1	0.44	1.0	
y_2	0.31	0.26	1.0

$$\beta_{12} = \frac{(0.31 - (0.26 * 0.44))}{1 - 0.44^2} = 0.25$$



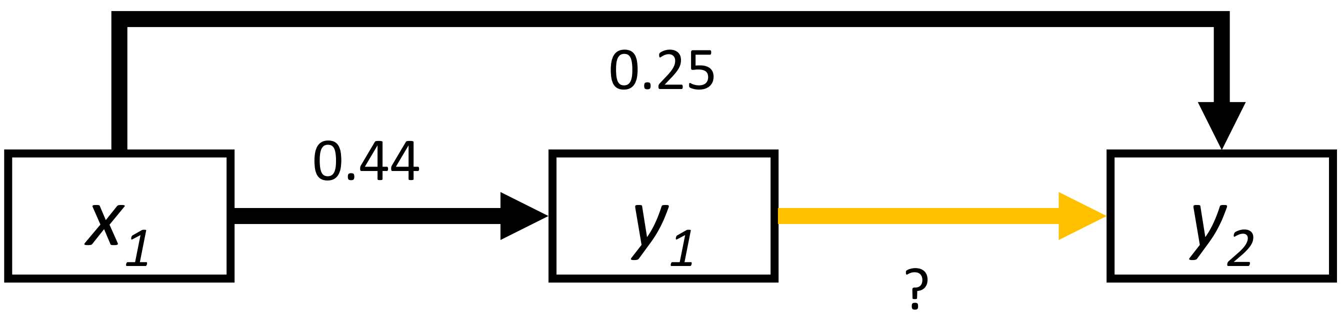
1.1 Coefficients. What is a partial coefficient?

Direct correlation — Indirect correlations

$$\beta_{21} = \frac{r_{y_1y_2} - (r_{x_1y_1} \times r_{x_1y_2})}{1 - r^2_{x_1y_1}}$$

Unshared variance between Predictors (denominator)

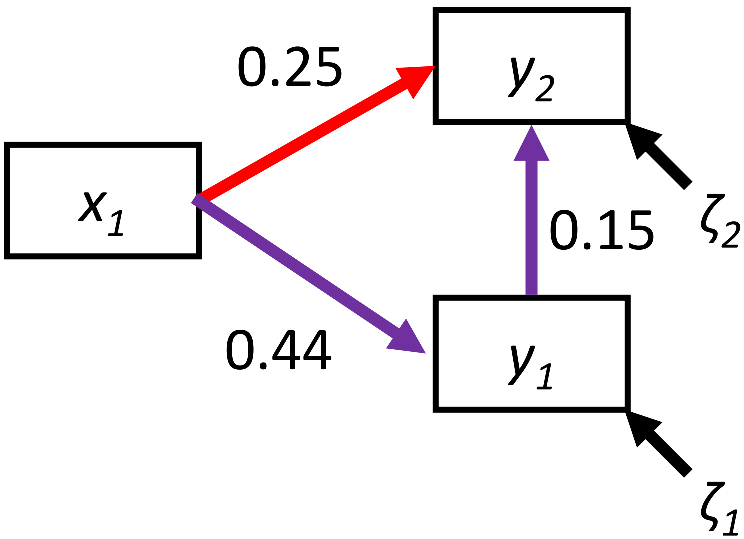
	x_1	y_1	y_2
x_1	1.0		
y_1	0.44	1.0	
y_2	0.31	0.26	1.0



$$\gamma_{21} = \frac{(0.26 - (0.44 * 0.31))}{1 - 0.44^2} = 0.15$$

1.1 Coefficients. Rule #8 of path coefficients

Eighth Rule of Path Coefficients: sum of all pathways between two variables (directed and undirected) equals the correlation.



Total Effects:

	x_1	y_1	y_2
x_1	1.0		
y_1	0.44	1.0	
y_2	0.31	0.26	1.0

0.25 + 0.44 * 0.15 = 0.31

1.2 Range Standardization

1.2. Range Standardization.

- Range standardization puts coefficients in units of range:

$$b = B_{xy} * \frac{(\max(x) - \min(x))}{(\max(y) - \min(y))}$$

- Interpreted as a moving *x* along its range would result in a % change in *y* along its range

- Good for binary or ordinal predictors (“moving *x* from off to on” or “moving *x* from one state to the next”)

1.2. Range Standardization.

- Relevant range standardization can define a custom range for *x* and *y*
  - More meaningful in certain contexts (e.g., "a % reduction in *x* leads to a % reduction in *y*")

