Extending linear regression

Adding more variables, transformations, and interactions

Diogo Melo - 28/01/2025

Linear regression is flexible!

Adding flexibility to our models

- The linear model we are using consists of making the parameters of probability distributions change according to some function

- The simplest function is a linear function

$$y_i \sim N(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta x_i$$

- Sometimes the relation between parameters and predictors is not linear

- We can use whatever functional shape we like, but it is useful to use transformations to linearize the relation

Linear regression is flexible!

Some simple transformations

- The linear model we are using consists of making the parameters of probability distributions change according to some function

- The simplest function is a linear function

- Sometimes the relation between parameters and predictors is not linear

- We can use whatever functional shape we like, but it is useful to use transformations to linearize the relation

$$y_i \sim N(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta x_i$$

$$log(y_i) \sim N(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta x_i$$

Log transforming the response

Multiplicative increments

- The **logarithmic transformation** of the response is a common transformation when the effect of the predictor on the response is thought to be multiplicative
  - A change of a unit in x is associated with a constant **percentage** change in y
- Many processes benefit from log transformation:
  - Growth is proportional do previous size
  - Any multiplicative process

$$y_i \sim N(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta x_i$$
$$\Downarrow$$
$$\Delta y \approx \beta \Delta x$$

$$log(y_i) \sim N(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta x_i$$
$$\Downarrow$$
$$\% \, \Delta y \approx (100 \times \beta) \Delta x$$

Houle, D., Pélabon, C., Wagner, G. P. & Hansen, T. F. Measurement and meaning in biology. Q. *Rev. Biol*. 86, 3–34 (2011).

Biomass by diameter

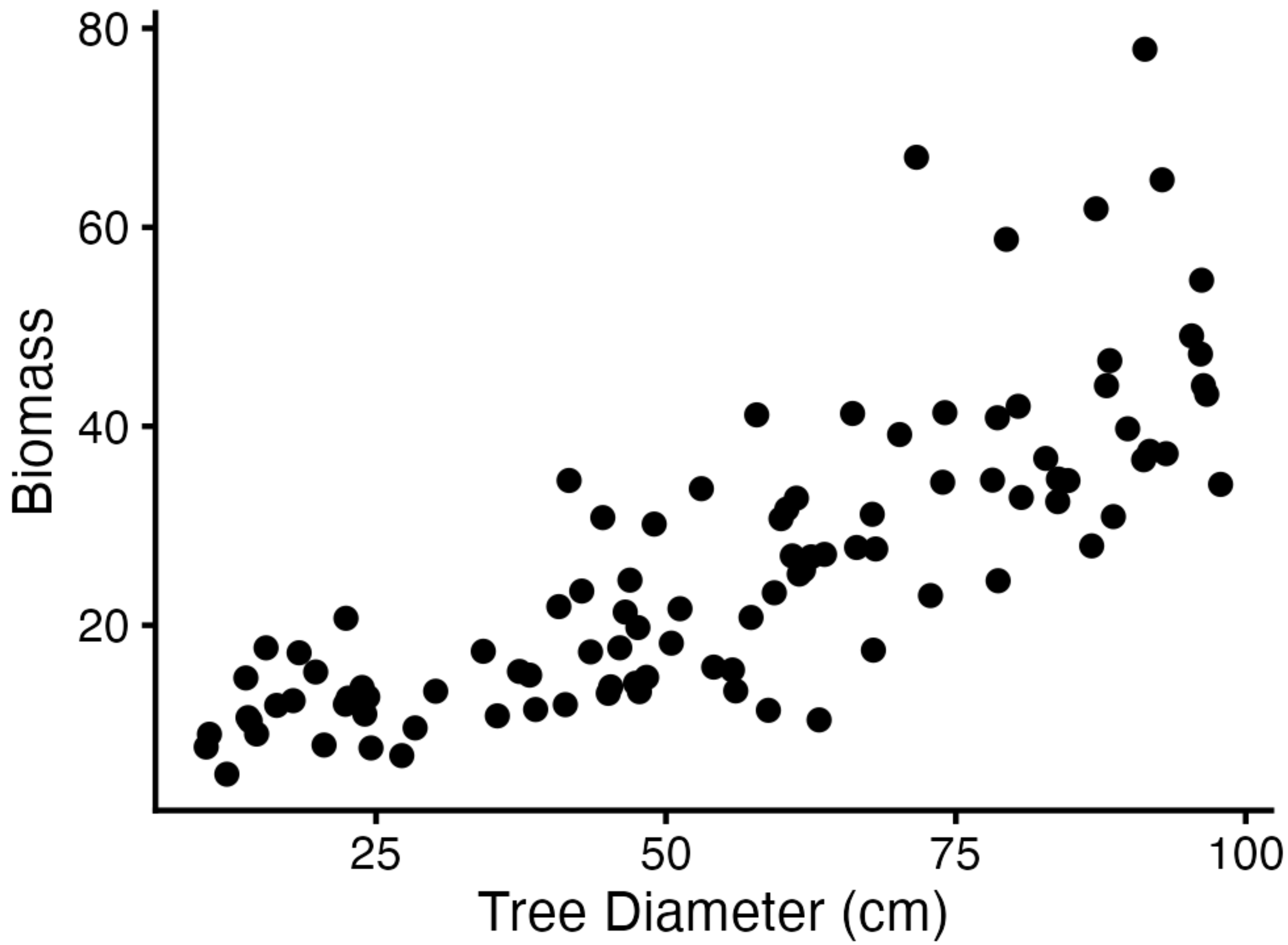
Example of non-linear relation

Option 1: log-transform y

```
df = data.frame(diameter, biomass)
stan_fit = stan_glm(log(biomass) ~ diameter,
                    data = df)
```

Option A: Assign y a log-normal likelihood

```
rt_fit = ulam(alist(
           biomass ~ lognormal(mu, sigma),
           mu <- a + b*diameter,
           a ~ normal(0, 2),
           b ~ normal(0, 1),
           sigma ~ exponential(1)),
        data = df, chains = 4, cores = 4)
```



Biomass by diameter

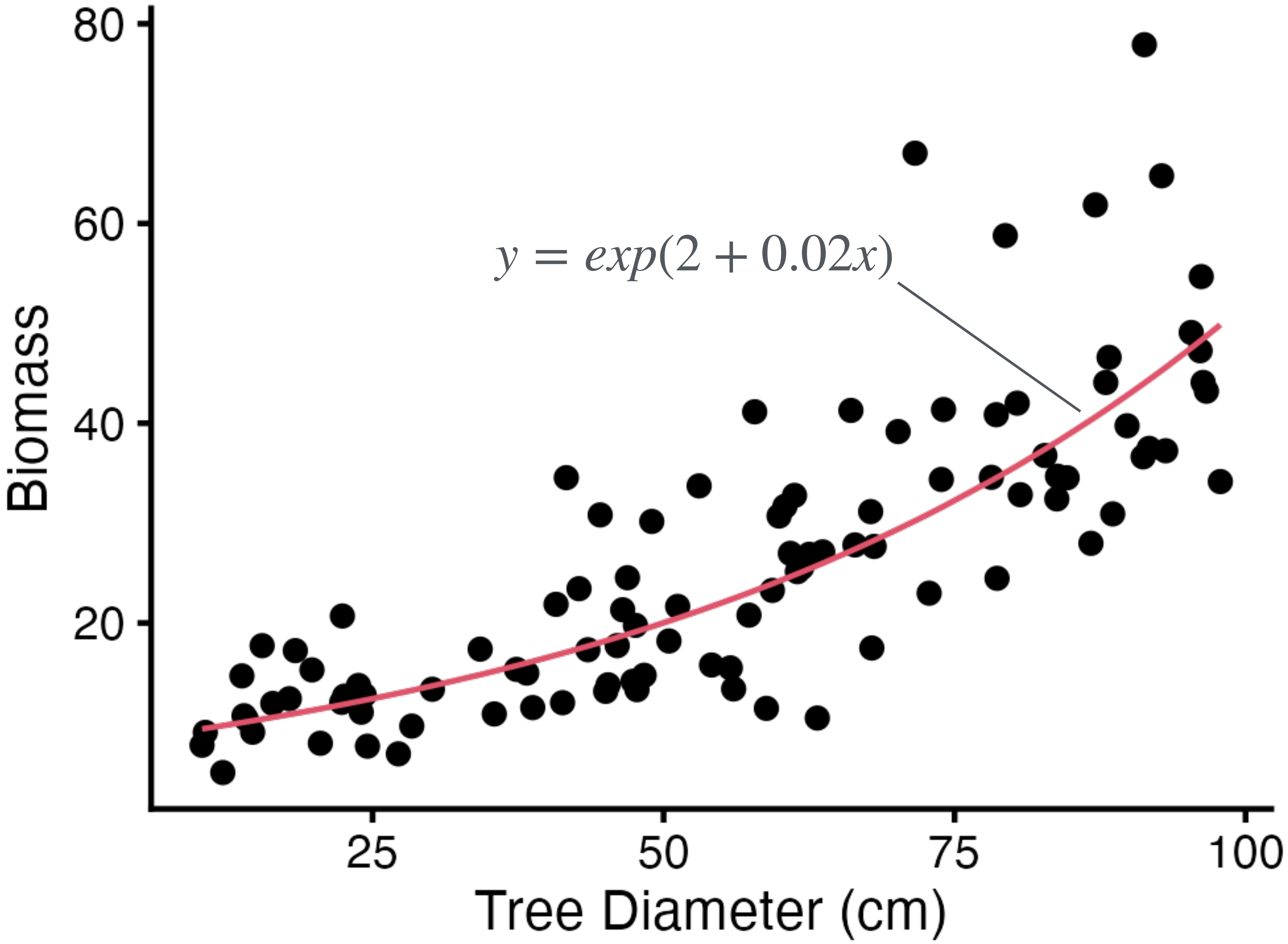
Example of non-linear relation

Option 1: log-transform y

```
df = data.frame(diameter, biomass)
stan_fit = stan_glm(log(biomass) ~ diameter,
                    data = df)
```

Option A: Assign y a log-normal likelihood

```
rt_fit = ulam(alist(
           biomass ~ lognormal(mu, sigma),
           mu <- a + b*diameter,
           a ~ normal(0, 2),
           b ~ normal(0, 1),
           sigma ~ exponential(1)),
        data = df, chains = 4, cores = 4)
```



Biomass by diameter

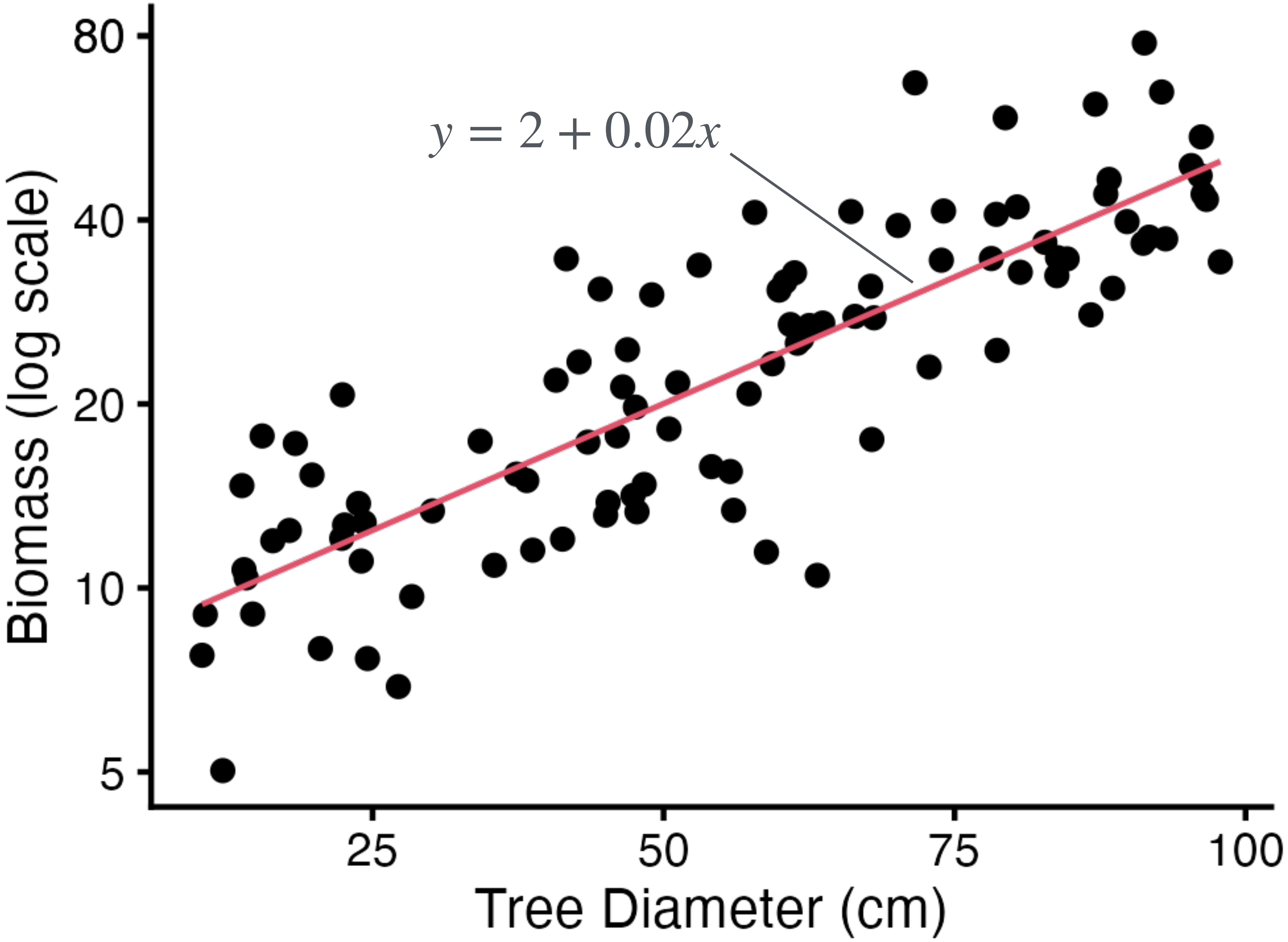
Example of non-linear relation

Option 1: log-transform y

```
df = data.frame(diameter, biomass)
stan_fit = stan_glm(log(biomass) ~ diameter,
                    data = df)
```

Option A: Assign y a log-normal likelihood

```
rt_fit = ulam(alist(
           biomass ~ lognormal(mu, sigma),
           mu <- a + b*diameter,
           a ~ normal(0, 2),
           b ~ normal(0, 1),
           sigma ~ exponential(1)),
        data = df, chains = 4, cores = 4)
```



