Session #3: Count Models

Guy F. Sutton
Centre for Biological Control
Rhodes University, South Africa
Email: g.sutton@ru.ac.za

Gaussian data is rare

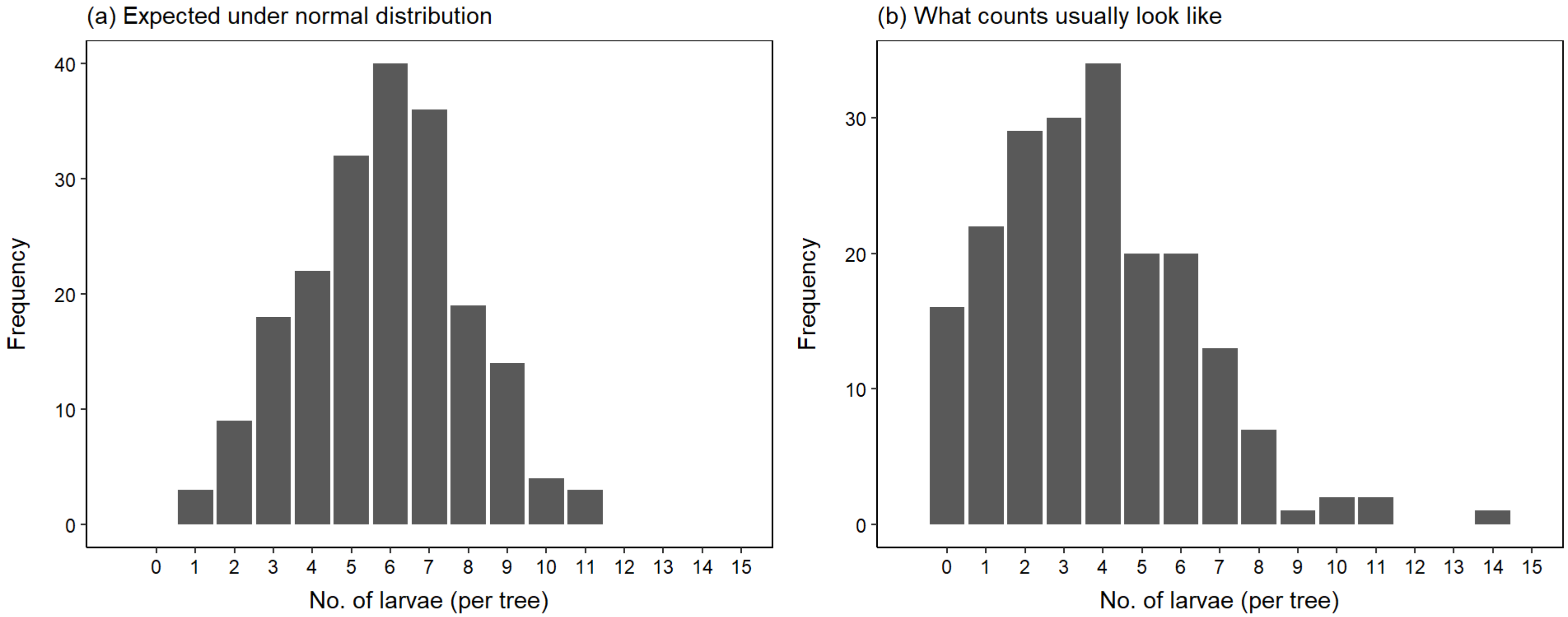
- The analyses so far have assumed that we are dealing with normal (Gaussian) distributed data that can be measured as whole numbers with decimal places (e.g. height, weight, diameter).
- However, most data in the real world either do not conform to the normal distribution, or they cannot be measured as numbers with decimal places, and can’t be analysed with Gaussian models.
    - For example:
        - Counts (e.g. species abundances, days till an event) which are measured as integers (cannot take a decimal place),
        - Binary data (e.g. dead/alive, 1/0, present/absent),
        - Proportion data (e.g. proportion survival, anything measured on a scale of [0,1])
- But, GLM’s can easily be extended to fitting count models and proportion/binary data

Count data

- Count data is ubiquitous in ecology.
  - e.g. No. of FCM per fruit
  - e.g. No. of ticks per zebra
  - e.g. Abundances of impala per hectare in the Kruger National Park
- Counts (or abundances) are defined as non-negative integers
  - I.e. They cannot take a decimal place
  - e.g. 0 FCM per fruit, 7 ticks on a zebra, 385 impala per hectare
    - NOT: 33.15 psyllids per leaf
    - ALSO NOT: 27% of the flies dead

Counts are not normal (literally)

- Counts typically do not conform to assumptions of normality for statistical tests
- Count data typically follows a strong mean-variance relationship
- In many count datasets, there are many zeros and small counts, and successively fewer larger counts