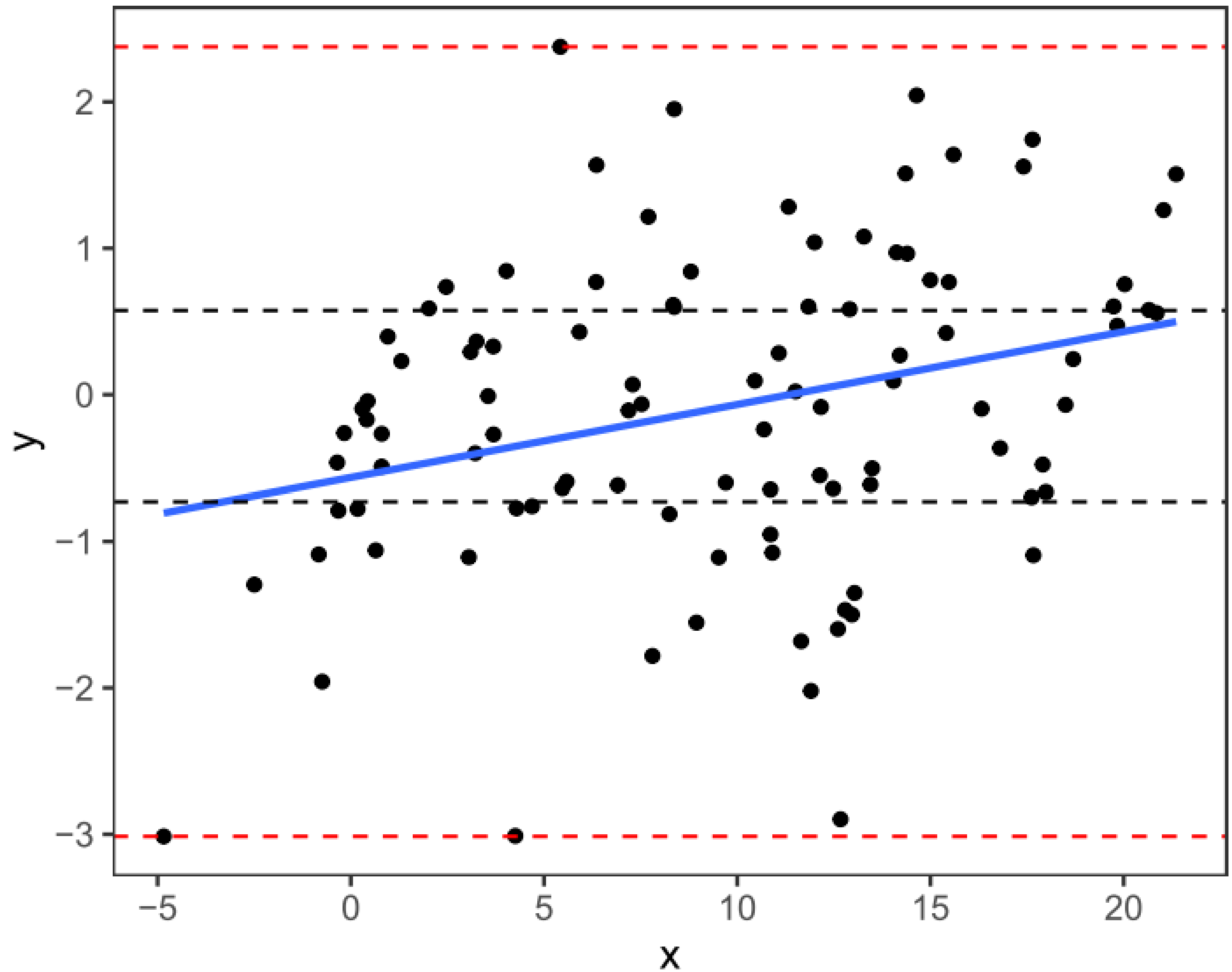
- Good for contextualizing variables with very different variances/distributions where 1 SD may equate to very different proportions of the total range

- Only in *piecewiseSEM*

1.2. Range Standardization.

```
# Generate fake data
set.seed(8)

data <- data.frame(y = rnorm(100))

data$x <- data$y * 2 + runif(100, 0, 20)

# Fit model
model <- lm(y ~ x, data)

piecewiseSEM::coefs(model, standardize = "range")

 Response Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate
1  y         x   0.0498    0.0161 98     3.0861  0.0026        0.242 **

# As you move along the entire range of x, you move along 25% of the
range of y
```