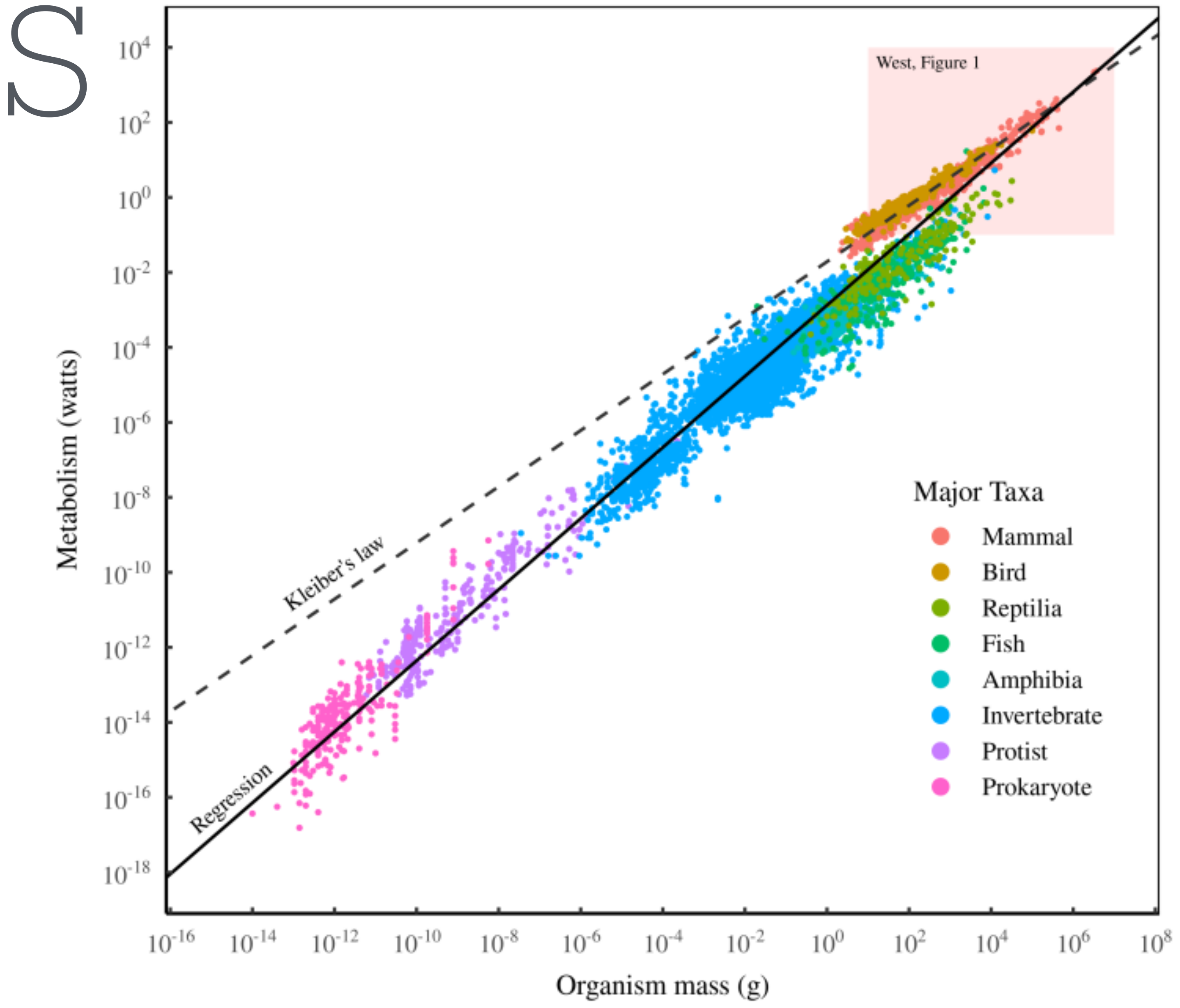
Power-law relations

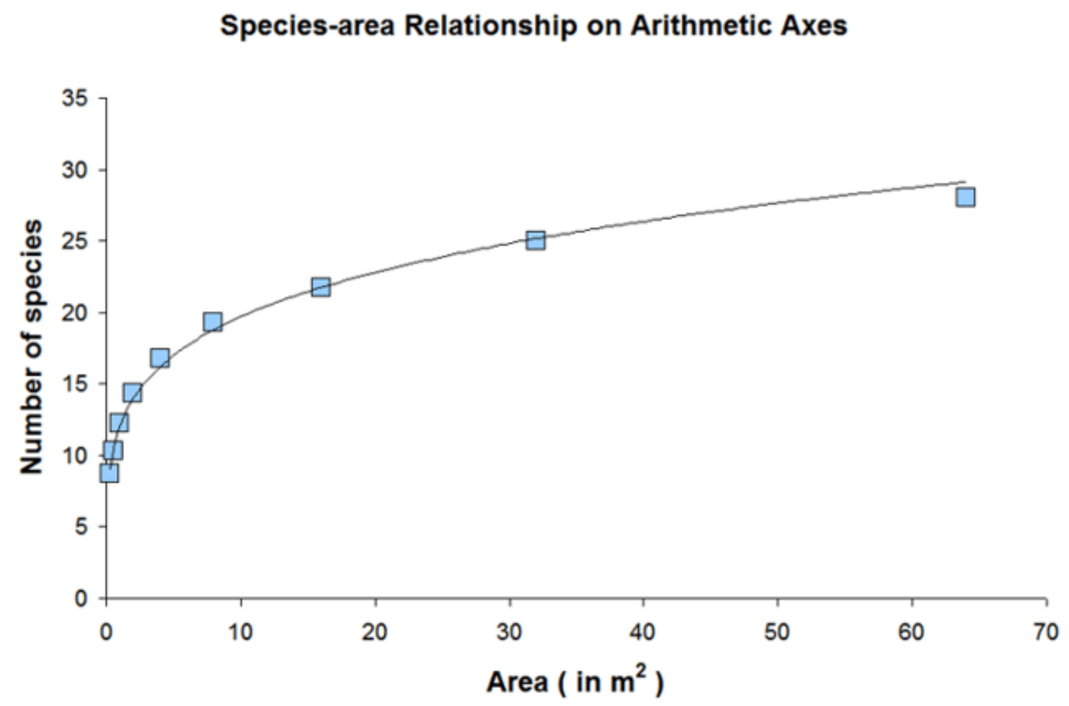
Log-log regressions

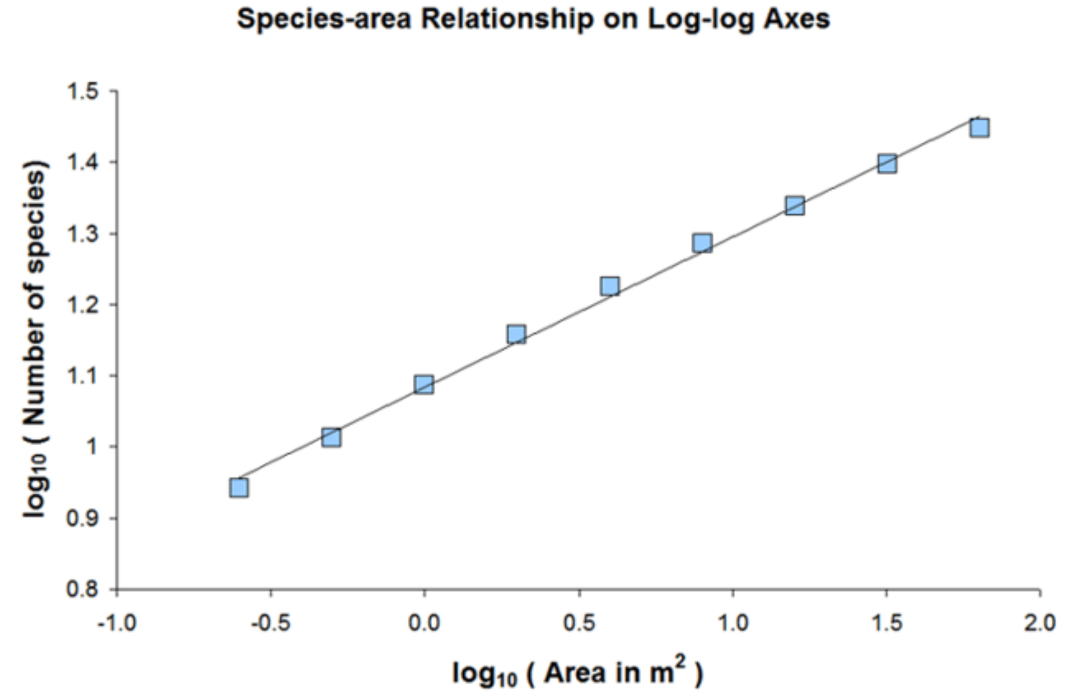
- Several biological relations take the form of power-law relations

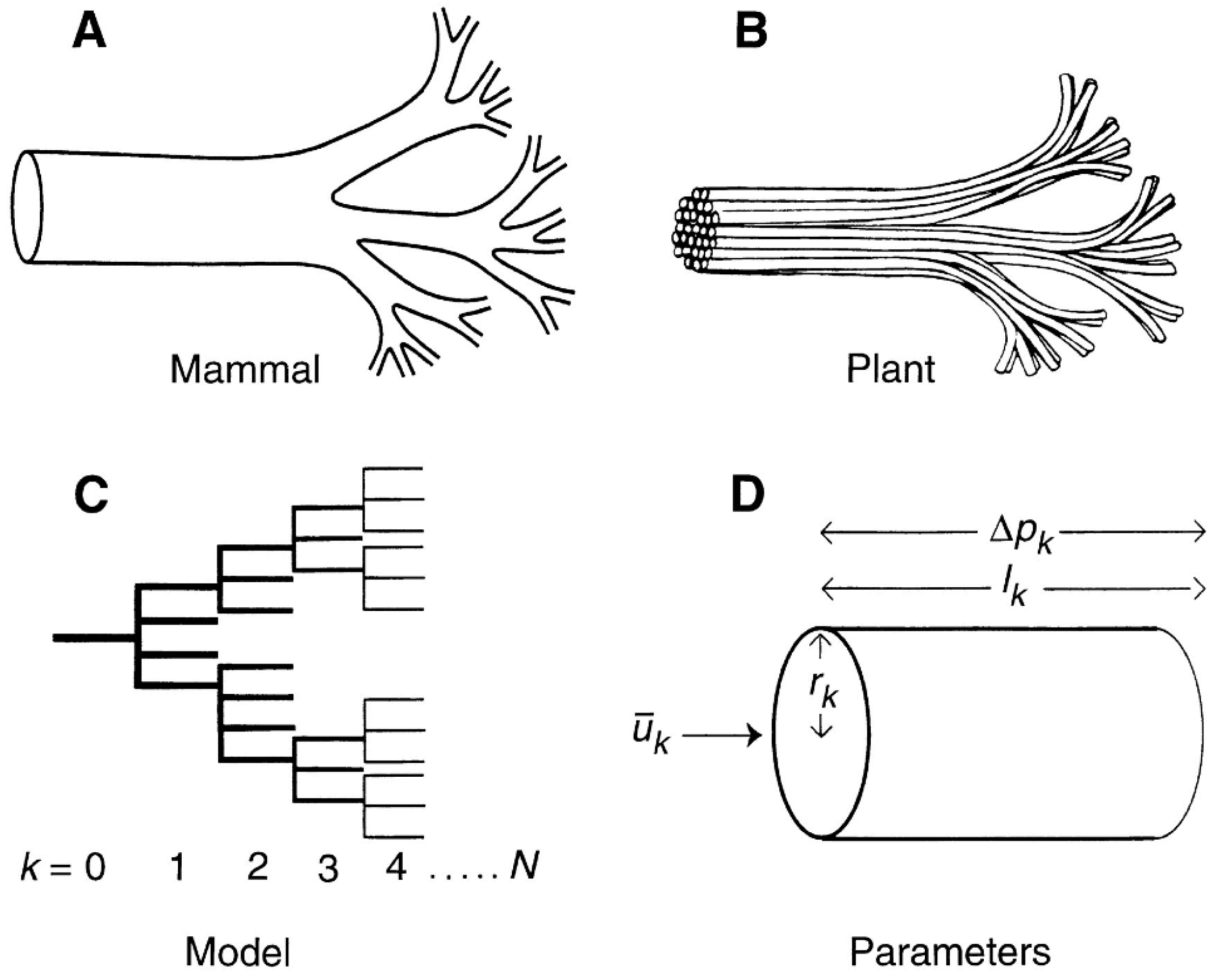
$$y \propto ax^b$$

- These appear for different reasons:
  - West et al. (1997) attempt to link scaling laws to fractal relations at different scales
  - Every time a proportional increase leads to a consistent proportional change (like in species-area relations)









West, G. B., Brown, J. H. & Enquist, B. J. A general model for the origin of allometric scaling laws in biology. *Science* **276**, 122–126 (1997).

Power-law relations

Log-log regressions

- Several biological relations take the form of power-law relations
- We can linearize these relations using a log-log transformation

$$y \propto ax^b$$

Take the log on both sides

$$log(y) \propto log(ax^b) = log(a) + log(x^b) =$$

$$\boxed{log(y) \propto log(a) \;+\; b\; log(x)}$$

West, G. B., Brown, J. H. & Enquist, B. J. A general model for the origin of allometric scaling laws in biology. *Science* **276**, 122–126 (1997).

Power-law relations

Log-log regressions

- Several biological relations take the form of power-law relations
- We can linearize these relations using a log-log transformation
- In this model, the slope is and estimate of the exponent of the power-law
- The interpretation of the slope is that a 1% increase in x leads to a $\beta\,\%$ increase in y

$$log(y_i) \sim N(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta \; log(x_i)$$

$$\Downarrow$$

$$\%\,\Delta y \approx \beta\,\%\,\Delta x$$

West, G. B., Brown, J. H. & Enquist, B. J. A general model for the origin of allometric scaling laws in biology. *Science* **276**, 122–126 (1997).

Quick reference for log transformations

Model	Dependent variable	Independent Variable	Interpretation of β
Level-level	y	x	$\Delta y \approx \beta \Delta x$
Level-log	y	log(x)	$\Delta y \approx (\frac{\beta}{100})\,\%\,\Delta x$
log-level	log(y)	x	$\%\,\Delta y \approx (100\ \beta)\Delta x$
Log-log	log(y)	log(x)	$\%\,\Delta y \approx \beta\,\%\,\Delta x$

Linear regression is flexible!

We can modify our functions however we like

- The linear model we are using consists of making the parameters of probability distributions change according to some function

- The simplest function involves a single predictor and slope

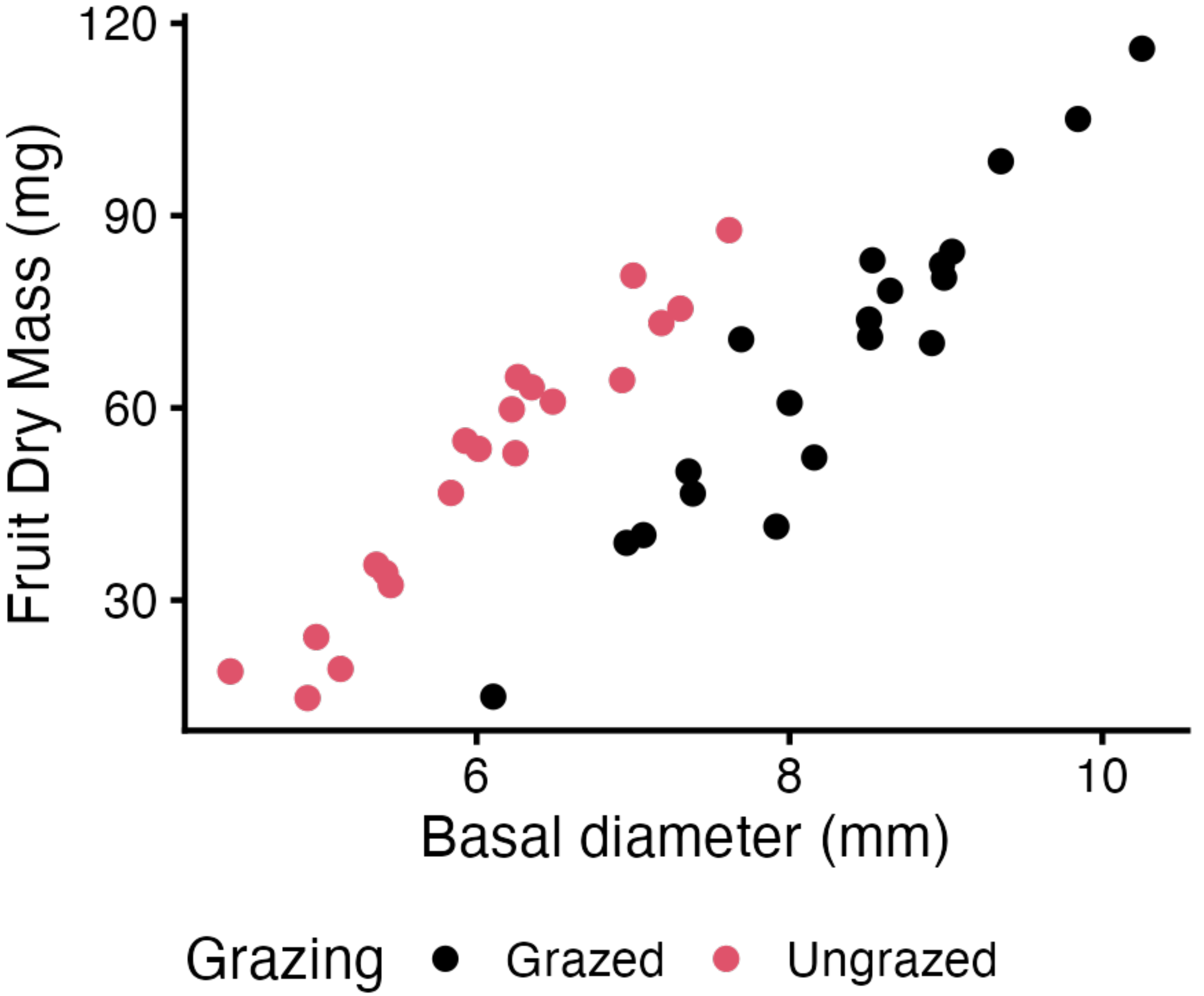
$$y_i \sim N(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta x_i$$

