Bee_analysis

Andre Bourbonnais

2023-11-24

Bee exercise: Linear regression

The following dataset includes the abundance of the bee species *Eulaema nigrita* in the Brazilian Atlantic forest, and a number of potential predictor variables.

- **MAT**: Mean annual temperature (°C)
- **MAP**: Mean annual precipitation (mm)
- **Tseason**: Temperature seasonality (coefficient of variation)
- **Pseason**: Precipitation seasonality (coefficient of variation)
- **forest.**: Proportion forest cover in the landscape
- **lu_het**: Land use heterogeneity (Shannon diversity of local land-use classes)

Use a GLM to build a model explaining the distribution patterns of *Eulaema nigrita*. Interpret the results and produce nice tables and figures.

Loack packages and read in data

```
library(MASS)
library(tidyverse)
library(flexplot)
library(ggpubr)
library(RColorBrewer)
```

Glimpse of the data

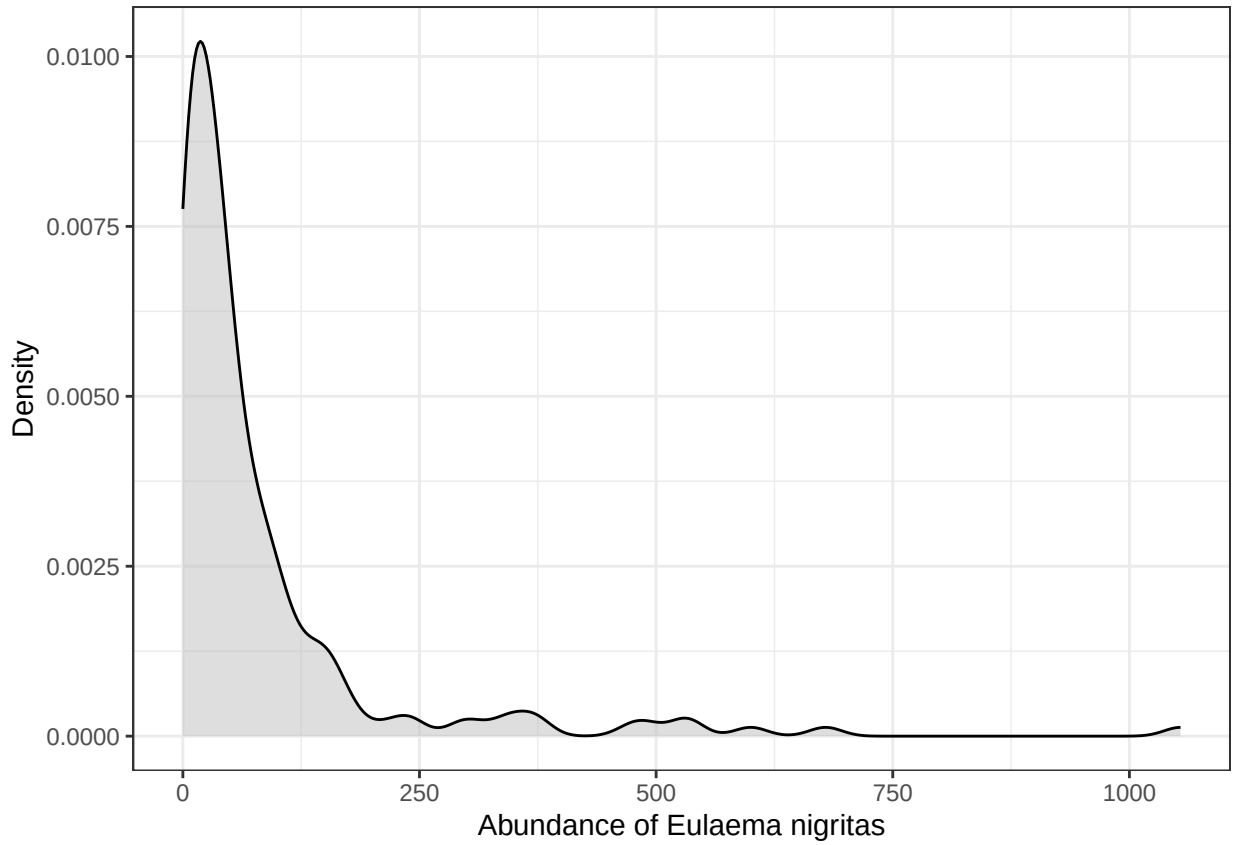
```
# Read in data
dat = read.csv("../00_DATA/Eulaema.csv")

# Inspect data
glimpse(dat)
```

```
## Rows: 178
## Columns: 10
## $ Eulaema_nigrita <int> 492, 372, 679, 600, 28, 535, 100, 100, 66, 100, 143, 2~
## $ method          <chr> "NetTraps", "Traps", "Traps", "Traps", "Net", "Net", "~
## $ effort          <dbl> 4.189655, 5.204007, 5.204007, 5.204007, 4.969813, 4.96~
## $ altitude        <int> 6, 17, 17, 30, 0, 43, 24, 24, 24, 24, 24, 24, 24, 24, ~
## $ MAT             <int> 235, 230, 230, 231, 259, 255, 232, 232, 232, 232, 232,~
## $ MAP             <int> 1073, 987, 987, 1030, 1693, 1697, 1127, 1127, 1127, 11~
## $ Tseason         <int> 2036, 1760, 1760, 1820, 1074, 1061, 2209, 2209, 2209, ~
## $ Pseason         <int> 53, 49, 49, 51, 62, 63, 57, 57, 57, 57, 57, 57, 57, 57~
## $ forest.         <dbl> 0.04416404, 0.18217054, 0.18217054, 0.01577287, 0.0500~
## $ lu_het          <dbl> 1.0531299, 0.7571063, 0.7571063, 0.8277580, 0.9672604,~
```

```r
# Lets factor the methods
dat$method <- as.factor(dat$method)

# Looks good
```