1.2. Range Standardization.



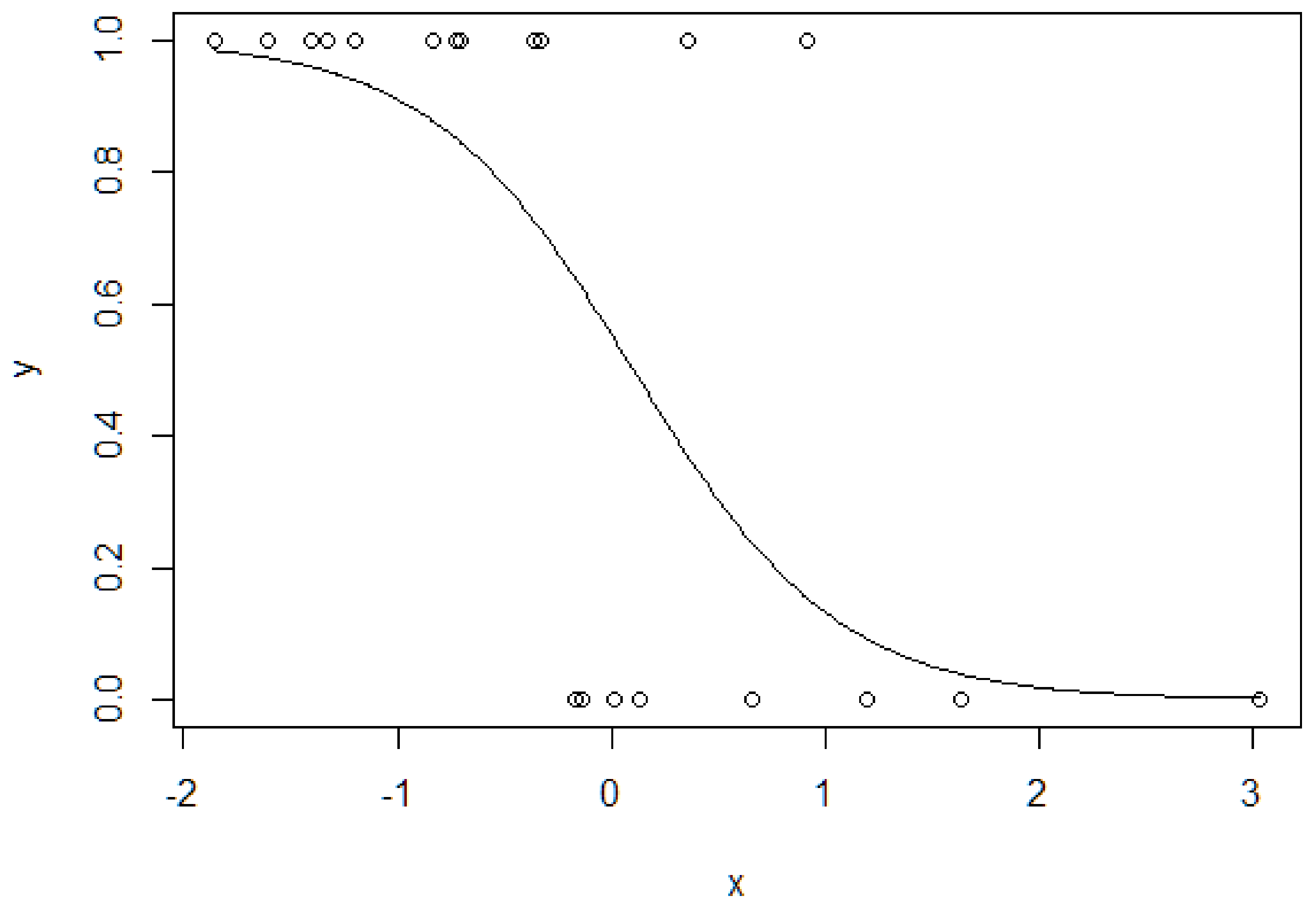
1.2. Range Standardization.

```
# Specify relevant range (20% increase in x)
piecewiseSEM::coefs(model, standardize = list(x = c(min(data$x),
max(data$x)*0.20)))

  Response Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate
1        y         x   0.0498    0.0161 98     3.0861  0.0026       0.0842 **
Warning message:
Relevant range not specified for variable 'y'. Using observed range instead

# a 20% increase in x2 leads to an 8.4% increase in y
```

1.3 GLM (Logistic Regression)

1.3. Logistic Regression. Components

Binary responses are not a linear function of *x*….



1.3. Logistic Regression. Components

1. The random component: specifying conditional distribution of values for the response variable *y*, e.g., *y* ~ *dbin*(μ).

2. Linear predictors: made up of *j* predictor (*x*) variables.

$$\eta = \sum x_j \beta_j$$

3. Link function: *g*(·) that transforms the expectation of the response variable to the linear predictors.

$$\eta_i = g(\mu_i)$$

$$\mu_i \stackrel{\text{def}}{=} E(y_i)$$

1.3. Logistic Regression. Components

1. Logit link

$$logit(\mu_i) = log\left(\frac{\mu_i}{1-\mu_i}\right) = log\left(\frac{P(y=1)}{P(y=0)}\right) = \sum_{j=1}^{p} \beta_j x_{ij}$$