- If we have more predictors, we can simply add them to the regression equation

$$y_i \sim N(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2}$$

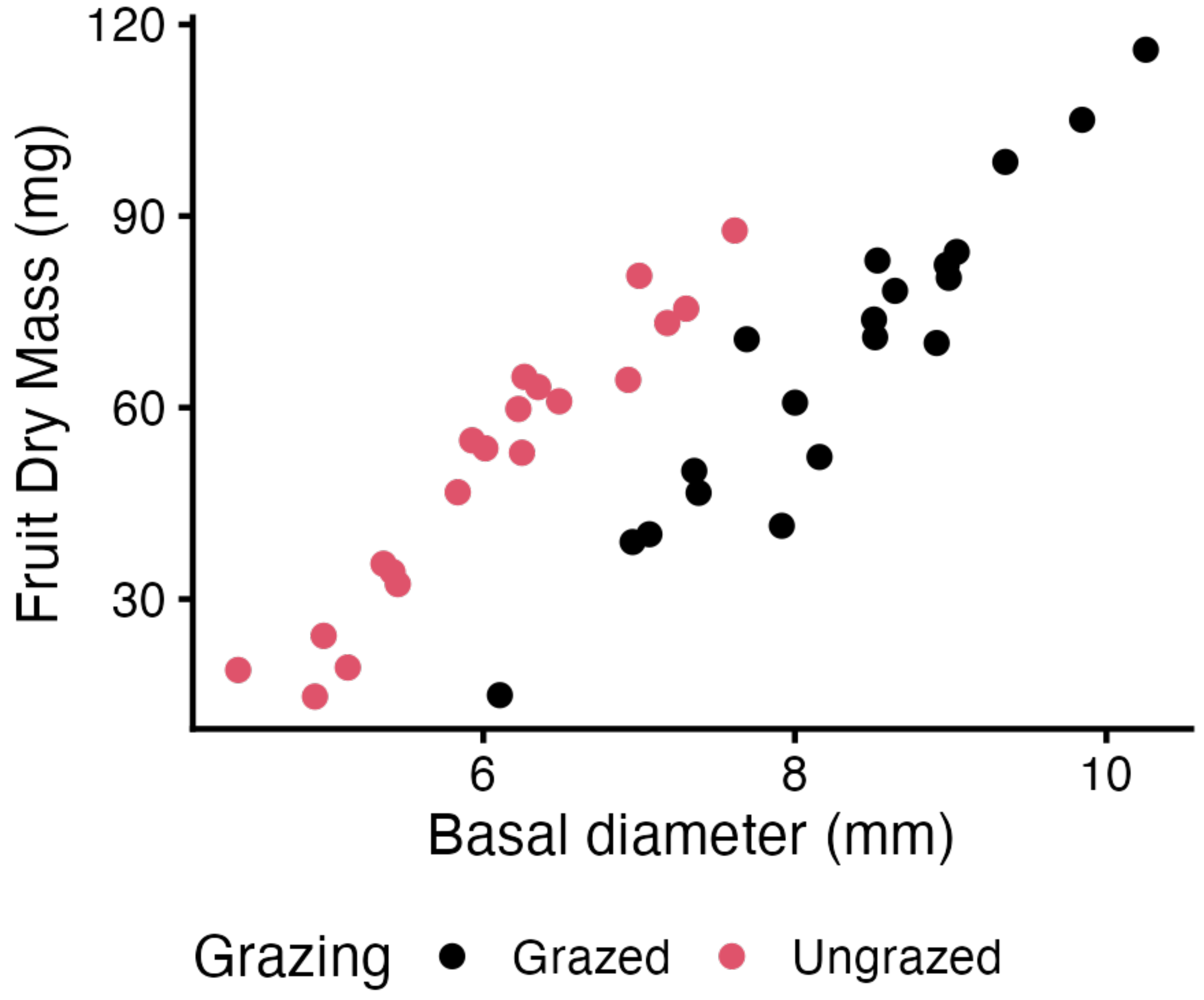
Example with more predictors

- **Question**: what's the impact of herbivory on plant fitness?
- **Field experiment**: 40 plants of *Ipomopsis agreggata* assigned at random to two treatments: unprotected from grazing by rabbits and protect from grazing by fenced cages.
- Response variable: fruit yield of each plant (mg dry mass)
- Predictor variables:
    - Treatment: fenced or non-fenced
    - Basal diameter of each plant (mm), measured before the treatment



Data from: Crawley (2007) The R Book

Scale variables



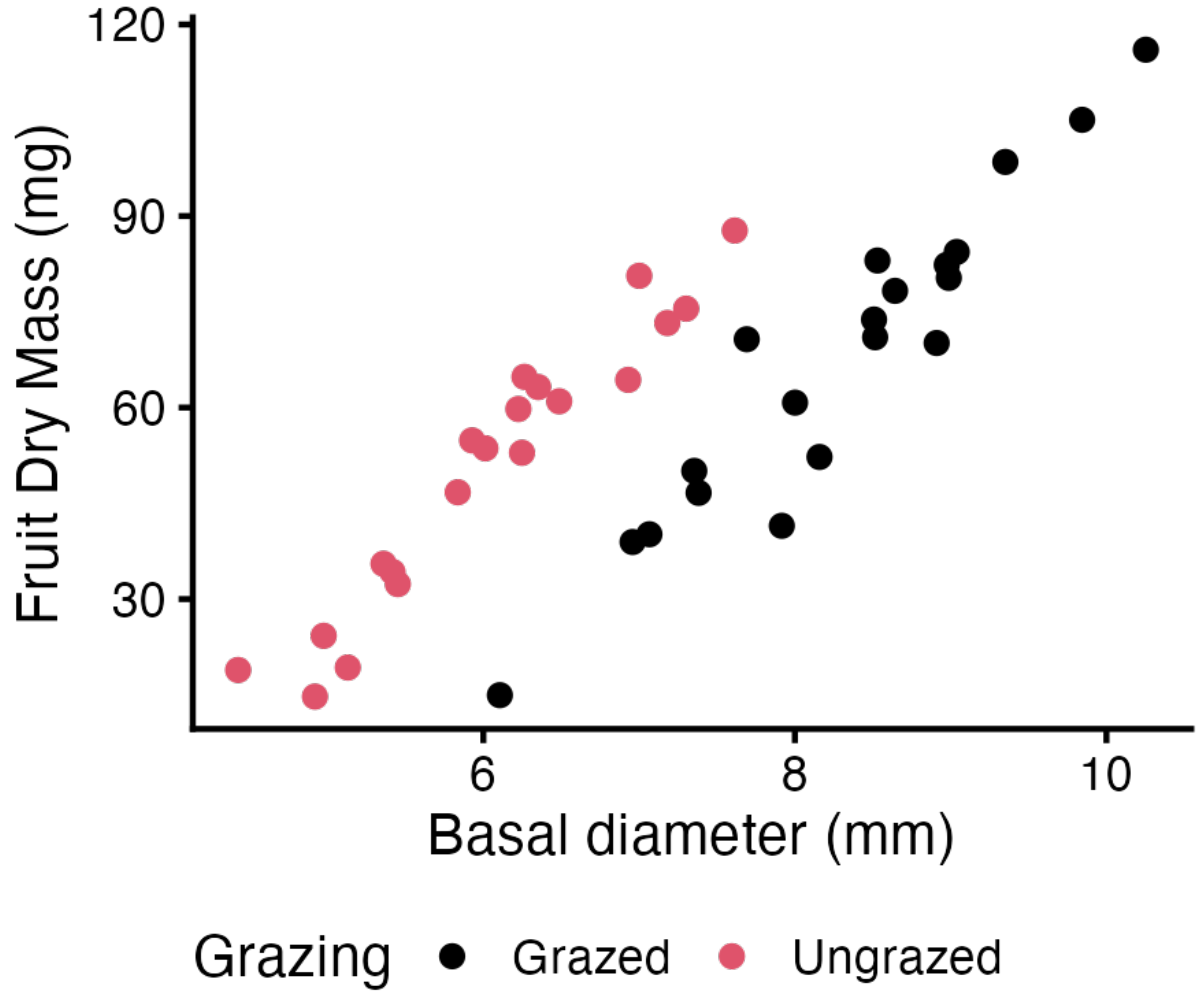
- It's good practice to scale variables by their standard deviation and subtract the mean:

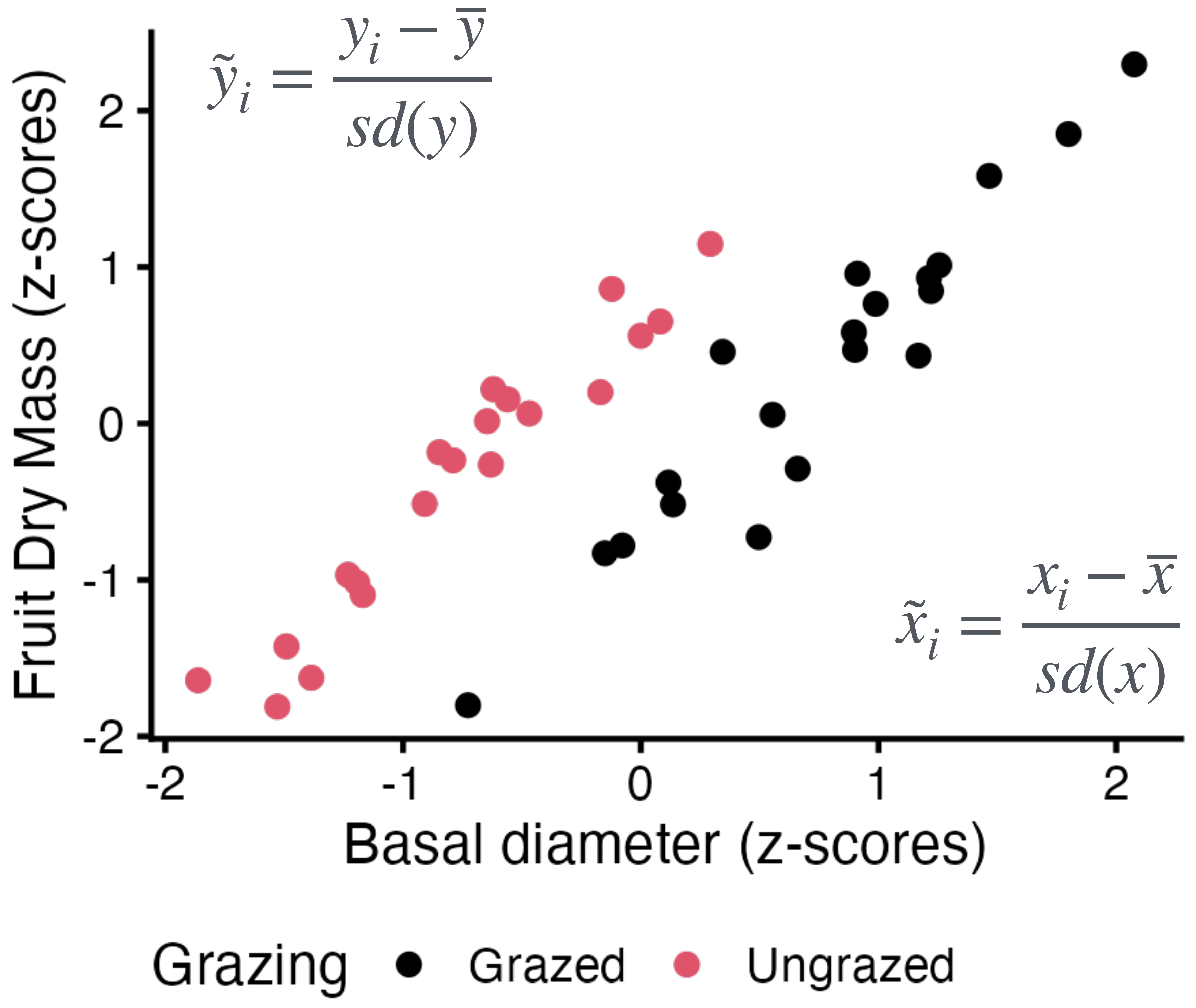
$$\tilde{y}_i = \frac{y_i - \bar{y}}{sd(y)}$$

- The **z-score** of a continuous variable is a measure of how many standard deviations a data point is from the mean of the dataset
- Using z-scores makes coefficients easier to interpret and comparable across variables with different scales
- The transformation is linear, and we can always recover parameter values on their original scale by multiplying by the standard deviation

Scale variables

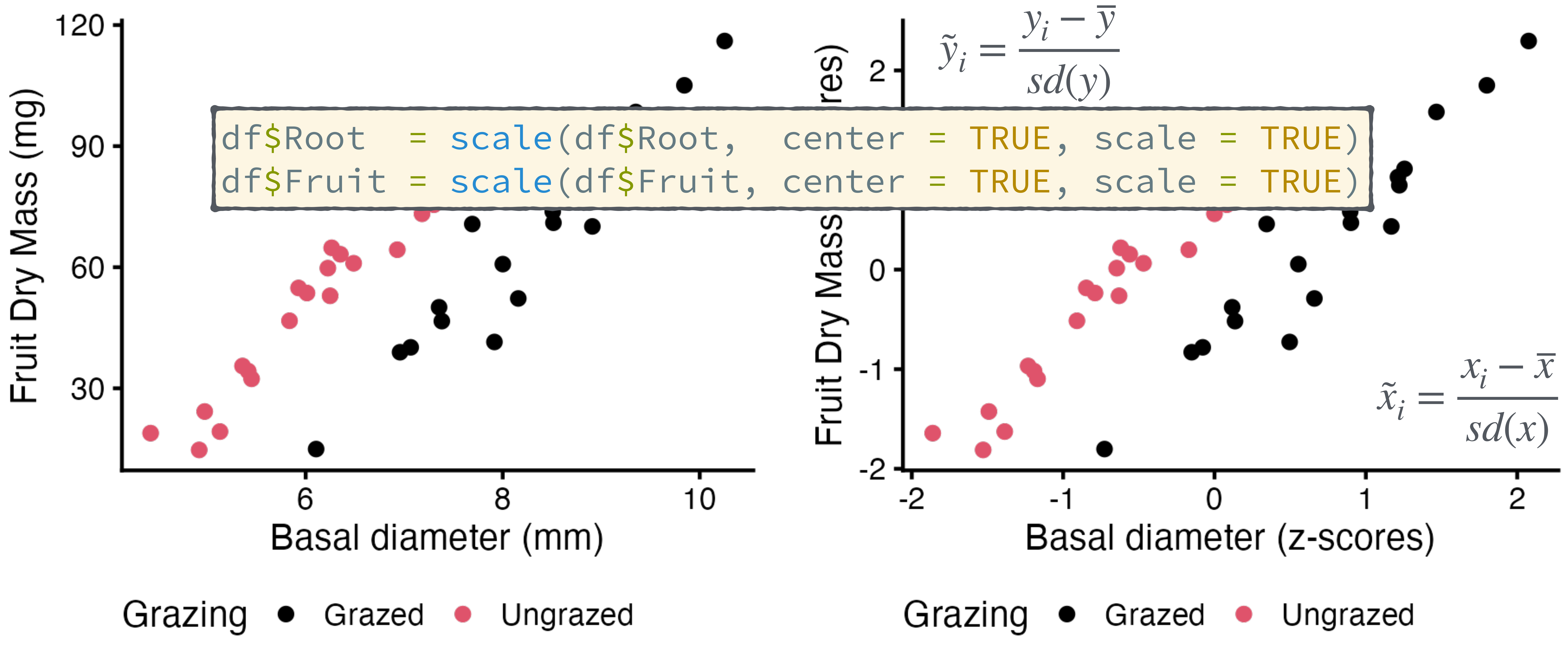
Using standard deviation units makes everything simpler