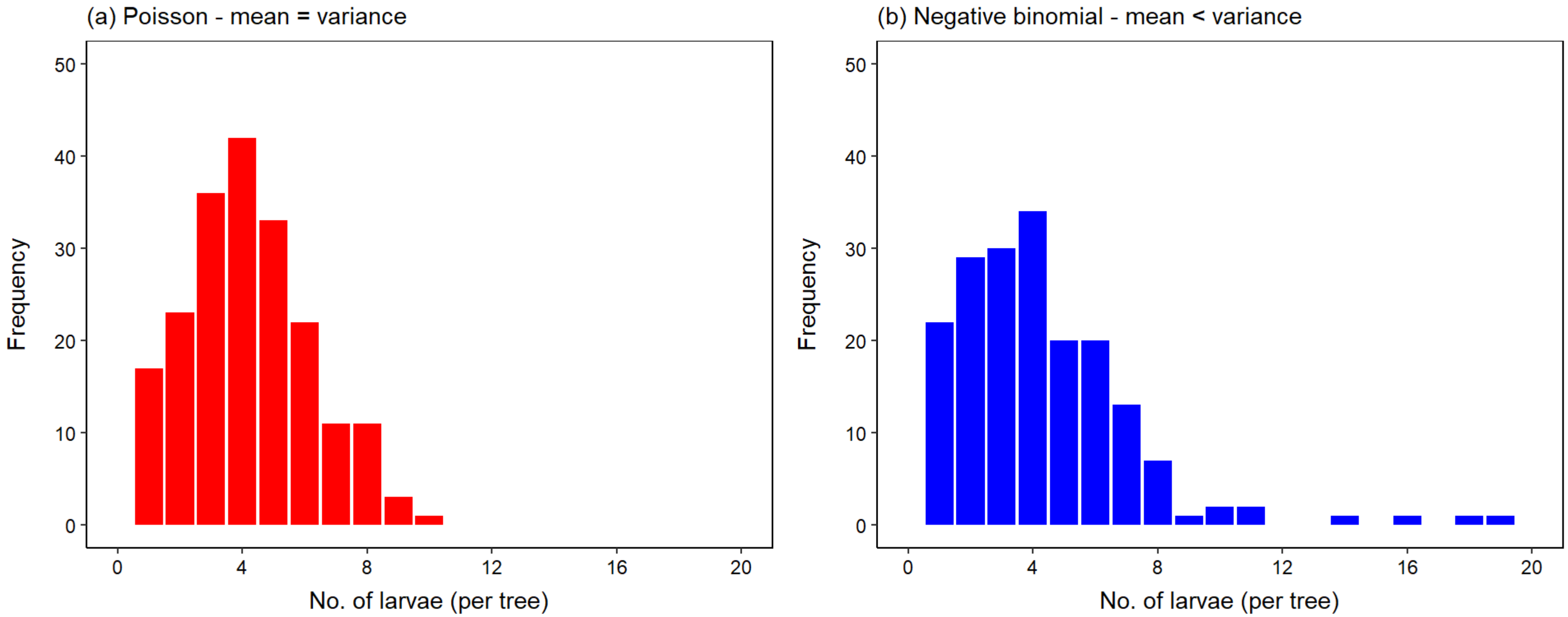
(a) Expected under normal distribution

(b) What counts usually look like

Count models

- There are two basic options for modelling count data:

  1. Poisson GLM - The Poisson distribution assumes the mean = variance.
  2. Negative binomial GLM - The NB distribution assumes the variance > mean



Count models in R

- Assuming our question is: *Does Y vary based on X?*
- We model the log of the expected mean count of Y as a function of X
- *Poisson GLM*:

```
mod_poisson <- glmmTMB::glmmTMB(data = data, family =
poisson(link = "log"), Y ~ X)
```