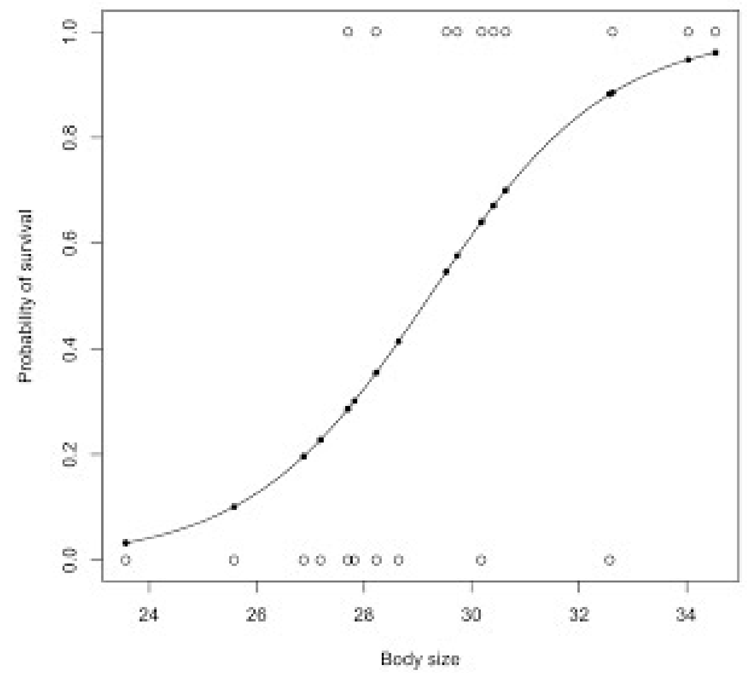
2. Probit link

$$probit(\mu_i) = \Phi^{-1}(\mu_i) = \sum_{j=1}^{p} \beta_j x_{ij}$$

PROBLEM: the relationship between *y* and *x* is non-linear = the coefficients are on a link-transformed *y** (linear scale)

So… the standard deviation of *y* is different than the standard deviation of *y*. How do we get sd(y*)??*

$$b = B_{xy*} * \frac{sd(x)}{sd(y^*)}$$

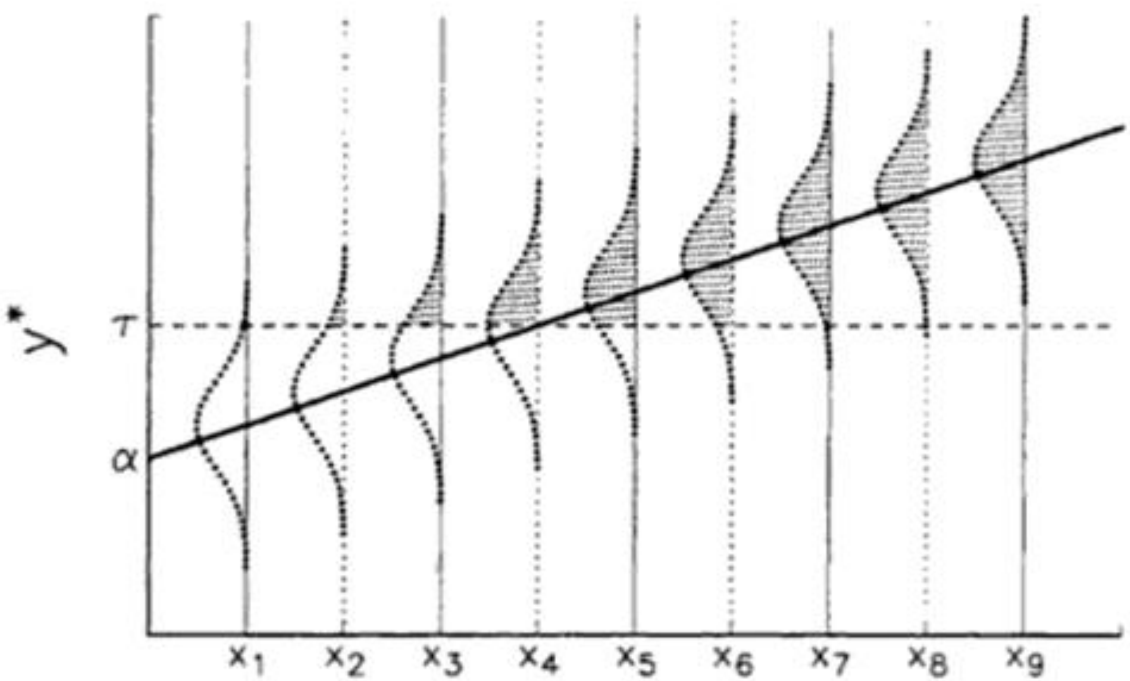


1.3. Logistic Regression. Latent theoretic approach

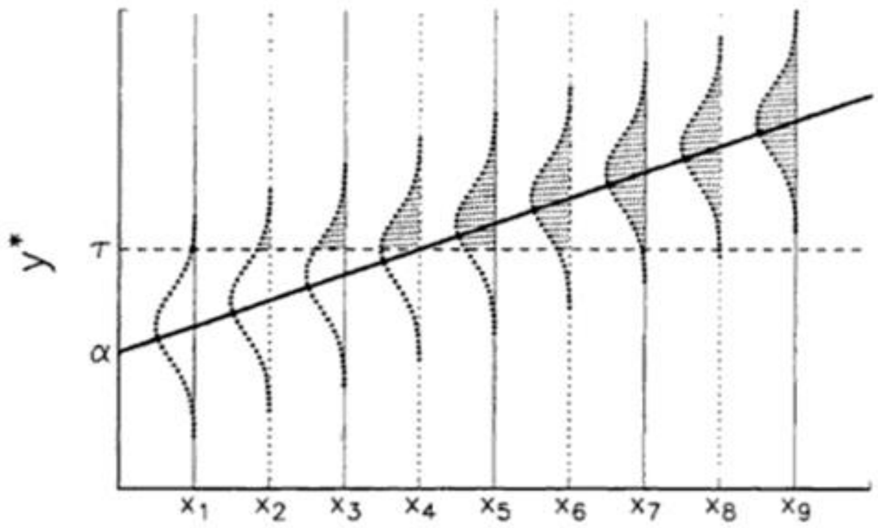
Imagine that every sample has some underlying probability of observing a 0 or a 1