Scale variables

Using standard deviation units makes everything simpler

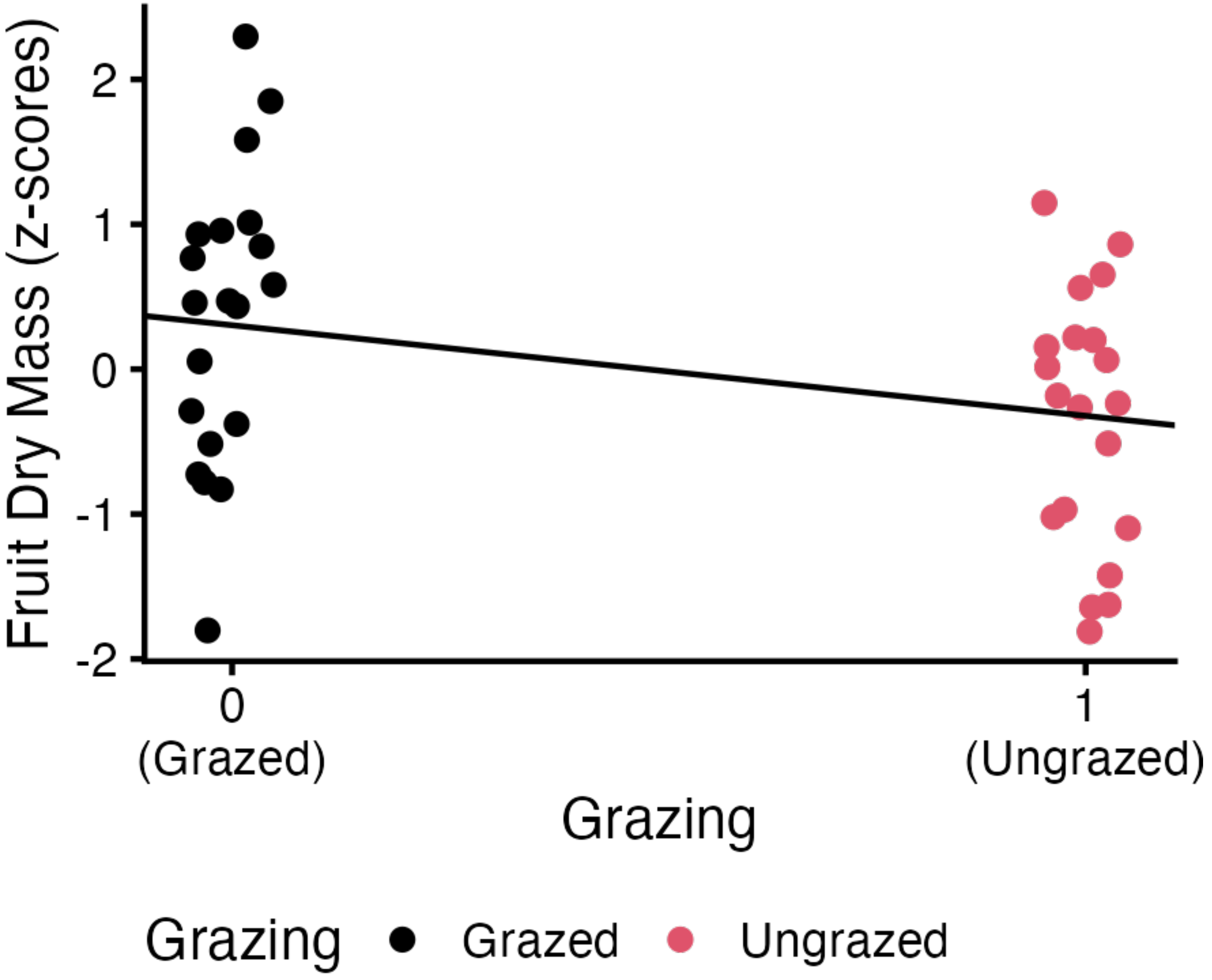



```r
df$Root  = scale(df$Root,  center = TRUE, scale = TRUE)
df$Fruit = scale(df$Fruit, center = TRUE, scale = TRUE)
```

Model with only treatment

```r
df$Grazing0 = ifelse(df$Grazing ==
"Ungrazed", 1, 0) # numeric coding

m1 = ulam(alist(Fruit ~ normal(mu, sigma),
          mu <- a + b*Grazing0,
          a ~ normal(0, 1),
          b ~ normal(0, 1),
          sigma ~ exponential(1)),
     data = df, chains = 4, cores = 4)
```

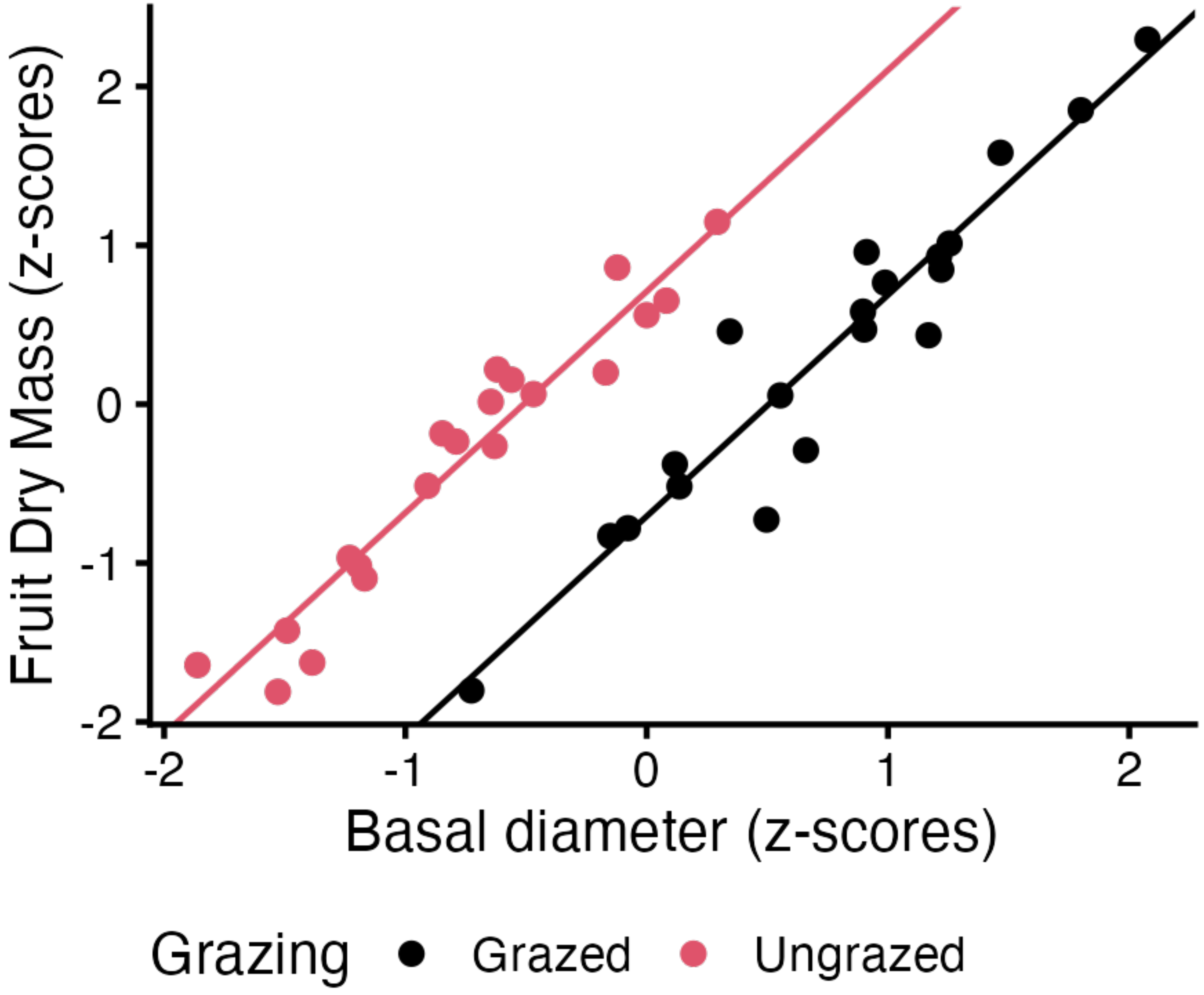
```
> precis(m1, prob = 0.95)
       mean   sd  2.5% 97.5%
a      0.32 0.21 -0.09  0.72
b     -0.66 0.30 -1.23 -0.06
sigma  0.97 0.12  0.77  1.22
```



Model with treatment and size

```
m2 = ulam(alist(
           Fruit ~ normal(mu, sigma),
           mu <- a + b*Grazing0 + c*Root,
           a ~ normal(0, 1),
           b ~ normal(0, 1),
           c ~ normal(0, 1),
           sigma ~ exponential(1)),
     data = df, chains = 4, cores = 4)
```

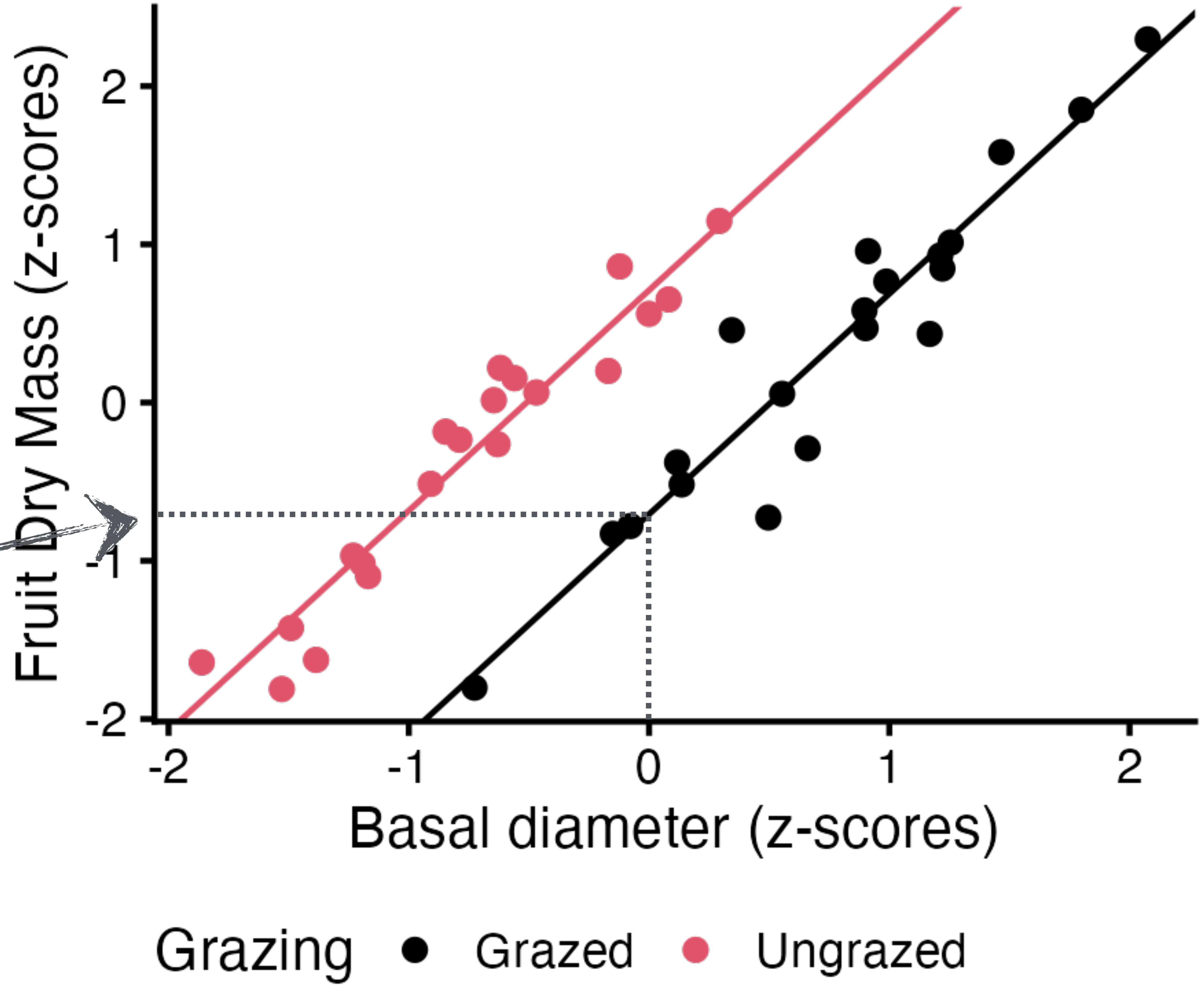
```
> precis(m2, prob = 0.95)
       mean   sd  2.5% 97.5%
a     -0.71 0.08 -0.86 -0.54
b      1.42 0.14  1.13  1.67
c      1.39 0.07  1.26  1.53
sigma  0.28 0.03  0.23  0.36
```



Model with treatment and size

```
m2 = ulam(alist(
           Fruit ~ normal(mu, sigma),
           mu <- a + b*Grazing0 + c*Root,
           a ~ normal(0, 1),
           b ~ normal(0, 1),
           c ~ normal(0, 1),
           sigma ~ exponential(1)),
     data = df, chains = 4, cores = 4)
```

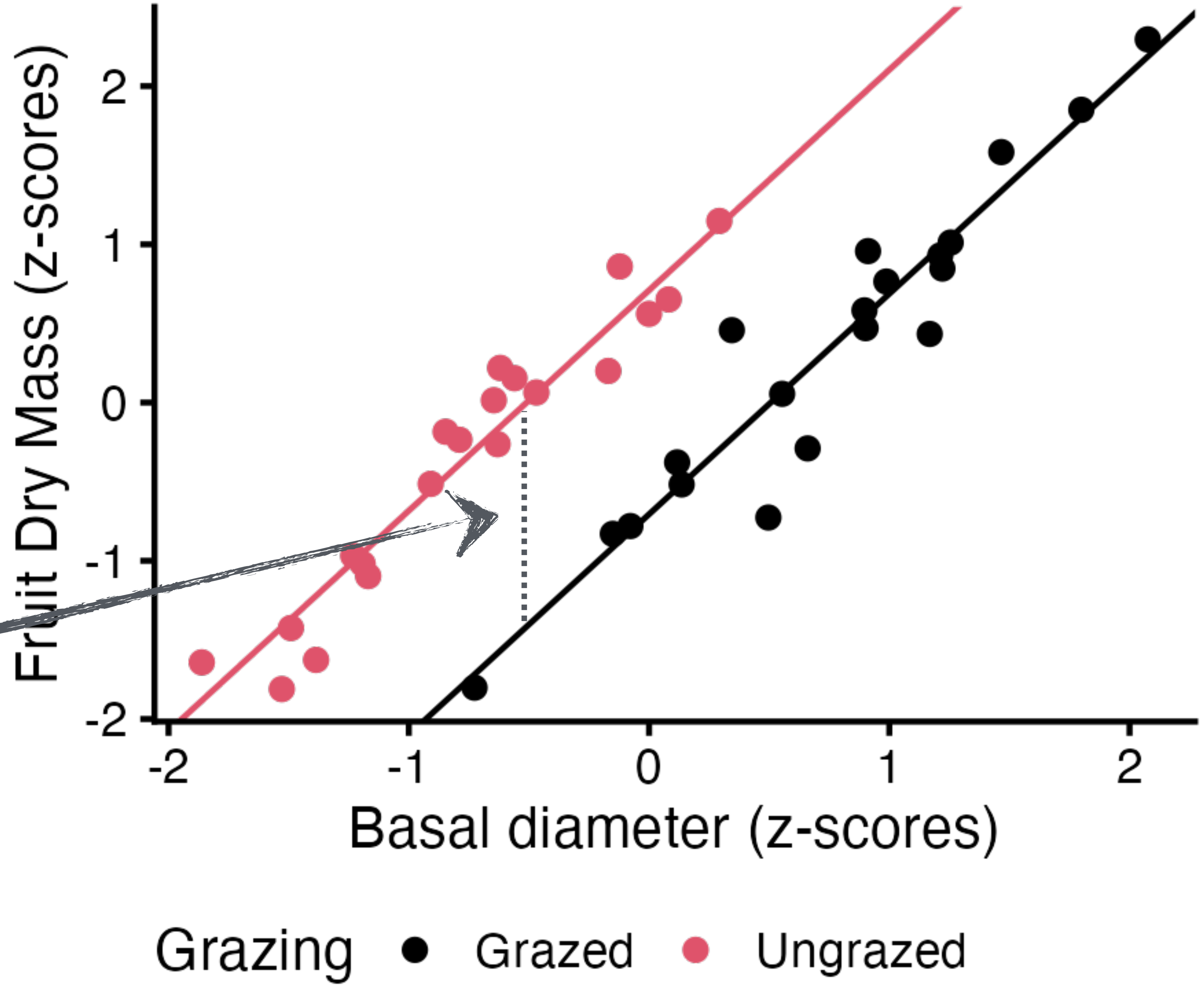
```
> precis(m2, prob = 0.95)
       mean   sd  2.5% 97.5%
a     -0.71 0.08 -0.86 -0.54
b      1.42 0.14  1.13  1.67
c      1.39 0.07  1.26  1.53
sigma  0.28 0.03  0.23  0.36
```