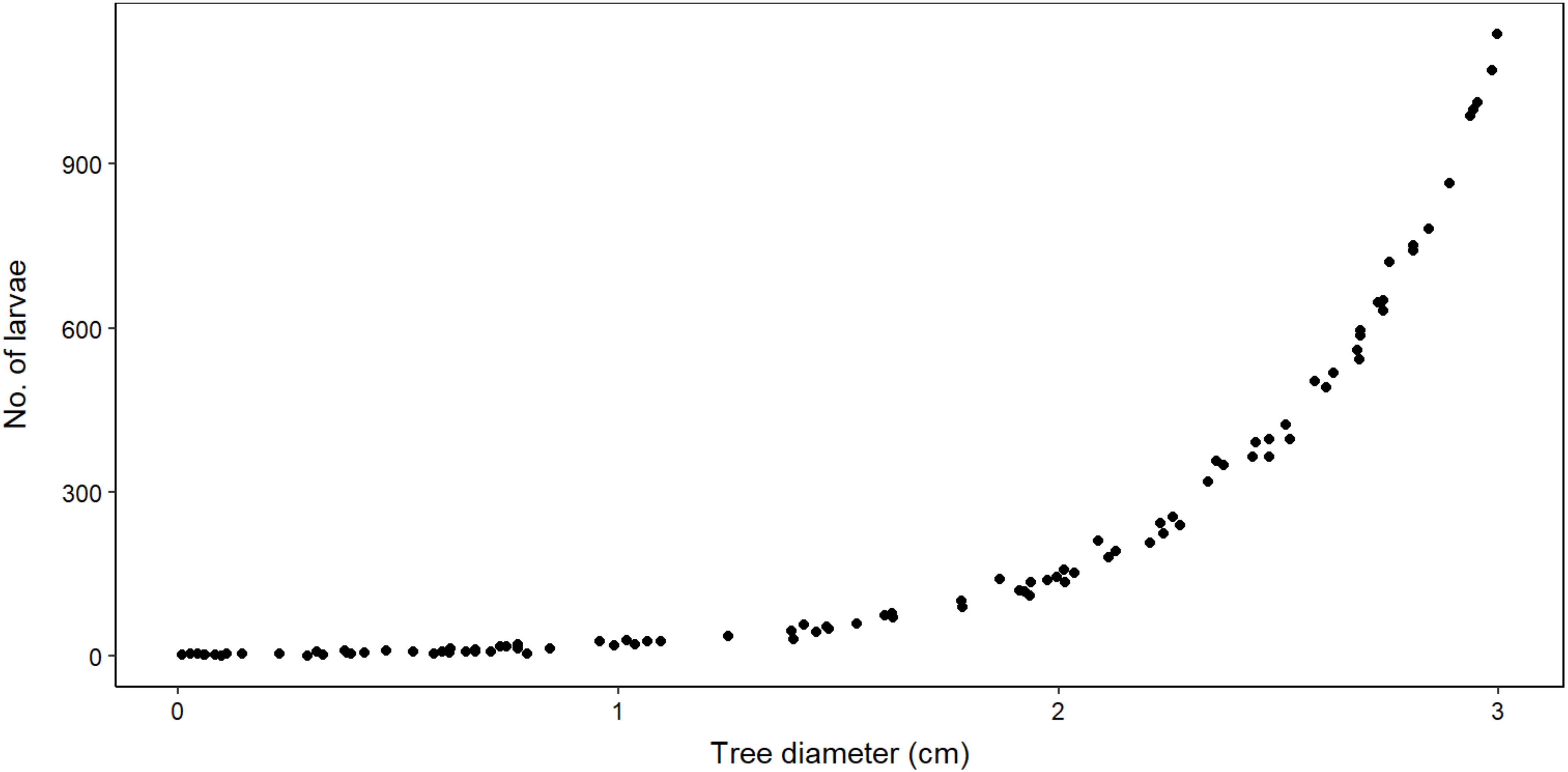
- *Negative binomial GLM*:

```
mod_nb <- glmmTMB::glmmTMB(data = data, family = nbinom2(link =
"log"), Y ~ X)
```

An example

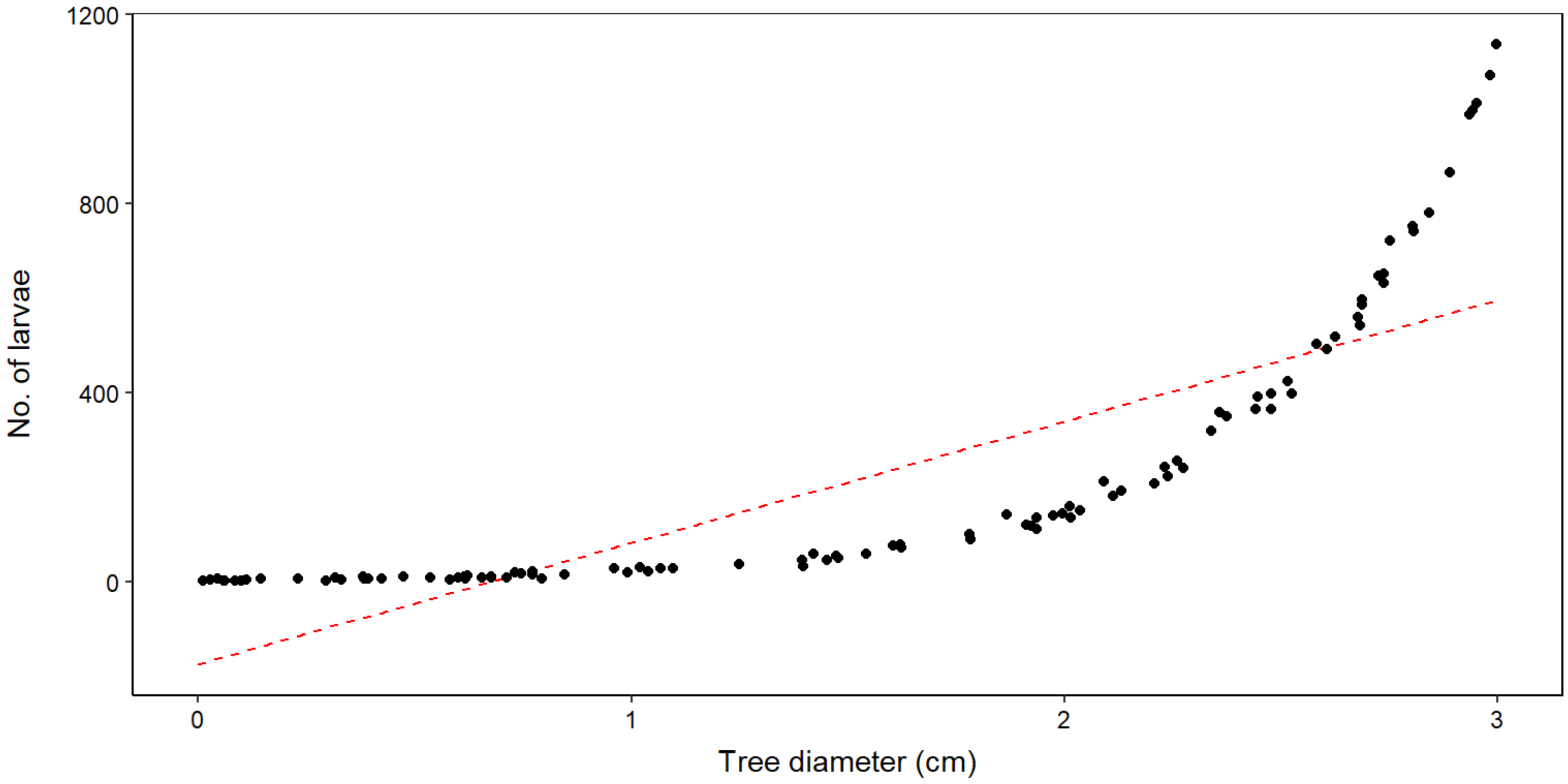
- I have simulated some data representing 100 trees that have been measured (`tree_diam`), and a count of the number of larvae recorded (`no_larvae`).
    - We want to know whether there is a relationship between `tree_diam` and `no_larvae`?
    - I have simulated the data so that:
        - The expected `no_larvae` when `tree_diam` = 0 is 2.71 larvae. Obviously nonsense, but keep this value in mind for later.
        - The `no_larvae` recorded increases by a factor of 7 per 1cm increase in `tree_diam`