E.g., sampling fish along an estuarine gradient

Arrange probabilities along *x* = linear change in mean probability



1.3. Logistic Regression. Latent theoretic approach



Some point where probability is 50/50 = cutoff point

In this model, the latent variable *y** is linked to the observed binary values of *y* via the following relationship:

$$y_i = \begin{cases} 1 \;\; if\; y_i^* > \tau \\ 0 \;\; if\, y_i^* < \tau \end{cases}$$

and τ is a cutpoint or threshold (generally 0.5)

1.3. Logistic Regression. Latent theoretic approach

A latent *y** is linearly related to predictors through a linear model

$$y_i^* = \mathbf{x}_i \boldsymbol{\beta}_i + \varepsilon_i.$$

Because *y** is unobserved (latent) we have no idea about its mean or variance

If we assume it follows a certain distribution (e.g., binomial) then we have theoretical error variances available for different link functions:

Logit = Var(ε) = $\pi^2/3$

Probit = Var(ε) = 1

1.3. Logistic Regression. Latent theoretic approach

If we assume those error variances, then the variance of *y**:

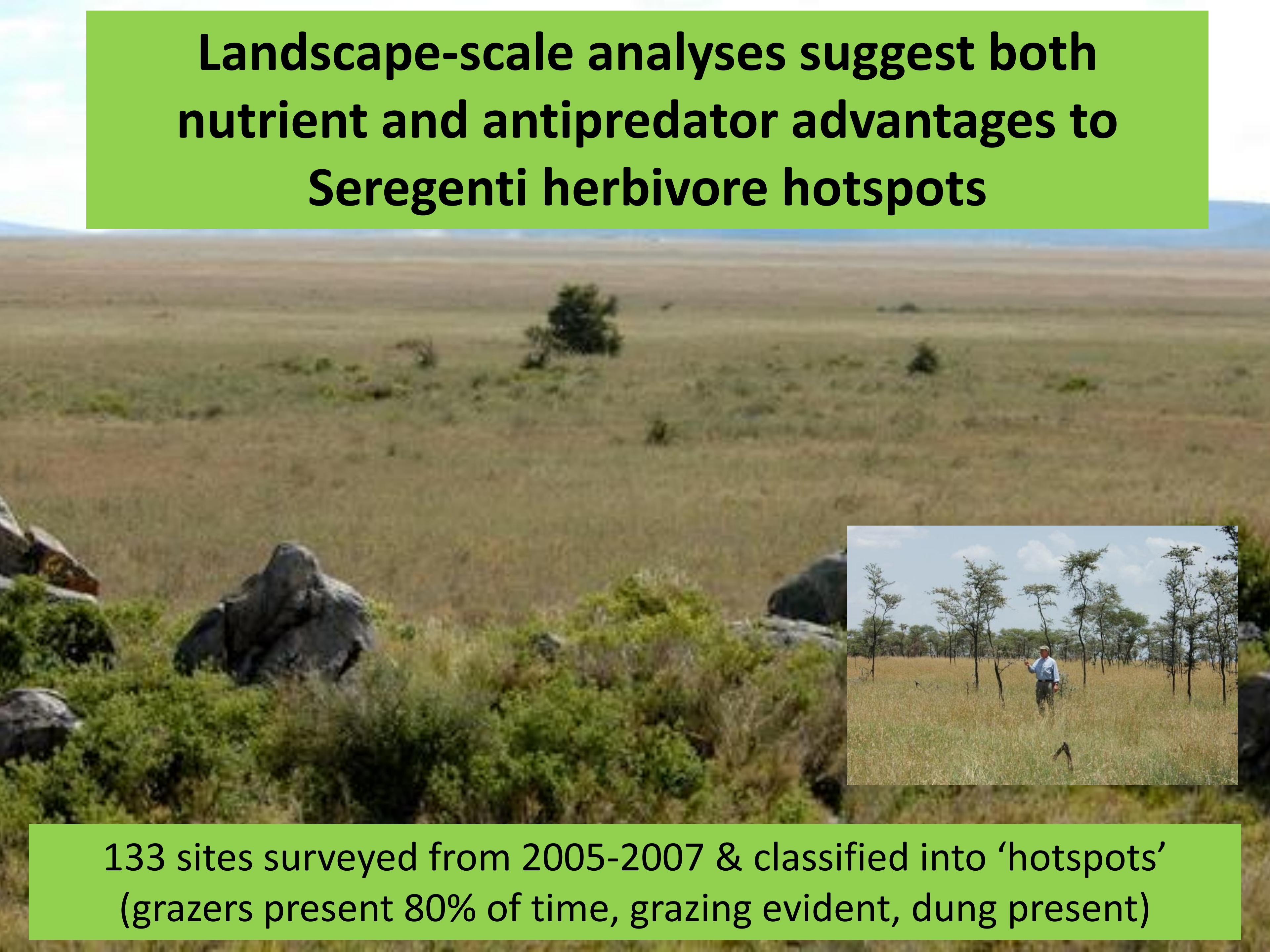
$$\sigma^2_{y^*} = \sigma^2_{\mathrm{x}\beta} + \sigma^2_{\epsilon}$$