Model with treatment and size

```
m2 = ulam(alist(
          Fruit ~ normal(mu, sigma),
          mu <- a + b*Grazing0 + c*Root,
          a ~ normal(0, 1),
          b ~ normal(0, 1),
          c ~ normal(0, 1),
          sigma ~ exponential(1)),
     data = df, chains = 4, cores = 4)
```

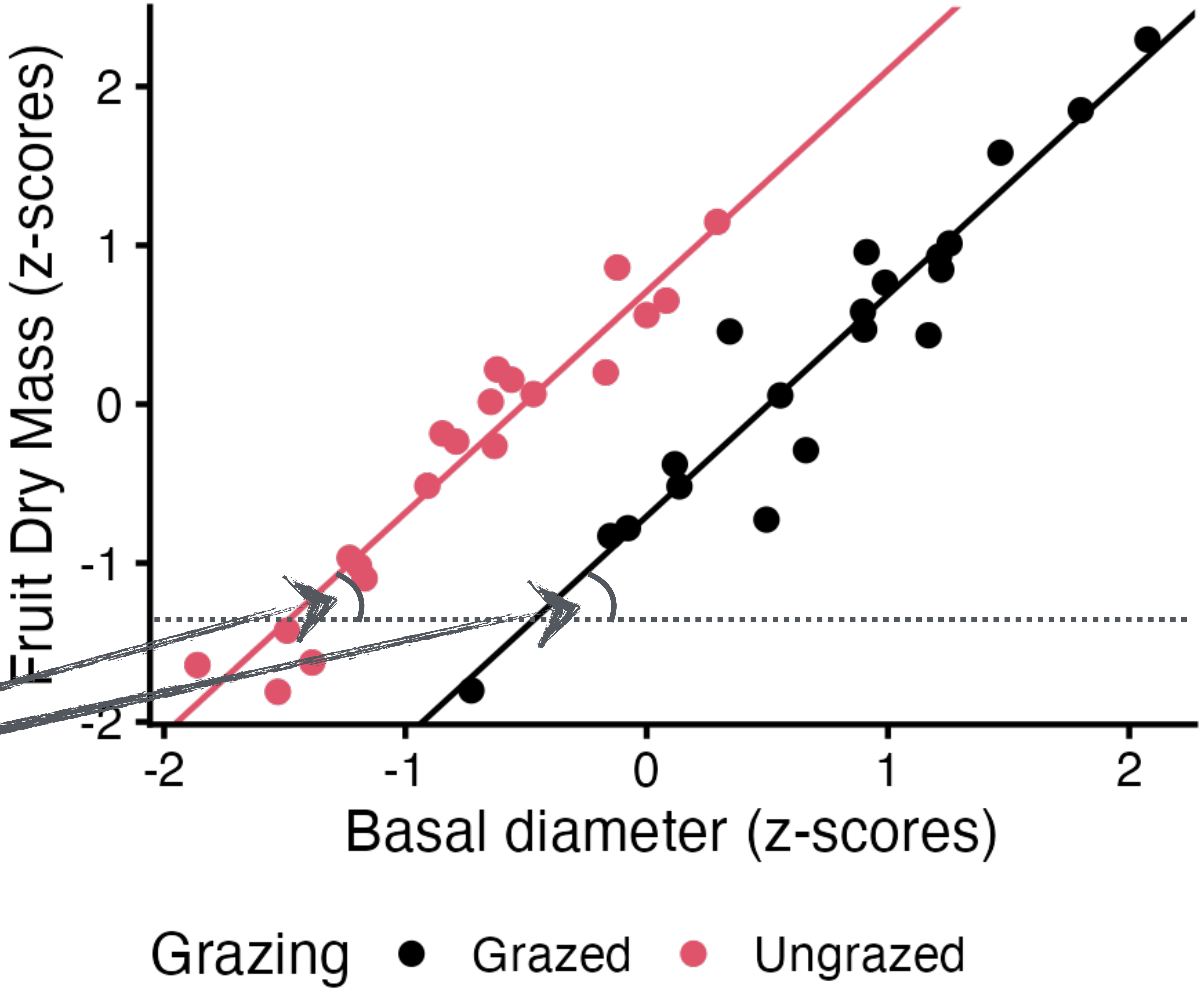
```
> precis(m2, prob = 0.95)
       mean   sd  2.5% 97.5%
a     -0.71 0.08 -0.86 -0.54
b      1.42 0.14  1.13  1.67
c      1.39 0.07  1.26  1.53
sigma  0.28 0.03  0.23  0.36
```



Model with treatment and size

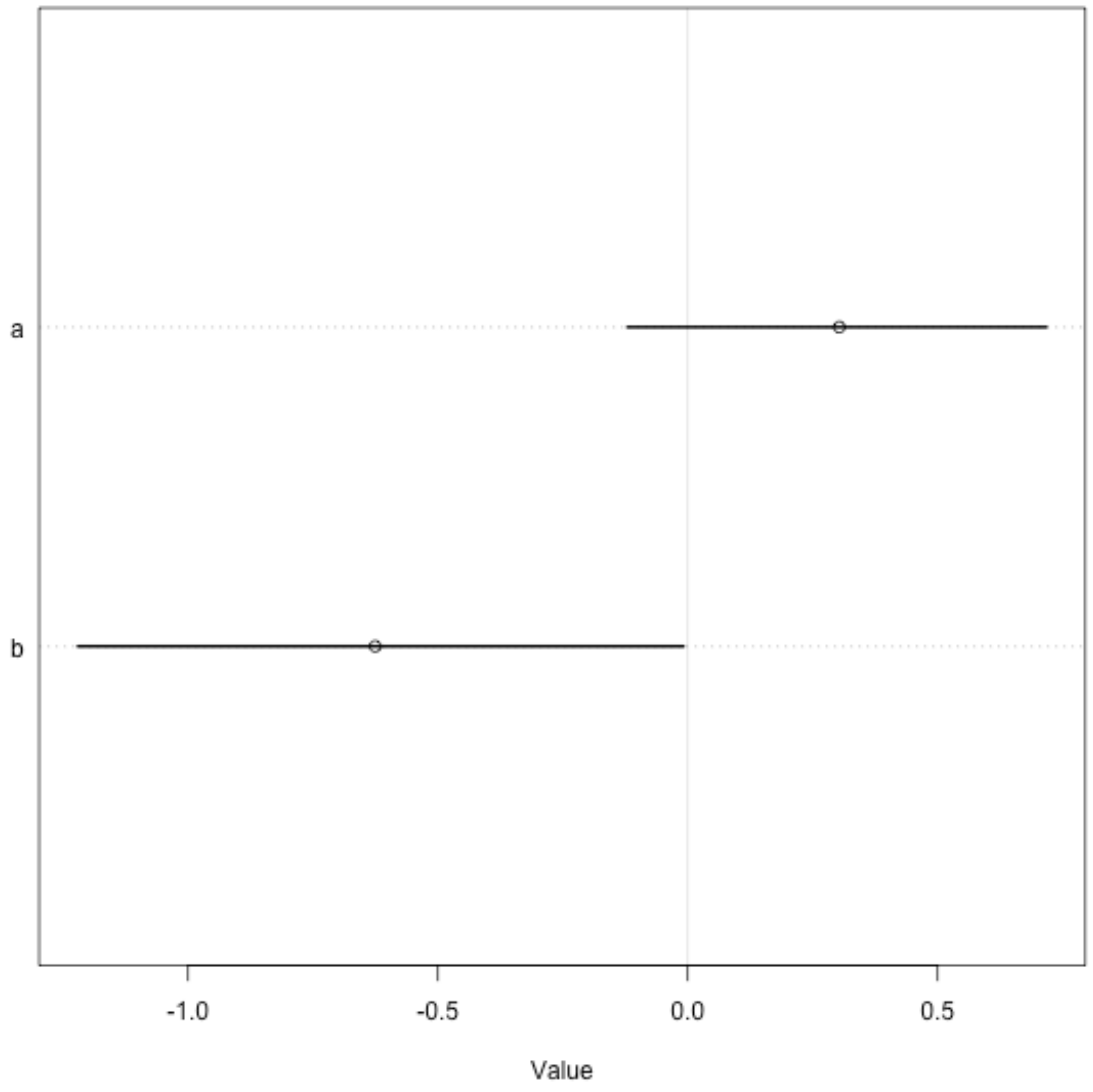
```r
m2 = ulam(alist(
            Fruit ~ normal(mu, sigma),
            mu <- a + b*Grazing0 + c*Root,
            a ~ normal(0, 1),
            b ~ normal(0, 1),
            c ~ normal(0, 1),
            sigma ~ exponential(1)),
      data = df, chains = 4, cores = 4)
```

```
> precis(m2, prob = 0.95)
       mean   sd  2.5% 97.5%
a     -0.71 0.08 -0.86 -0.54
b      1.42 0.14  1.13  1.67
c      1.39 0.07  1.26  1.53
sigma  0.28 0.03  0.23  0.36
```

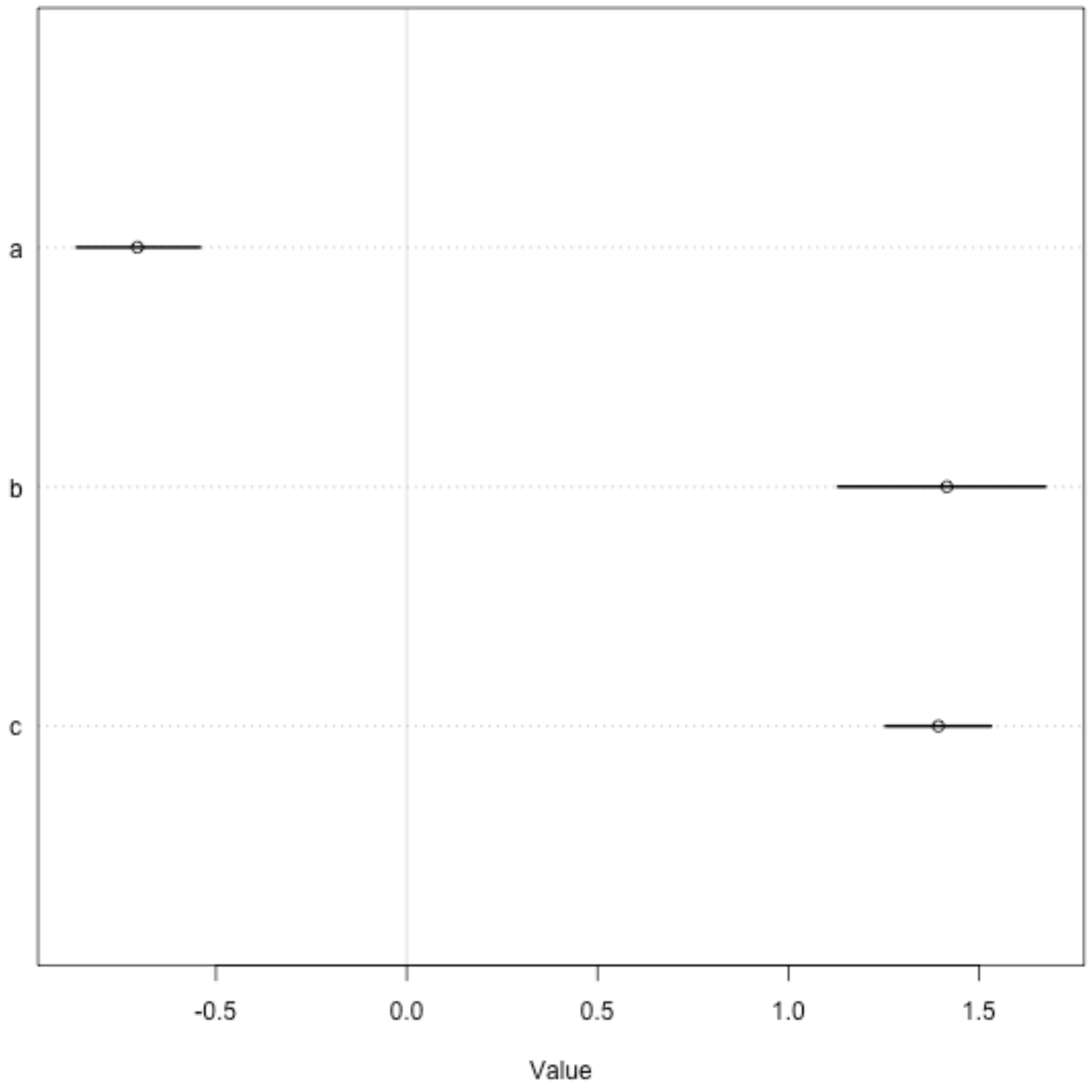


Comparing model estimates

$$y_i \sim N(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta x_i$$



$$y_i \sim N(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2}$$



Why do the coefficients change?

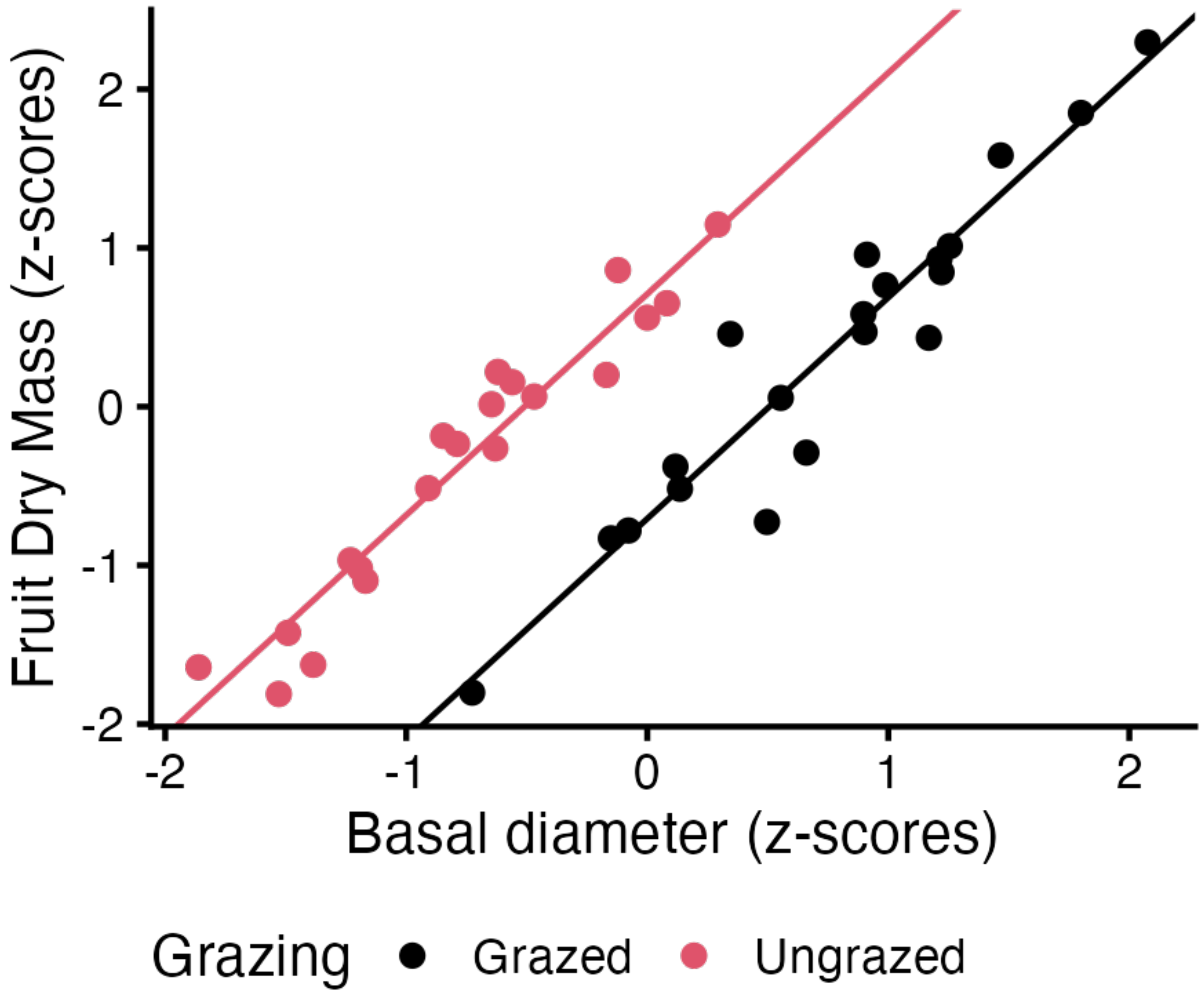
Multiple regression as control

- Coefficients are comparisons, and adding predictors to a regression has the same effect as **stratifying** the data