Exploring the data

```r
# Plot the Eulaema nigrita as density plot
dat %>%
  ggplot(aes(x = Eulaema_nigrita)) +
  geom_density(fill = "grey", alpha = 0.5) +
  labs(x = "Abundance of Eulaema nigritas", y = "Density") +
  theme_bw()
```



The plot is indicating that the data is following a *Poisson distribution*. Lets explore the mean to variance relation to determine if we need to use *bionomial* or *quasi-poisson* distribution to fit a model.

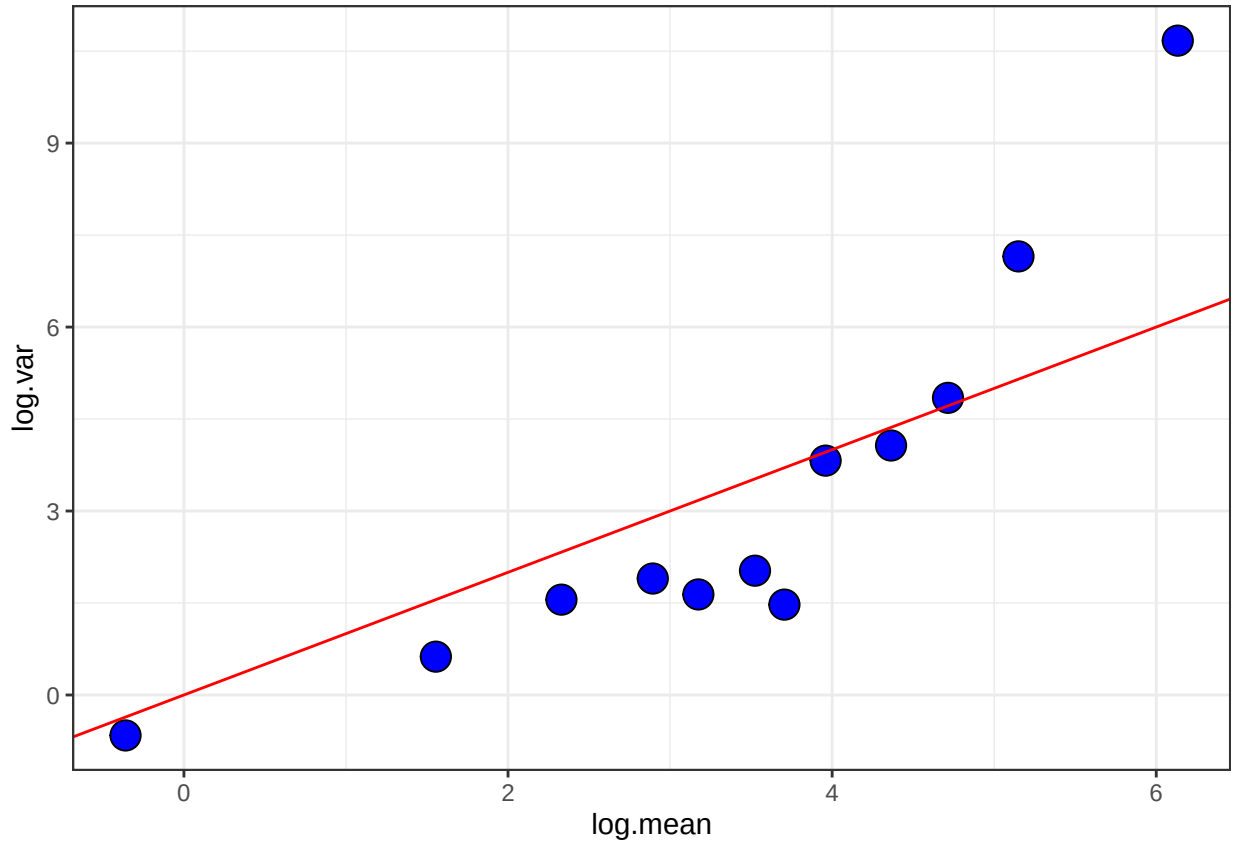
I will split up the data into bins and take the mean and variance for each bin to plot against.

```
# Generating variance vs mean plot
nrow(dat) # 178
```

```
## [1] 178
```

```
# Number of counted cases
nrows <- 1:nrow(dat)

# Sorted E. nigritas
E_nig_sorted <- sort(dat$Eulaema_nigrita)

# Number of bins
num_bins <- 12

# Calculate bin size
bin_size <- ceiling(length(E_nig_sorted) / num_bins)

# Initialize vectors for variance and mean
vars <- numeric(num_bins)
means <- numeric(num_bins)
```

```r
# Bin the cases and calculate variance and mean for each bin
for (i in 1:num_bins) {
  start_index <- (i - 1) * bin_size + 1
  end_index <- min(i * bin_size, length(E_nig_sorted))