Variance of predicted values on the linear (i.e., link) scale

Assumed error variance due to non-linearity

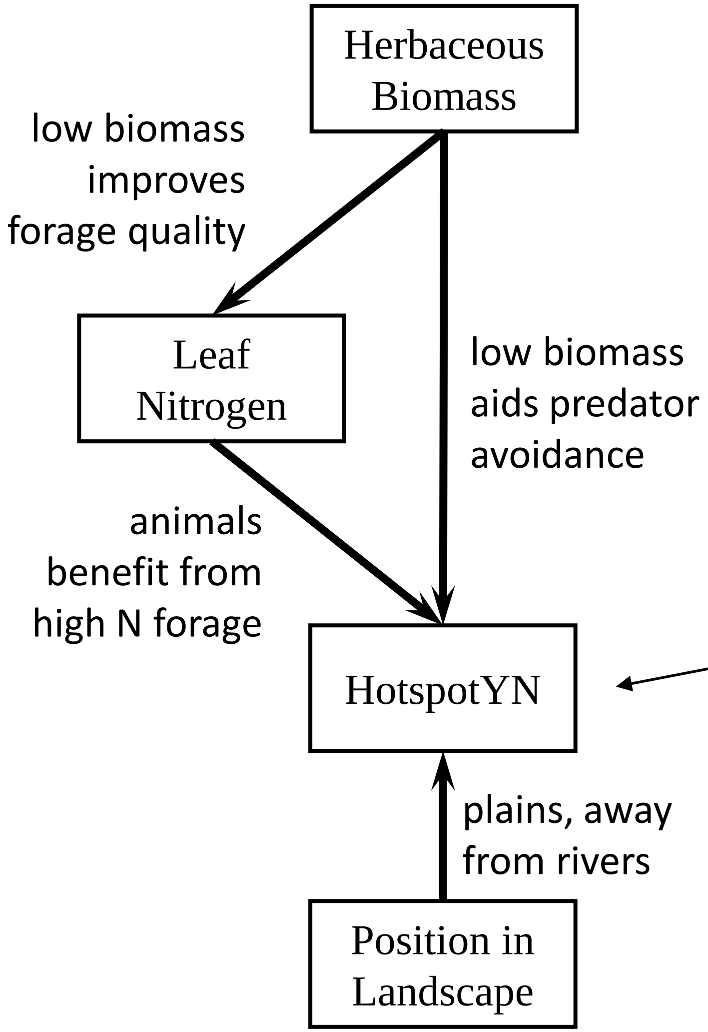
Taking the square-root yields of the sd of *y*, which can be used in standardization

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)

1.3. Logistic Example. Anderson & Grace



Model using `glm` and `family = "binomial"`

1.3. Logistic Example.

```
# read in data
anderson <- read.csv("anderson.csv")

# construct glm
anderson_glm <- glm(hotspotYN ~ leafN + biomass.kg + landscape,
"binomial", anderson)

summary(anderson_glm)


.......

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -12.4026     4.8352  -2.565  0.01031 *
leafN         6.6867     2.7818   2.404  0.01623 *
biomass.kg   -7.7838     3.5694  -2.181  0.02921 *
landscape     1.3600     0.4955   2.745  0.00605 **
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

1.3. Logistic Example.

```
# get fitted values of linear y*
preds <- predict(anderson_glm, type = "link") # linear predictions

# latent theoretic
sd.ystar <- sqrt(var(preds) + (pi^2)/3) # for default logit-link

# get coefficients from GLM output
betas <- summary(anderson_glm)$coefficients[2:4, 1]

# get vector of sd's of x's
sd.x <- apply(anderson[, names(betas)], 2, sd)
```

1.3. Logistic Example.

```
# conduct SEM
anderson_sem <- psem(
  glm(hotspotYN ~ leafN + biomass.kg + landscape, "binomial",
anderson),
  lm(leafN ~ biomass.kg, anderson),
  data = anderson
)