Visualise relationship



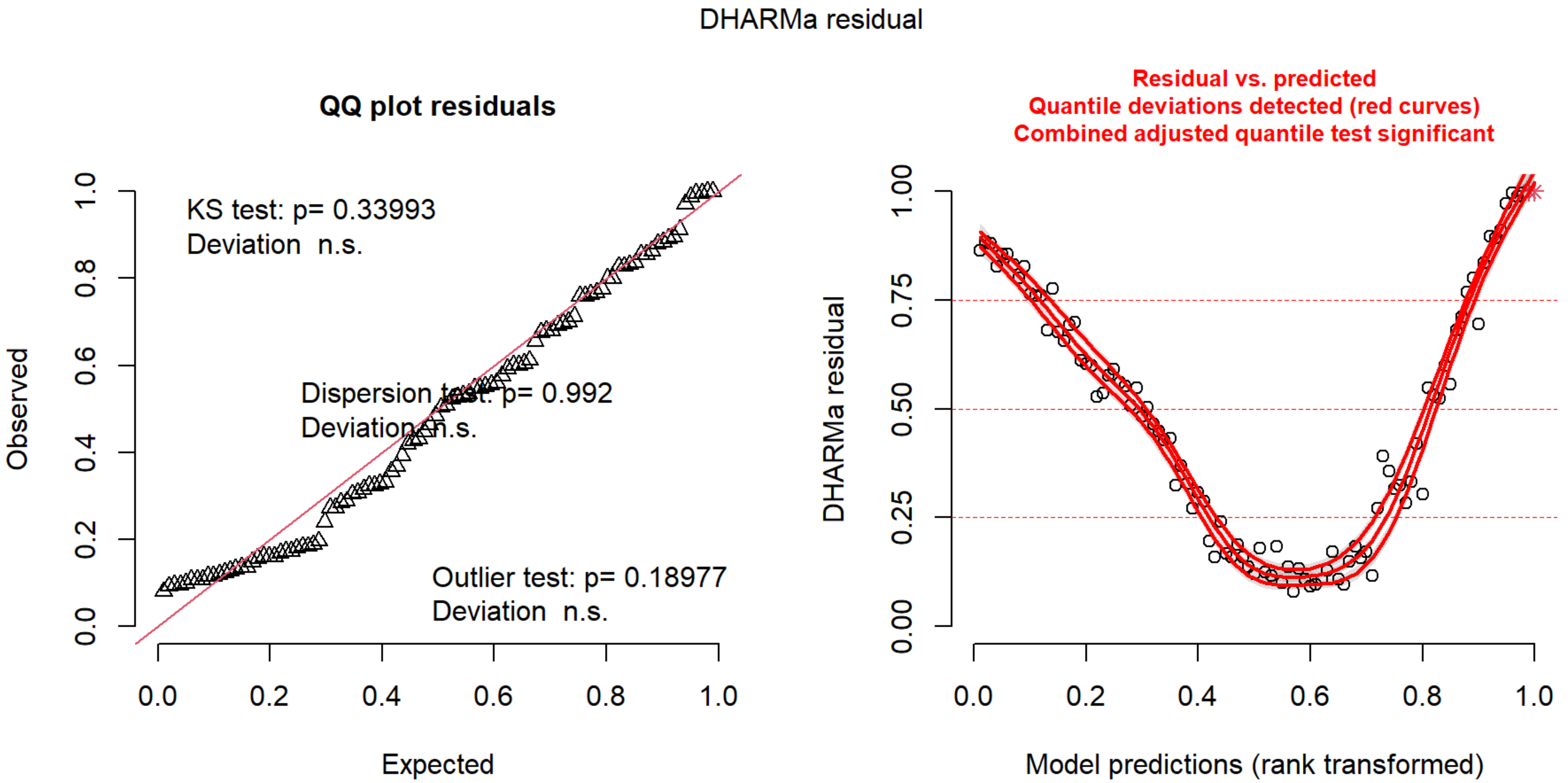
A Gaussian model

```r
# Fit Gaussian model
m_g <- glmmTMB::glmmTMB(
  data = data,
  family = gaussian(link = "identity"),
  no_larvae ~ 1 + tree_diam
)
```