- The objetive of adding more predictors to a linear model is to compare **like-to-like**:

  1. What is the difference between treatments for plants of the same initial size?

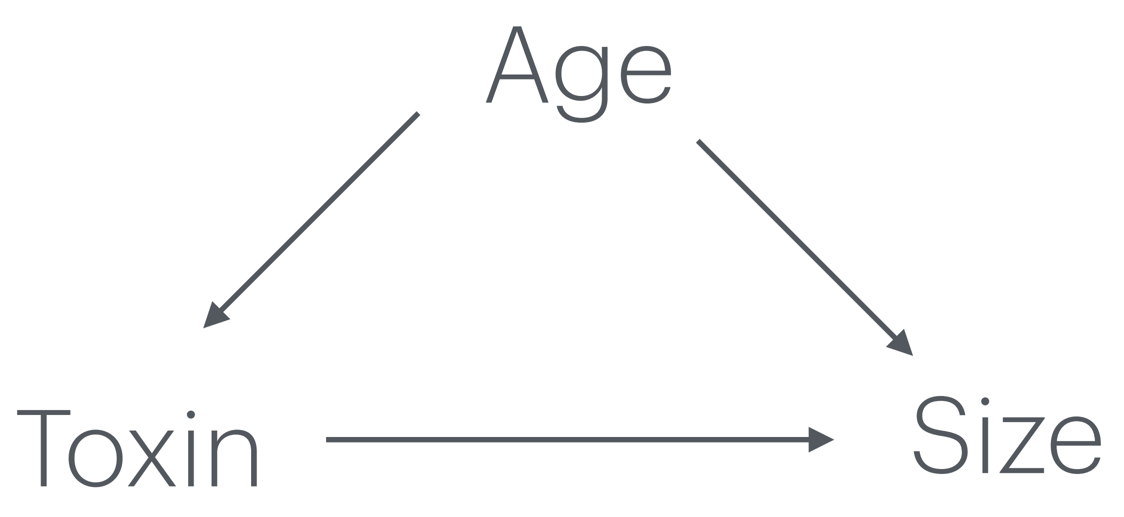
  2. What is the effect of initial size given a particular treatment?



Using simulations to understand multiple regression

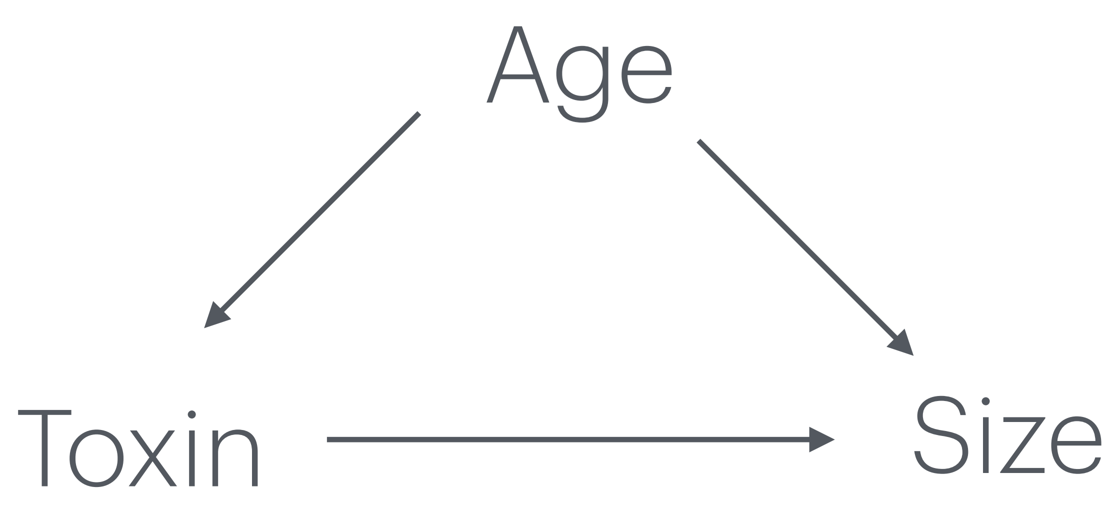
- Simulations are a powerful tool to understand how these models behave
- **Use them liberally! Use them often!**
- Ask:
  - What is the data generating process of system?
  - What model represents this process?
- Simulate data under this model, try to fit the data

- Example:
  - **Question**: What is the effect of toxin exposure on the size of individuals in a population?
  - Age affects both toxin exposure and size



Using simulations to understand multiple regression

- Question: What is the effect of toxin exposure on the size of a population?
- Age affects both toxin exposure and size

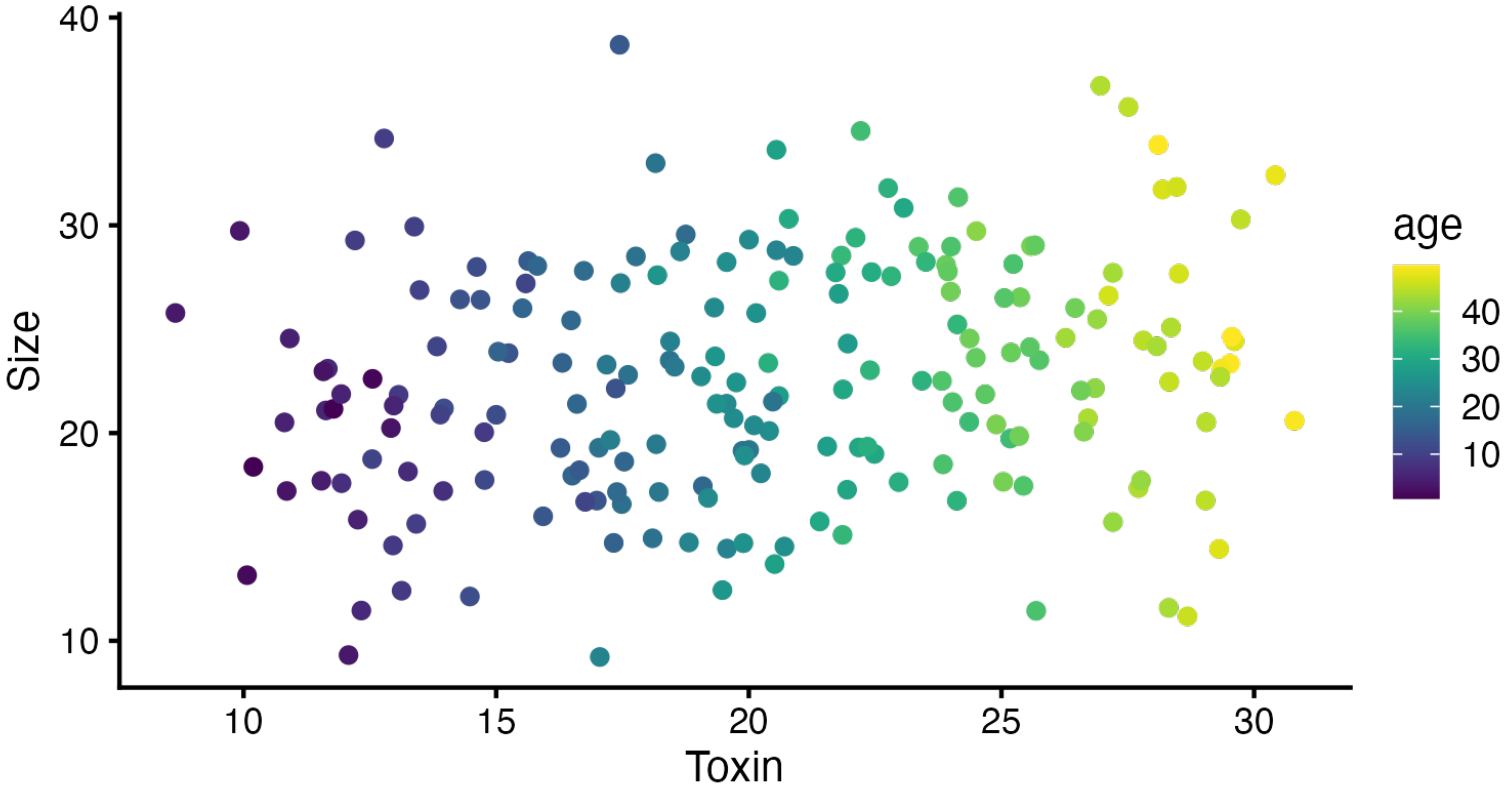


Possible model:

$$age_i = Uniform(0,\ 50)$$
$$toxin_i = Normal(10 + 0.4 \times age_i,\ 1)$$
$$size_i = Normal(30 - 1 \times toxin_i + 0.5 \times age_i,\ 5)$$

```
set.seed(1)
age <- runif(200, 0, 50)
toxin <- rnorm(200, 10 + 0.4*age, 1)
size = 30 - 1*toxin + 0.5*age + rnorm(200, 0, 5)
df = data.frame(age = age,
                toxin = toxin,
                size = size)
```

Simulated data



Center data and define model

```
df0 = df
df0$age   = scale(df0$age, scale = FALSE)
df0$toxin = scale(df0$toxin, scale = FALSE)
df0$size  = scale(df0$size, scale = FALSE)
```

$$size_i \sim Normal(\mu_i, \sigma)$$

$$\mu_i = a + b \times toxin_i + c \times age_i$$

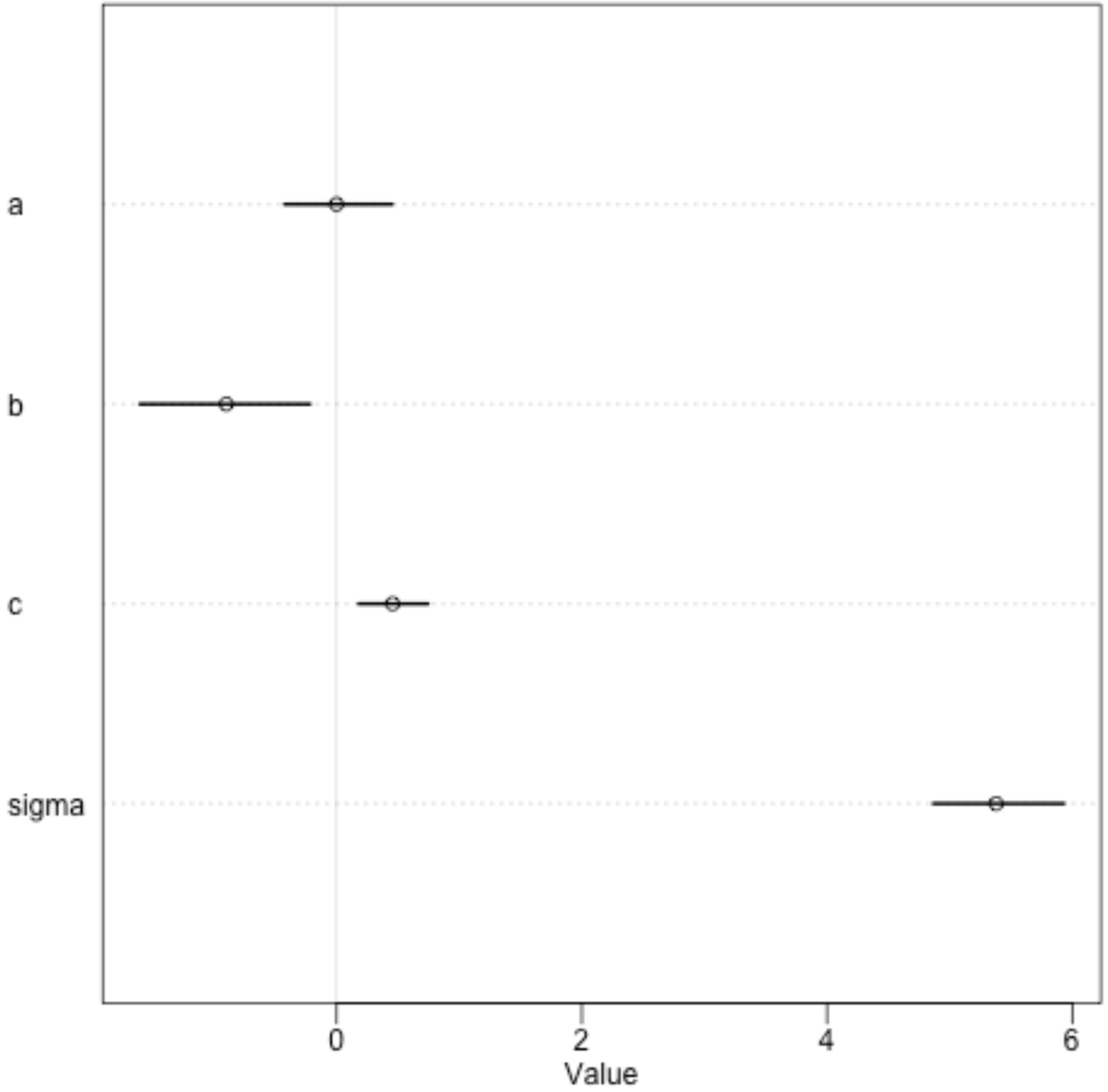
```
rt_fit = ulam(alist(size ~ normal(mu, sigma),
           mu <- a + b*toxin + c*age,
           a ~ normal(0, 0.3),
           b ~ normal(0, 1),
           c ~ normal(0, 1),
           sigma ~ exponential(1)),
        data = df0, chains = 4, cores = 4)
```

Model fit

$$size_i \sim Normal(\mu_i, \sigma)$$

$$\mu_i = a + b \times toxin_i + c \times age_i$$

```
> precis(rt_fit, prob = 0.95)
       mean   sd  2.5% 97.5% rhat ess_bulk
a      0.00 0.22 -0.43  0.45    1  1679.38
b     -0.90 0.35 -1.60 -0.22    1   821.89
c      0.46 0.14  0.18  0.74    1   791.07
sigma  5.38 0.27  4.87  5.93    1  1225.56

> plot(precis(rt_fit, prob = 0.95))
```