# get summary output
summary(anderson_sem)
```

```
Structural Equation Model of anderson_sem

Call:
  hotspotYN ~ leafN + biomass.kg + landscape
  leafN ~ biomass.kg

    AIC
 4.617

---
```

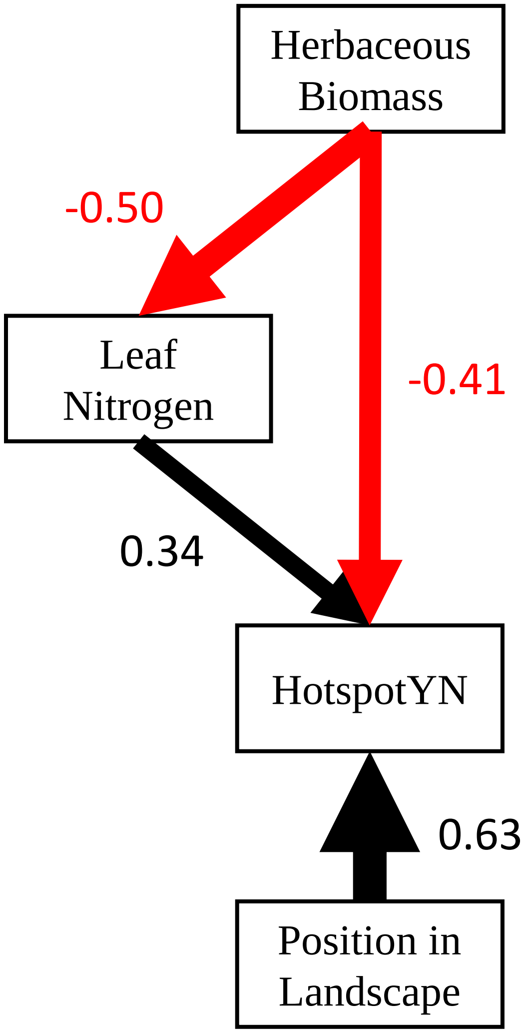
1.3. Logistic Example.

```
Tests of directed separation:

           Independ.Claim Test.Type DF Crit.Value P.Value
  leafN ~ landscape + ...      coef 64    -1.0718  0.2878

--
Global goodness-of-fit:

Chi-Squared = 1.192 with P-value = 0.275 and on 1 degrees of freedom
Fisher's C = 2.491 with P-value = 0.288 and on 2 degrees of freedom
```

1.3. Logistic Example.

```
Coefficients:

   Response  Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate
  hotspotYN      leafN   6.6867    2.7818 63     2.4037  0.0162       0.3399   *
  hotspotYN biomass.kg  -7.7838    3.5694 63    -2.1807  0.0292      -0.4050   *
  hotspotYN  landscape   1.3600    0.4955 63     2.7449  0.0061       0.6332  **
      leafN biomass.kg  -0.4880    0.1050 65    -4.6486  0.0000      -0.4995 ***

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

---
Individual R-squared:

   Response     method R.squared
  hotspotYN nagelkerke      0.61
      leafN       none      0.25
```

1.3. Logistic Example.



Indirect effect:
-0.50 * 0.34 = -0.17

- As herbaceous biomass goes up, it reduces the chances of being a hotspot (reduced visibility)
- It also dilutes forage quality, further reducing the chances of being a hotspot
- The direct effect is ~2x that of the indirect effect

1.3. Logistic Regression. OE Approach

If non-linear *y* is truly discrete... (aka, not latent continuous but actually binary or continuous, such as counts)

For GLM models we can compute an approximate R^2 as the squared correlation between observed and fitted values (both of which we know)

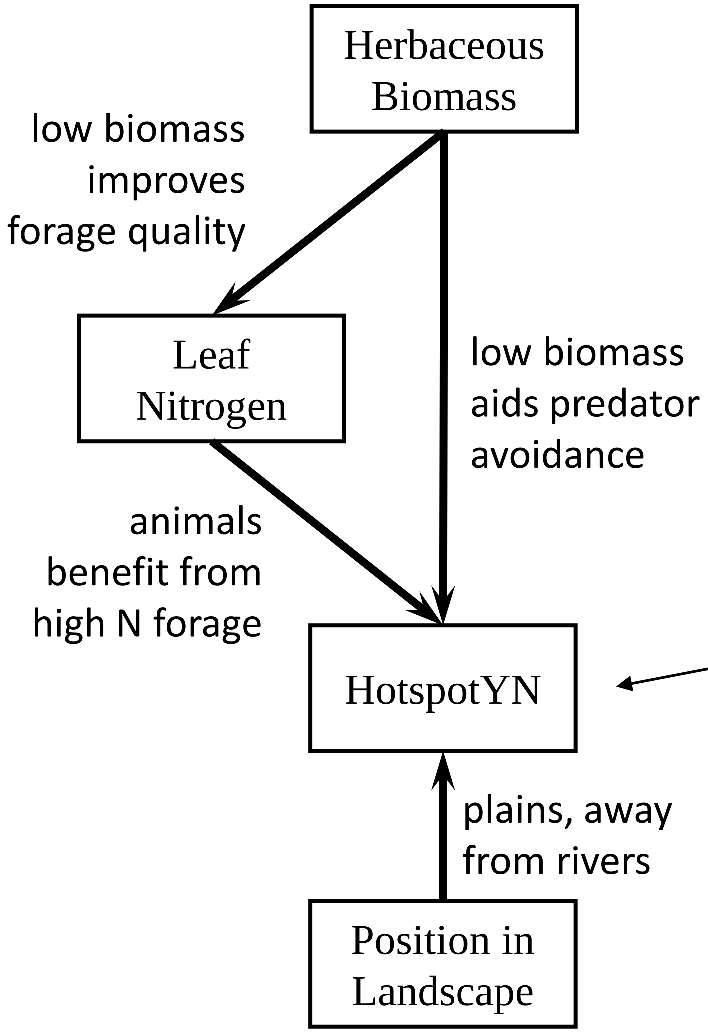
We also know the variance of the fitted (linear) values ($\sigma^2_{\hat{y}}$)

We can use this equation to solve for the variance of the non-linear *y:*

$$R^2 = \frac{\sigma^2_{\hat{y}}}{\sigma^2_y} \text{ such that } \sigma^2_y = \frac{\sigma^2_{\hat{y}}}{R^2}$$

Where we can take the square-root to get the *sd(y)*

1.3. Logistic Example. Anderson & Grace



Model using `glm` and `family = “binomial”`

1.3. Logistic Example.