Check model diagnostics - Gaussian model

```r
# Check residuals
sim_out <- DHARMa::simulateResiduals(fittedModel = m_g, plot = T)
```

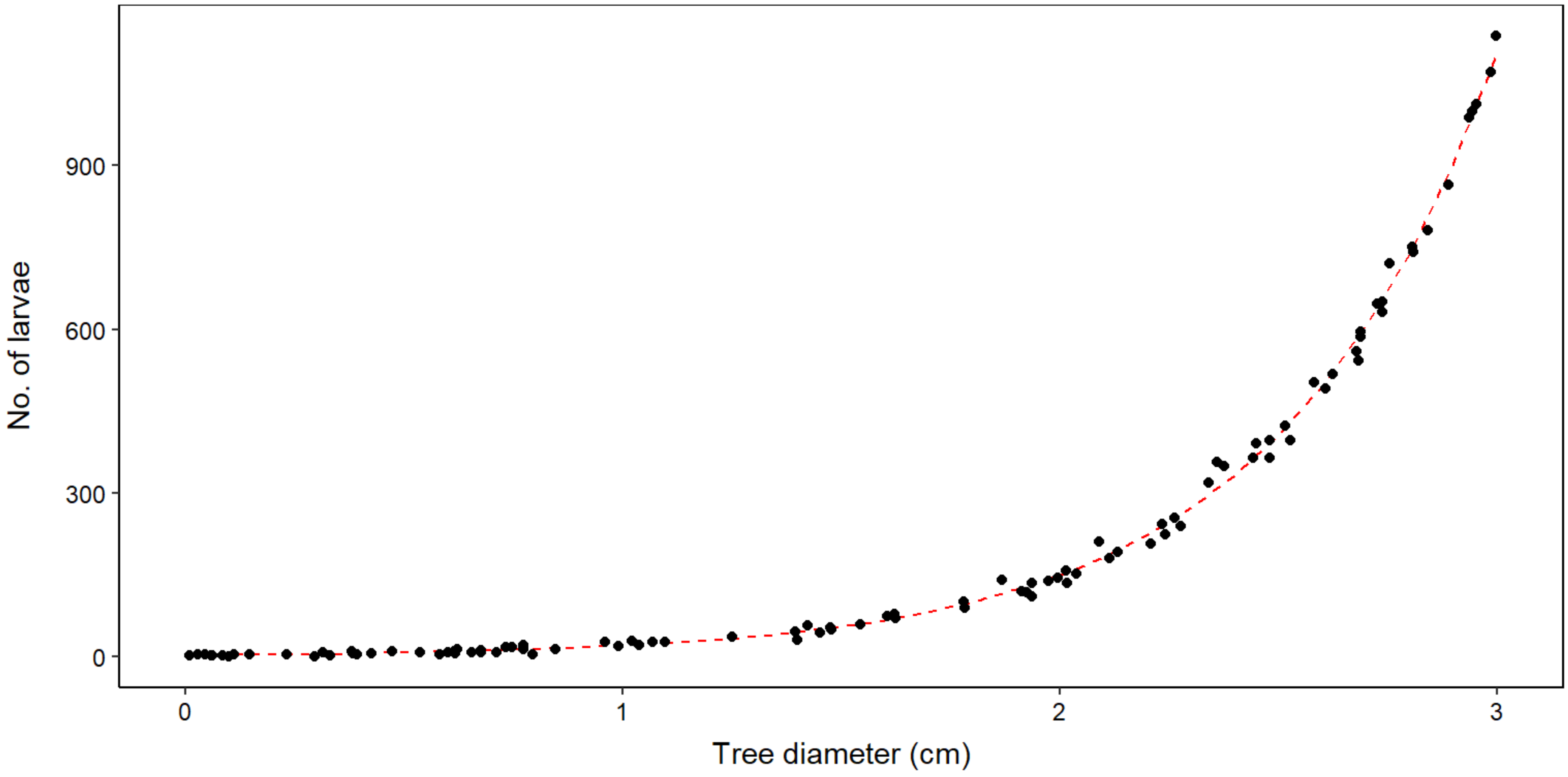


Fit Poisson model

- Here, we model the expected mean number of larvae (`no_larvae`) as a function of `tree_diam`