Estimates and true values

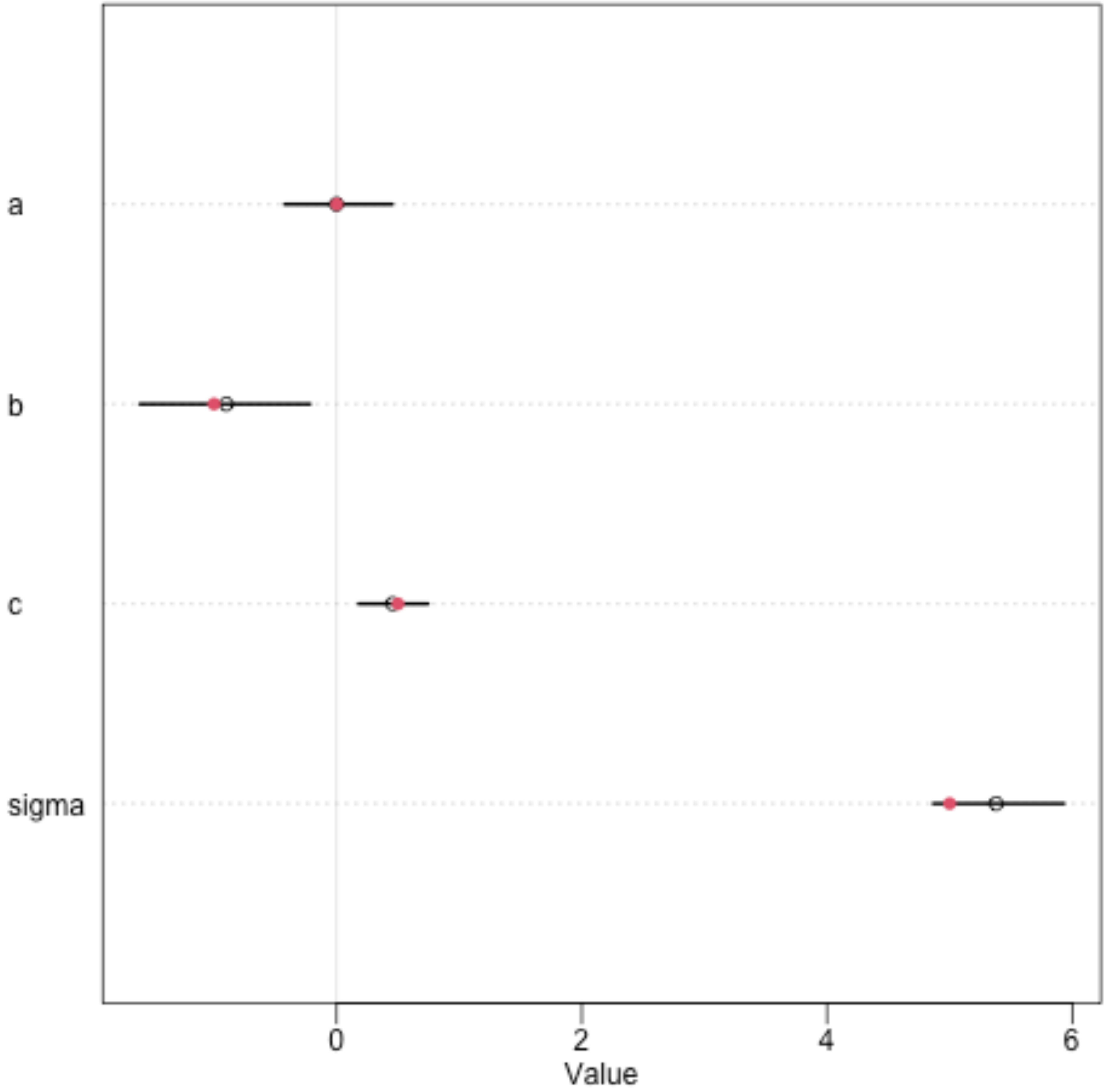
Simulation:

$$age_i = Uniform(0,\ 50)$$
$$toxin_i = Normal(10 + 0.4 \times age_i,\ 1)$$
$$size_i = Normal(30 - 1 \times toxin_i + 0.5 \times age_i,\ 5)$$

Model:

$$size_i \sim Normal(\mu_i, \sigma)$$
$$\mu_i = a + b \times toxin_i \ + \ c \times age$$

Exercise: What happens if we don't include age?



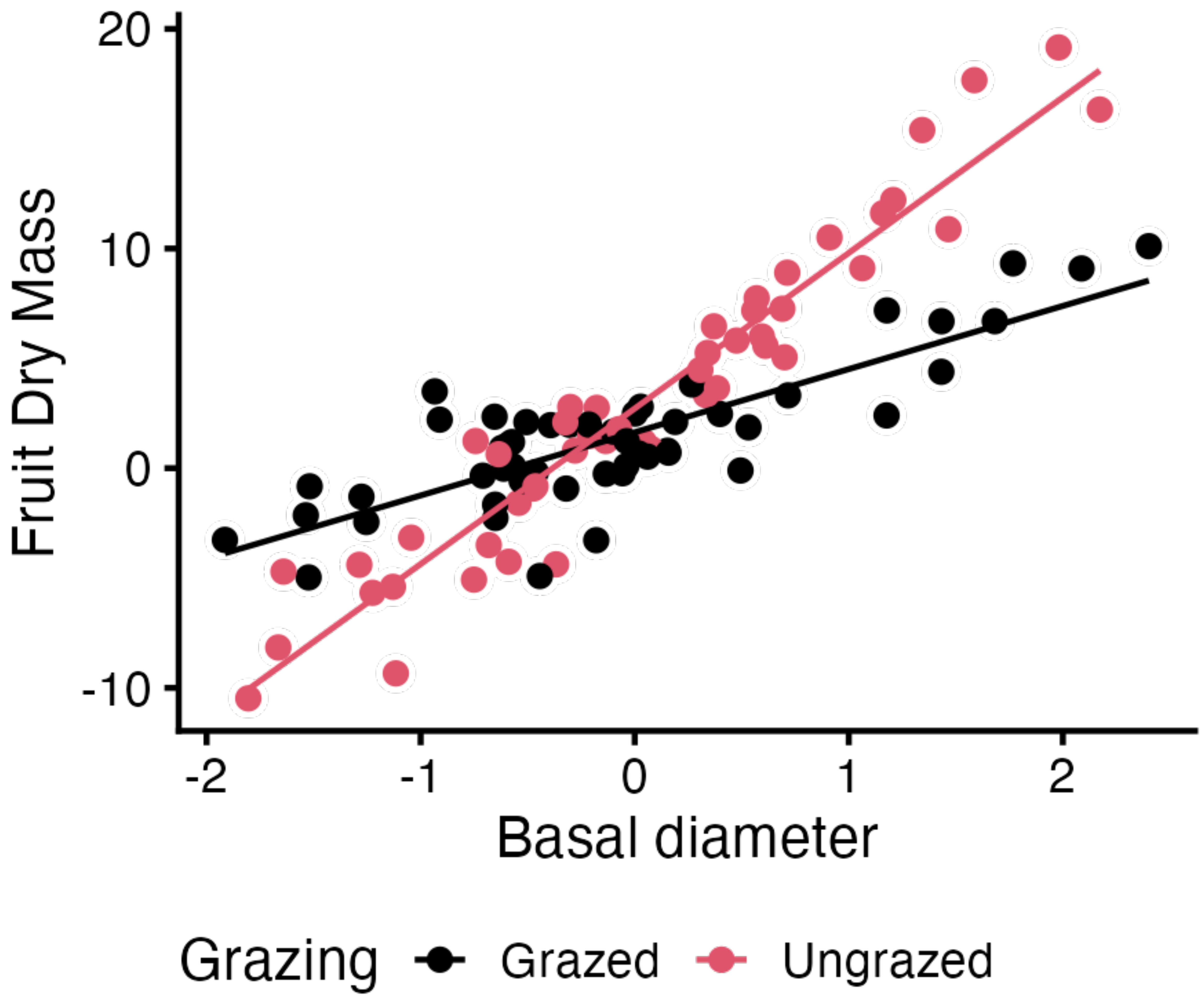
If your model does not work on **simulated** data, it will <u>**never**</u> work on **real** data!

Interactions

Allowing coefficients to change according to other variables

- What if the relation between a predictor and an outcome changes depending on another predictor?
- We can account for this by adding product terms in the model
  - With two predictors, x and z:

$$\mu = \alpha + \beta_1 x + \beta_2 z + \beta_3 xz$$



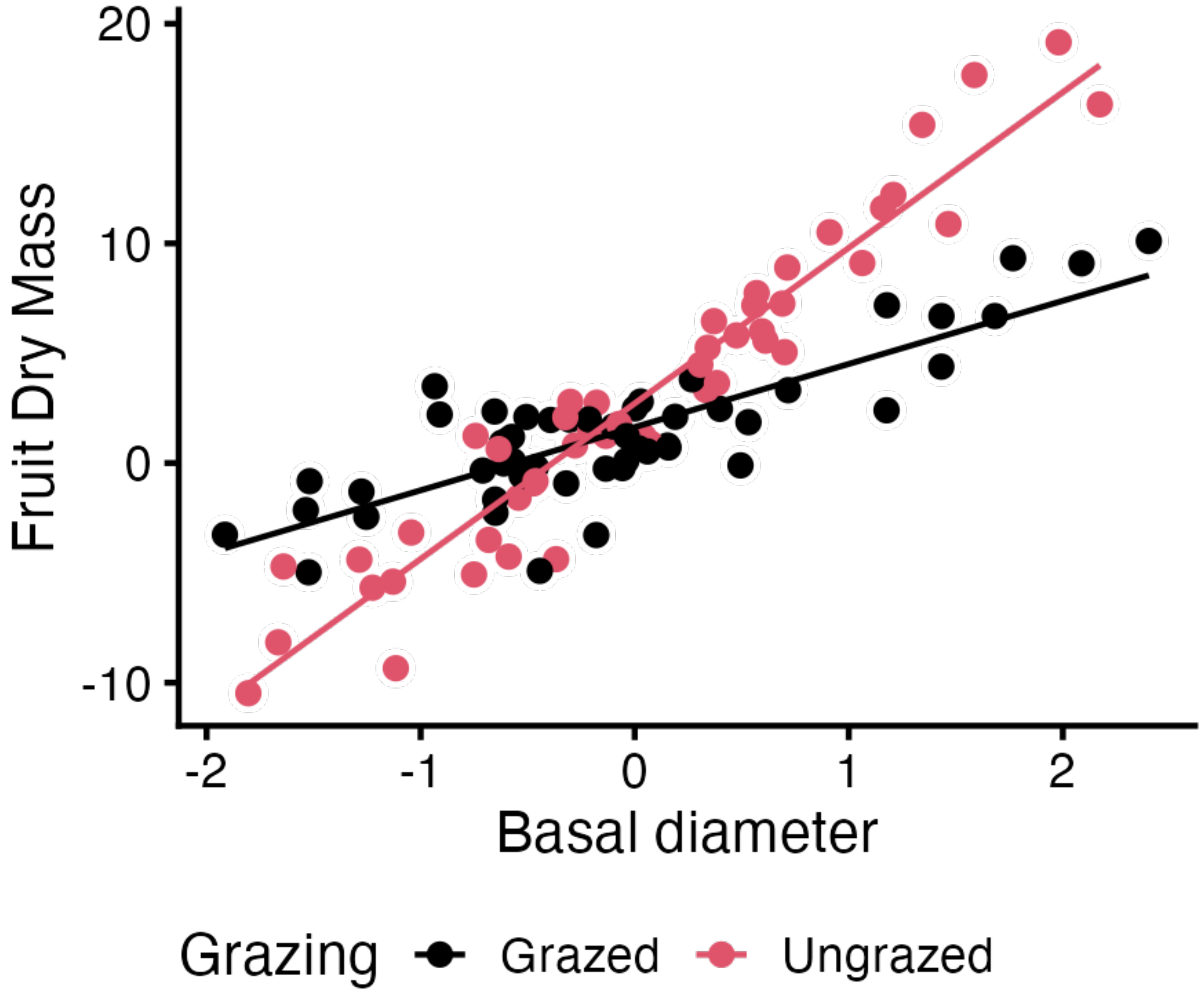
Interactions

Allowing coefficients to change according to other variables

- What if the relation between a predictor and an outcome changes depending on another predictor?
- We can account for this by adding product terms in the model
  - With two predictors, x and z:

$$\mu = \alpha + \beta_1 x + \beta_2 z + \beta_3 xz$$

Interaction term!



Example with interactions

- **Question**: what are the best soil moisture and light availability to grow tulips?

- Greenhouse experiment: beds of tulips kept in nine combinations of soil moisture and shading. Three replicates for each combination (total of 27 beds).

- **Response variable**: mean plant height in each bed

- **Predictor variables**:

    - Water: 3 levels (low, med, high)

    - Shade: 3 levels (low, med, high)

Data from: Grafen & Hails (2002) Modern Statistics for the Life Sciences.

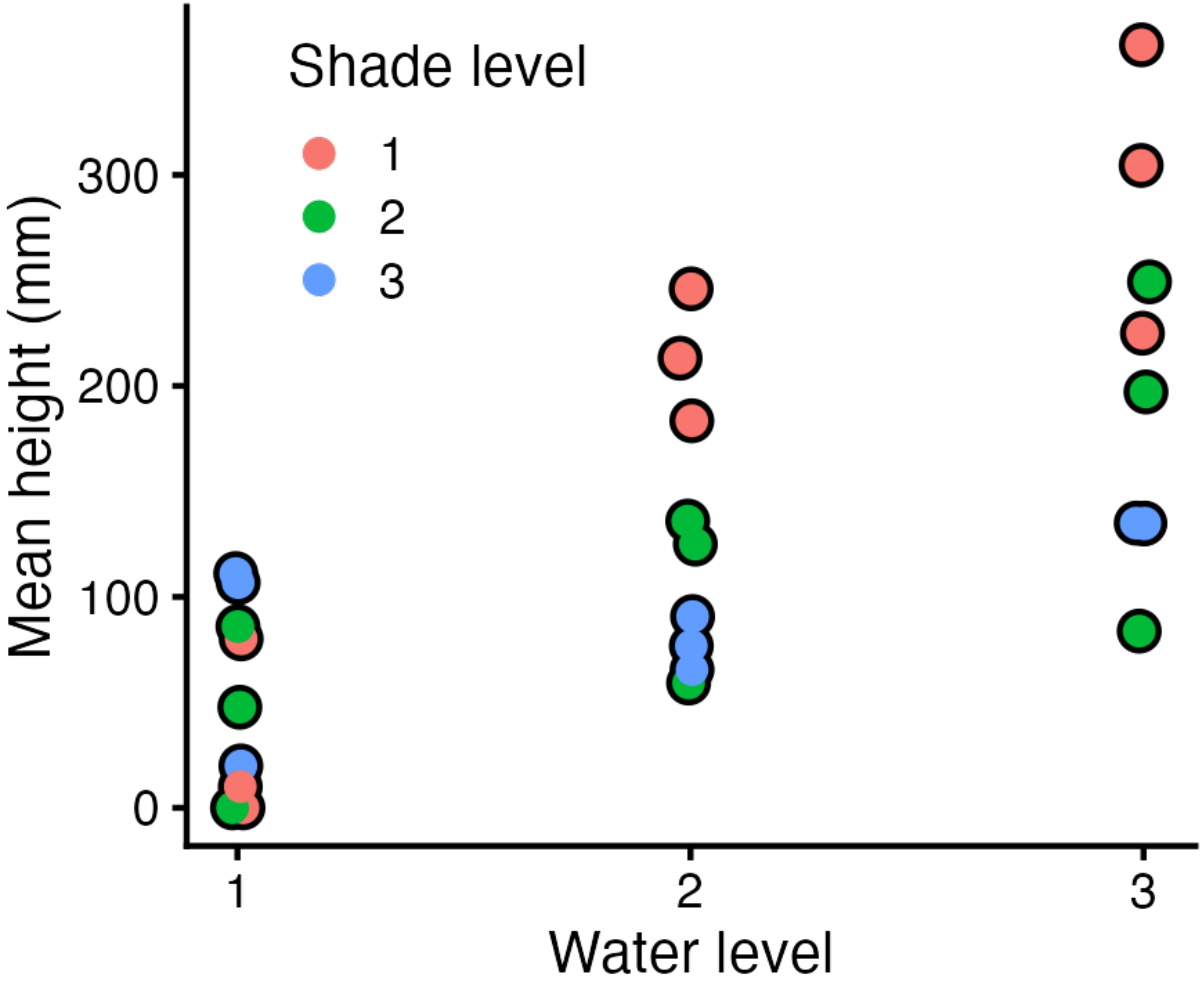
Tulips data

Shade level:

1. Low
2. Medium
3. High

Water level:

1. Low
2. Medium
3. High



Quick transformations

Always think about a scale that makes the model easier to interpret!