```
# get sd of fitted values
preds <- predict(anderson_glm, type = "link")

# get sd based on observed variance
R2 <- cor(anderson$hotspotYN, predict(anderson_glm, type =
"response"))^2

# observed empirical sd
sd.yhat <- sqrt(var(preds)/R2)

# get coefficients
betas <- summary(anderson_glm)$coefficients[2:4, 1]

# get vector of sd's of x's
sd.x <- apply(anderson[, names(betas)], 2, sd)

# get OE standardized betas
(OE_betas <- betas * (sd.x/sd.yhat))

 leafN biomass.kg  landscape
 0.2637846 -0.3142830  0.4913118
```

1.3. Logistic Example.

```
# get observation empirical standardization
coefs(anderson_glm, standardize.type = "Menard.OE")

   Response  Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate
1 hotspotYN      leafN   6.6867    2.7818 63     2.4037  0.0162       0.2638  *
2 hotspotYN biomass.kg  -7.7838    3.5694 63    -2.1807  0.0292      -0.3143  *
3 hotspotYN  landscape   1.3600    0.4955 63     2.7449  0.0061       0.4913 **

# compare to latent linear approach
coefs(anderson_glm, standardize.type = "latent.linear") # default

 Response  Predictor Estimate Std.Error DF Crit.Value P.Value Std.Estimate
1 hotspotYN      leafN   6.6867    2.7818 63     2.4037  0.0162       0.3399  *
2 hotspotYN biomass.kg  -7.7838    3.5694 63    -2.1807  0.0292      -0.4050  *
3 hotspotYN  landscape   1.3600    0.4955 63     2.7449  0.0061       0.6332 **

# standardized coefs are smaller (since not incorporating binomial
distribution-specific error variance in the denominator of sd(y))
```

1.3. Logistic Example. Conclusions

- Both forms of standardization allow for fair comparison of effect sizes and calculation of indirect effects
- Is binary response generated by underlying probability? = latent theoretic. If not? = observation empirical
- Observation-empirical approach tends to yield lower standardized coefficients than the latent theoretic
  - Linear approximation (R^2) of a non-linear process = dampening of signal

1.4 GLM (Poisson Regression)

1.4. Poisson Example. Observation Empirical

- If response are true counts, observation empirical can be extended to other distributions (Poisson, negative binomial)
- These other distributions have no theoretical variance (like binomial)
- “Go ahead, log-transform count data” (Ives 2015)
  - Compare standardized coefficients from LM fit to log(*y*) vs. GLM fit to Poisson distribution to see how close we can get use this OE approach

1.4. Poisson Example.

```
# Generate Poisson distributed data
set.seed(100)

count_data <- data.frame(y = rpois(100, 10))

count_data$x <- count_data$y * runif(100, 0, 5)

# Fit log-transformed response using LM and extract standardized
coefficient
lm_model <- lm(log(y) ~ x, count_data)

stdCoefs(lm_model)$Std.Estimate
[1] 0.5346

with(count_data, cor(x, log(y))) # same as correlation
[1] 0.5345506
```

1.4. Poisson Example.

```
# fit GLM and extract coefficient (link-scale)
glm_model2 <- glm(y ~ x, family = poisson(link = "log"), count_data)

coef(glm_model2)[2]
 x
0.01204693

# compute observation empirical sd by hand
R2 <- cor(count_data$y, predict(glm_model2, type = "response"))^2

sd.yhat <- sqrt(var(predict(glm_model2, type = "link"))/R2)

coef(glm_model2)[2] * sd(count_data$x)/sd.yhat
 x
0.5695438

# get from coefs
stdCoefs(glm_model2)$Std.Estimate
[1] 0.5695438

# compare to LM model r.squared
sqrt(summary(lm_model)$r.squared)
[1] 0.5345506
```

1.4. Poisson Example. Observation Empirical

- Standardized coefficient from log-transformed LM *very* similar to GLM fit with a log-link (differences due to under-the-hood machinery)

- Extends to negative binomial as well

- Should be link-invariant (exercise: repeat with sqrt-link)

- Other distributions (beta, gamma, etc.) have multiple parameters that denote the shape of the relationship → still working on how to extend this observation empirical approach