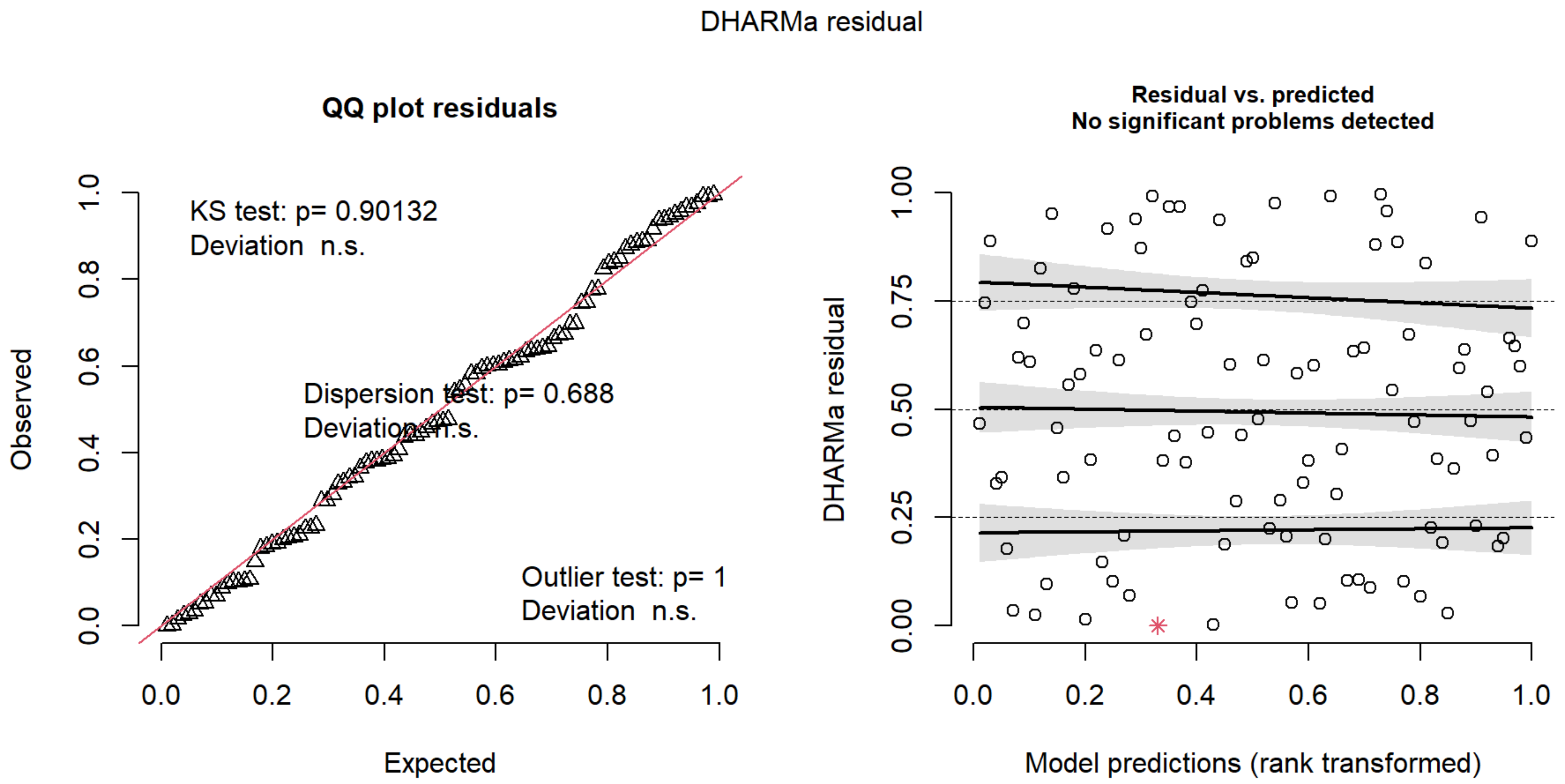
```r
# Fit Poisson model
m_p <- glmmTMB::glmmTMB(
  data = data,
  family = poisson(link = "log"),
  no_larvae ~ 1 + tree_diam
)
```

Plot Poisson prediction



Check model diagnostics - Poisson model

```r
# Check residuals
sim_out <- DHARMa::simulateResiduals(fittedModel = m_p, plot = T)
```



Likelihood Ratio Test

Test the hypothesis of a `tree_diam` effect on the `no_larvae`

```
Likelihood ratio test

Model 1: no_larvae ~ 1
Model 2: no_larvae ~ 1 + tree_diam
  #Df   LogLik Df Chisq Pr(>Chisq)
1   1 -17349.2
2   2   -353.8  1 33991  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There is support for a statistically significant relationship between `tree_diam` and the `no_larvae` counter per tree (X2 = 33991, df = 1, *P* < 0.001).

Interpret coefficients

```r
summary(m_p)
```

```
 Family: poisson  ( log )
Formula:          no_larvae ~ 1 + tree_diam
Data: data

     AIC      BIC   logLik deviance df.resid
   711.6    716.9   -353.8    707.6       98


Conditional model:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.01366    0.04078   24.86   <2e-16 ***
tree_diam    1.99834    0.01559  128.19   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- (Intercept) = 1.0136
  - Always have to exponentiate (`exp(value)`) to get interpretable coefficients
  - `exp(1.0136)` = 2.74
  - The expected `no_larvae` when `tree_diam` = 0 is 2.74 larvae.