Shade level:

-1 : Low

0: Medium

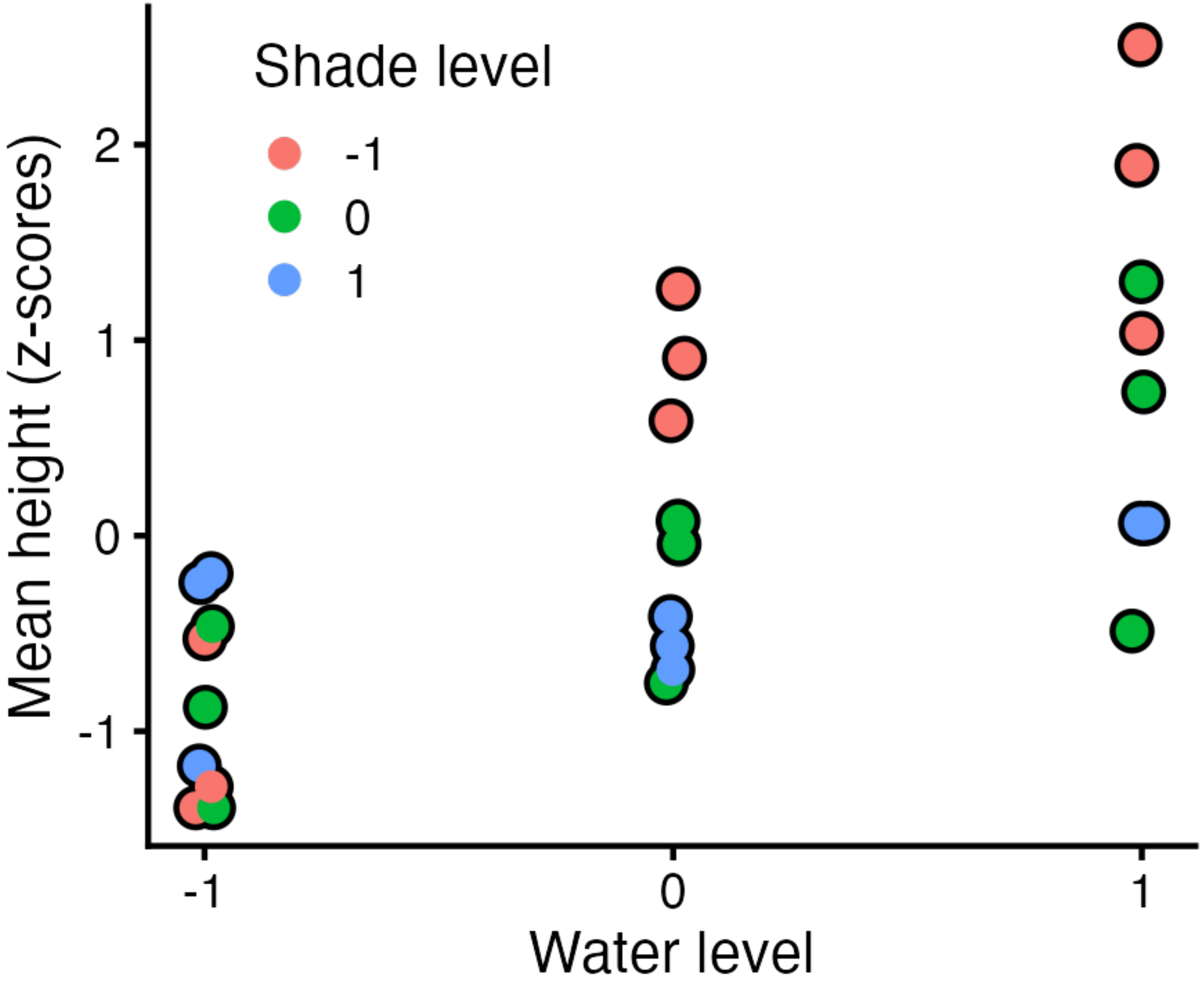
1: High

Water level:

-1 : Low

0: Medium

1: High

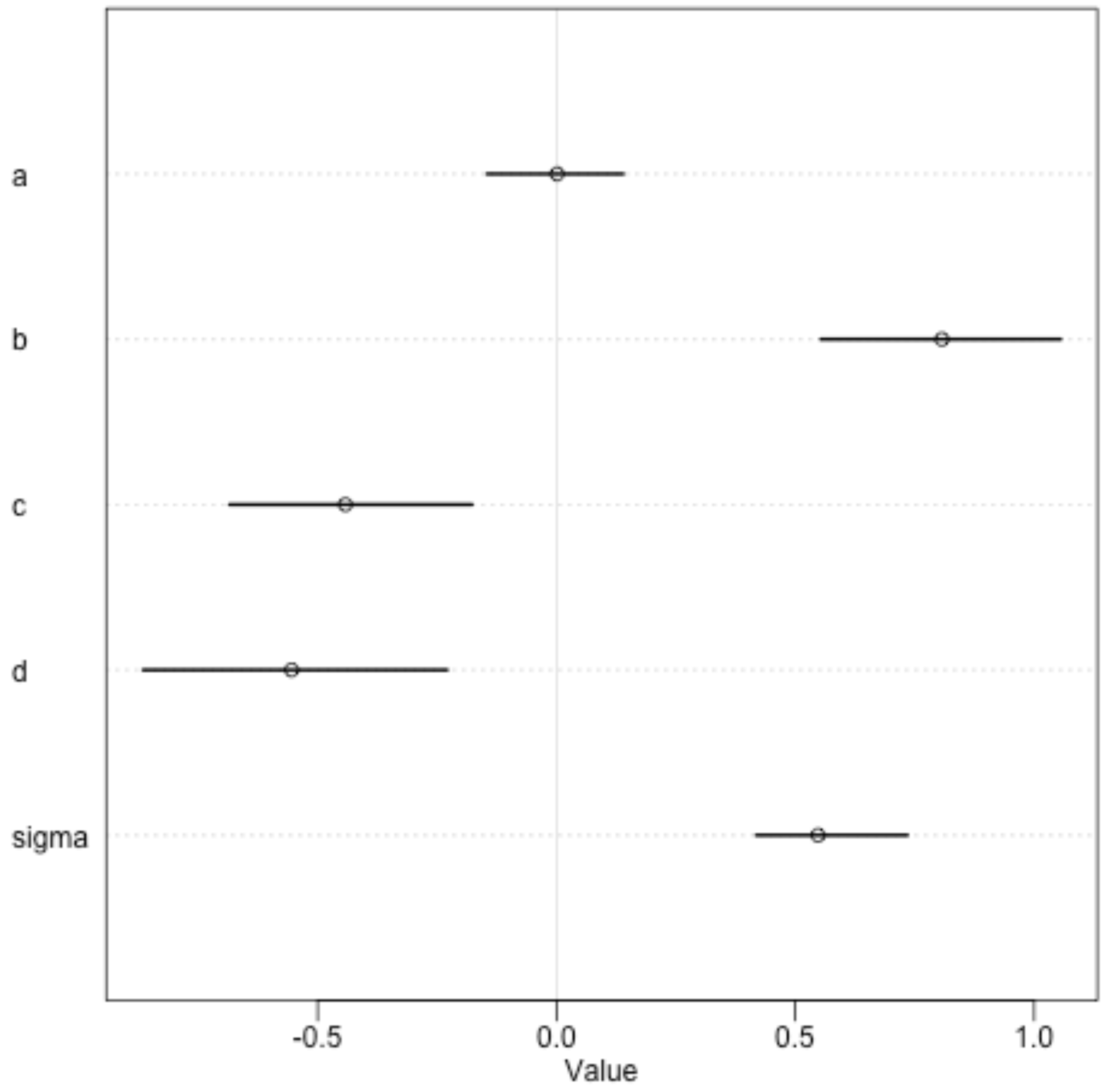


Model with interaction

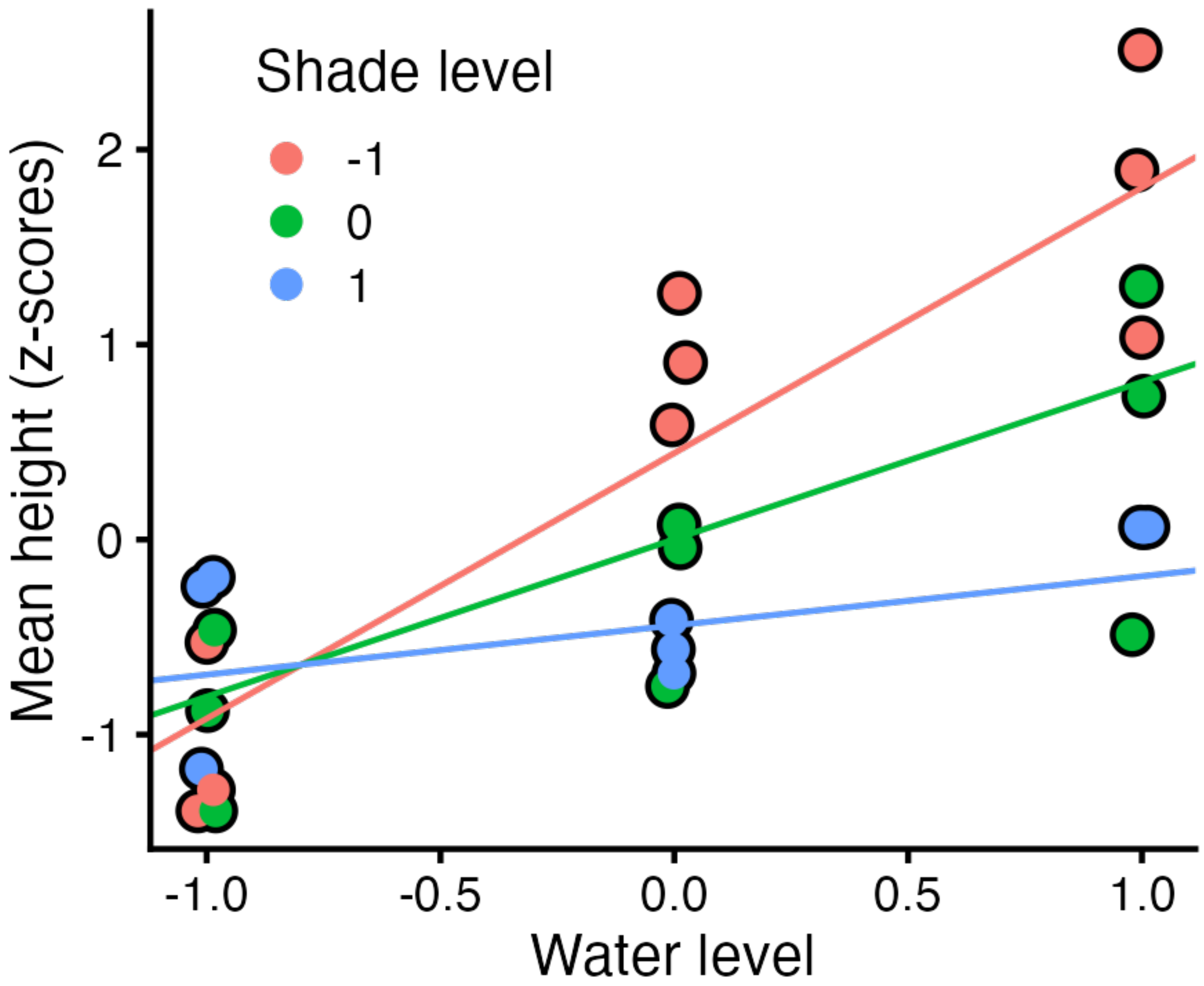
$$Height_i \sim Normal(\mu_i, \sigma)$$

$$\mu_i = a + b\ water_i + c\ \ shade + d\ water_i \times shade_i$$

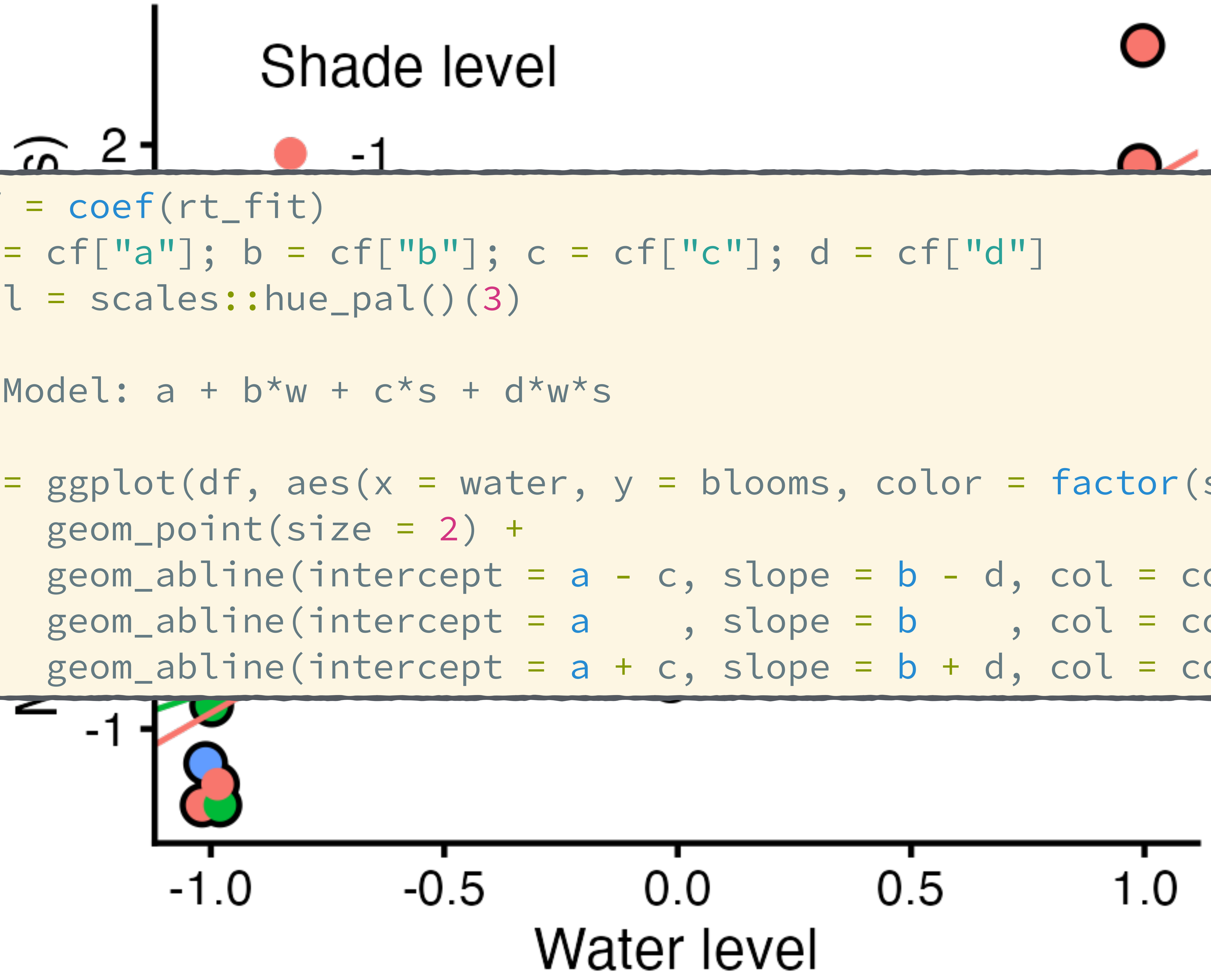
```
data("tulips")
df = tulips
df$water  = scale(df$water, scale = FALSE)
df$shade  = scale(df$shade, scale = FALSE)
df$blooms = scale(df$blooms)

rt_fit = ulam(alist(blooms ~ normal(mu, sigma),
           mu <- a + b*water + c*shade + d*water*shade,
           a ~ normal(0, 0.1),
           b ~ normal(0, 1),
           c ~ normal(0, 1),
           d ~ normal(0, 1),
           sigma ~ exponential(1)),
        data = df, chains = 4, cores = 4)
```



Prediction lines



Prediction lines




```r
cf = coef(rt_fit)
a = cf["a"]; b = cf["b"]; c = cf["c"]; d = cf["d"]
col = scales::hue_pal()(3)

# Model: a + b*w + c*s + d*w*s

p = ggplot(df, aes(x = water, y = blooms, color = factor(shade)) +
    geom_point(size = 2) +
    geom_abline(intercept = a - c, slope = b - d, col = col[1]) +
    geom_abline(intercept = a     , slope = b     , col = col[2]) +
    geom_abline(intercept = a + c, slope = b + d, col = col[3])
```

Summary

- Multiple linear models allow us to use more than one predictor in a linear model
- These models do a form of automatic **stratification**
  - **Ex:** difference in size for individuals of the same age, effect of treatment for individuals of the same size
- The objetive is to compare **like-to-like**

- Coefficients can and do change with the inclusion of more predictors
- Coefficient interpretation is hard, use plots, predictions, scaling and transformations to make models easier to interpret
- Next week, we talk about principled ways of choosing if a variable should be added to a model, stay tuned!