Interpret coefficients

```r
summary(m_p)
```

```
 Family: poisson  ( log )
Formula:          no_larvae ~ 1 + tree_diam
Data: data

     AIC      BIC   logLik deviance df.resid
   711.6    716.9   -353.8    707.6       98


Conditional model:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)  1.01366    0.04078   24.86   <2e-16 ***
tree_diam    1.99834    0.01559  128.19   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

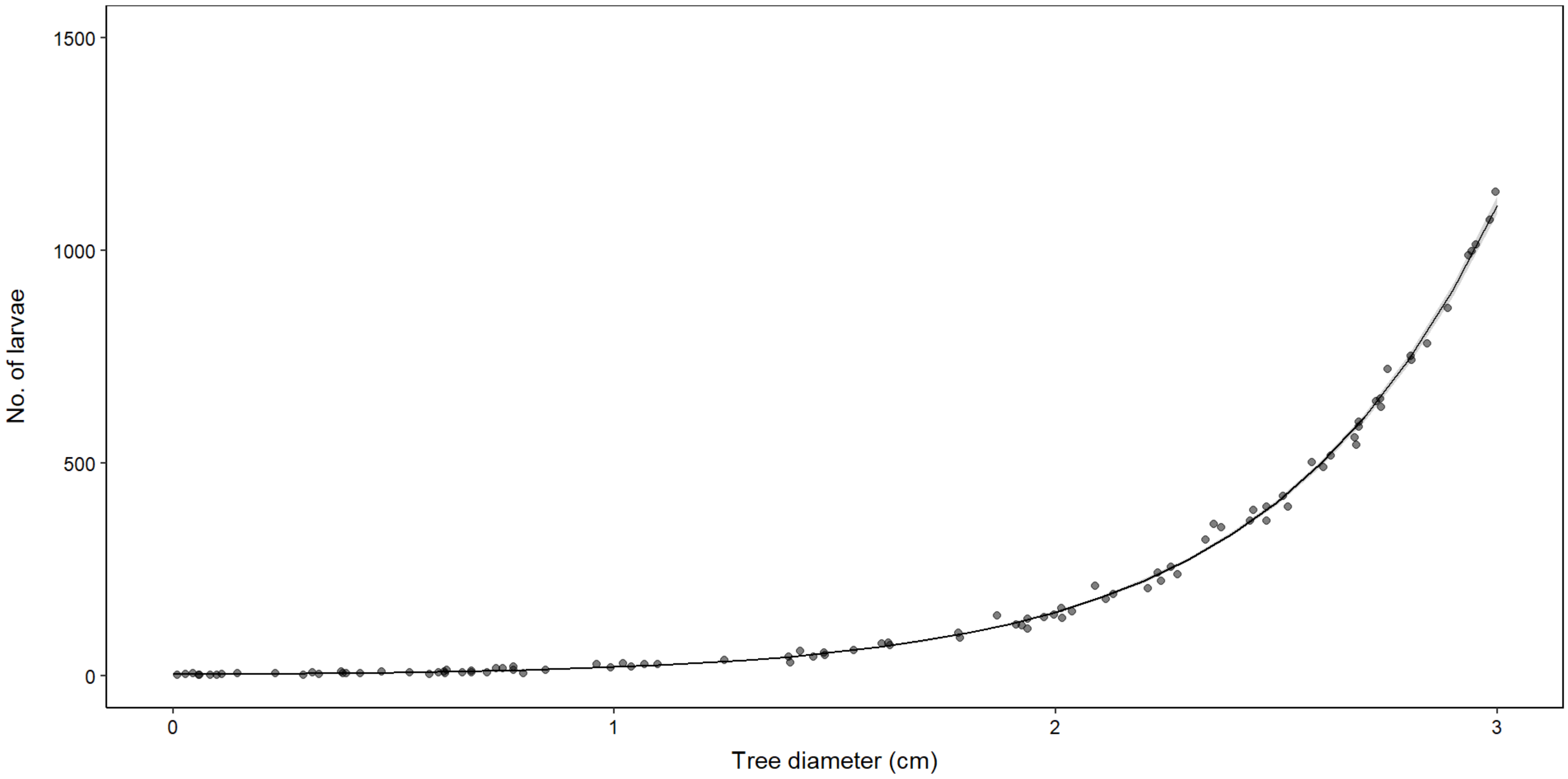
Extracting marginal means

```r
# Extract expected relationship between X and Y
preds <- ggeffects::ggeffect(
  model = m_p,
  terms = c("tree_diam [0:3 by = 0.1]"),
  type = "fixed",
  interval = "confidence"
  ) %>%
  # Convert predictions into a data.frame
  as.data.frame() %>%
  # Rename columns for easier plotting
  dplyr::mutate(
    tree_diam = x
  )
```

Plot marginal effect plot

Plot

Code



Fit negative binomial model

- Here, we model the expected mean number of larvae (`no_larvae`) as a function of `tree_diam`

```r
# Fit NB model
m_nb <- glmmTMB::glmmTMB(
  data = data,
  family = nbinom2(link = "log"),
  no_larvae ~ 1 + tree_diam
)
```

Is Poisson or NB better for me?

The typical approach is to fit both Poisson and NB models, compare fits, and then perform inference (if at least one of these models is suitable).

We can compare the two models using a **Wald Test**:

```r
anova(m_p, m_nb)
```

```
Data: data
Models:
m_p: no_larvae ~ 1 + tree_diam, zi=~0, disp=~1
m_nb: no_larvae ~ 1 + tree_diam, zi=~0, disp=~1
     Df     AIC     BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
m_p   2 711.65 716.86 -353.82   707.65
m_nb  3 713.65 721.47 -353.82   707.65 1e-04      1     0.9926
```

Compare Poisson vs NB using Wald Test

We can compare the two models using a **Wald Test**:

```r
anova(m_p, m_nb)
```

```
Data: data
Models:
m_p: no_larvae ~ 1 + tree_diam, zi=~0, disp=~1
m_nb: no_larvae ~ 1 + tree_diam, zi=~0, disp=~1
     Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
m_p   2 711.65 716.86 -353.82   707.65
m_nb  3 713.65 721.47 -353.82   707.65 1e-04      1     0.9926
```

There is no evidence that the negative binomial model provided a better fit to the data than the Poisson model (X2 = 0.0001, df = 1, *P* = 0.993). As such, you use the simpler Poisson model (assuming the model diagnostics are suitable).

Why not take the log?

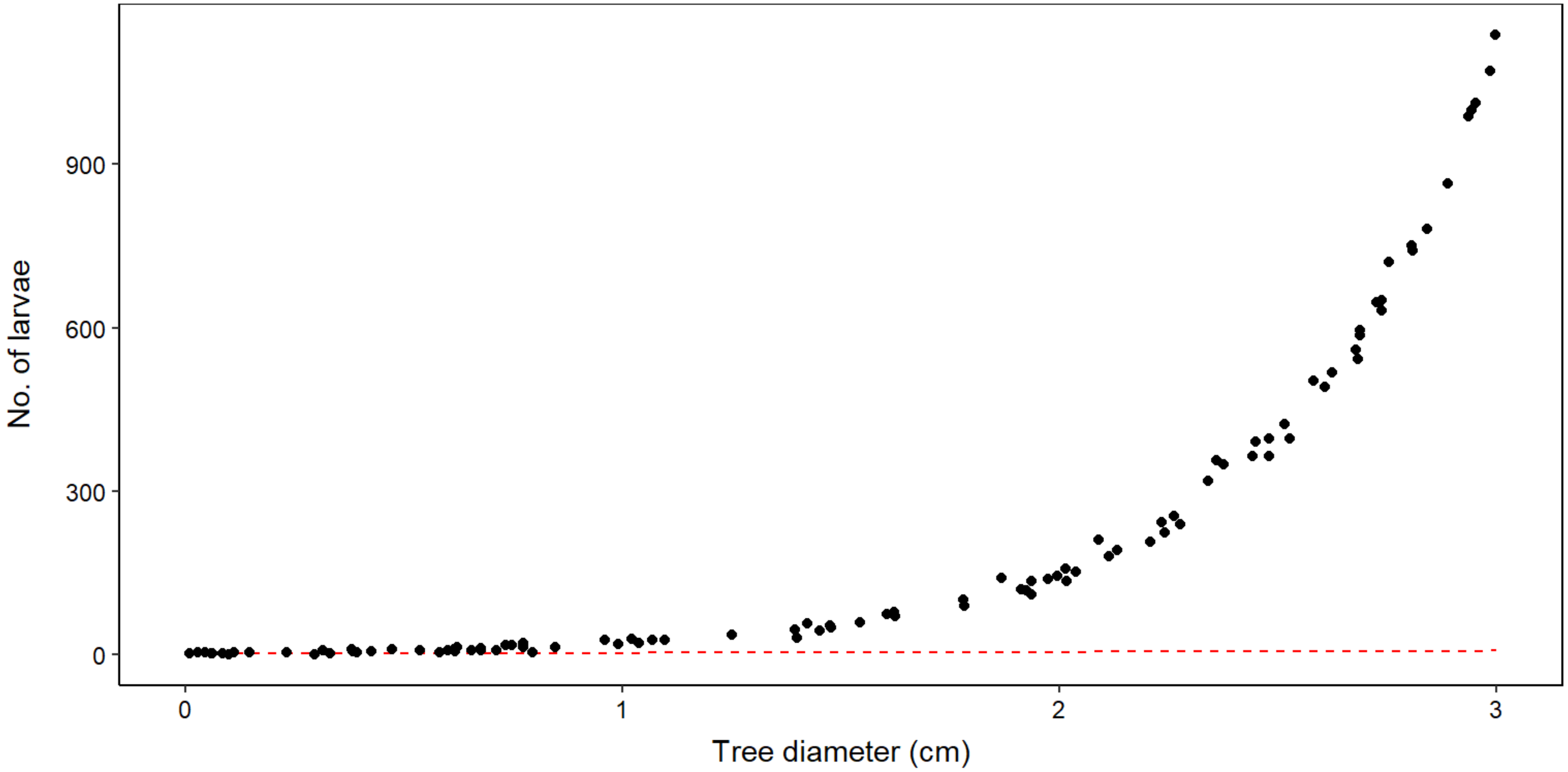
In the above models, we are modelling the log of the expected counts. So, why not just take the log of the response variable?

```r
# Fit log-linear model
m_l <- glmmTMB::glmmTMB(
  data = data,
  family = gaussian(link = "identity"),
  log(no_larvae + 1) ~ 1 + tree_diam
)
```

It’s all about variances

This will not work, as the log-linear model assumes variance is proportional to squared expected values, while the poisson/NB assume variance is equal or greater than the expected mean.

As such, log-linear model will underestimate the expected counts.


No. of larvae
Tree diameter (cm)