  bin <- E_nig_sorted[start_index:end_index]
  vars[i] <- var(bin)
  means[i] <- mean(bin)
}

# Generate variance/mean dataframe
df_bins <- data.frame(var = vars, mean = means)

# Plotting the variance against mean
df_bins %>%
  mutate(log.var = log(var)) %>%
  mutate(log.mean = log(mean)) %>%
  ggplot(aes(log.mean, log.var)) +
  geom_jitter(size=5, shape = 21, fill = "blue") +
  geom_abline(intercept = 0, slope = 1, color = "red") +
  theme_bw()
```



The plot shows a clear overdispersion, especially at larger mean values. To be able to handle the tail, a *quasi-poisson* model will not work. Here, a *binomial* model will work better.

Here we see how the variance is not equal to the mean.

```r
# Checking the mean and var for all the data
var(E_nig_sorted); mean(E_nig_sorted)
```

```
## [1] 18549.07
```

```
## [1] 81.14045
```

With the large overdispersion there is some underlying factor that is generating this variance. Lets check if the methods maybe act on variance

```r
# Count number of methods
unique(dat$method)    # "NetTraps" "Traps"    "Net"
```

```
## [1] NetTraps Traps    Net
## Levels: Net NetTraps Traps
```

```r
sum(dat$method == "NetTraps")  # 29
```

```
## [1] 29
```

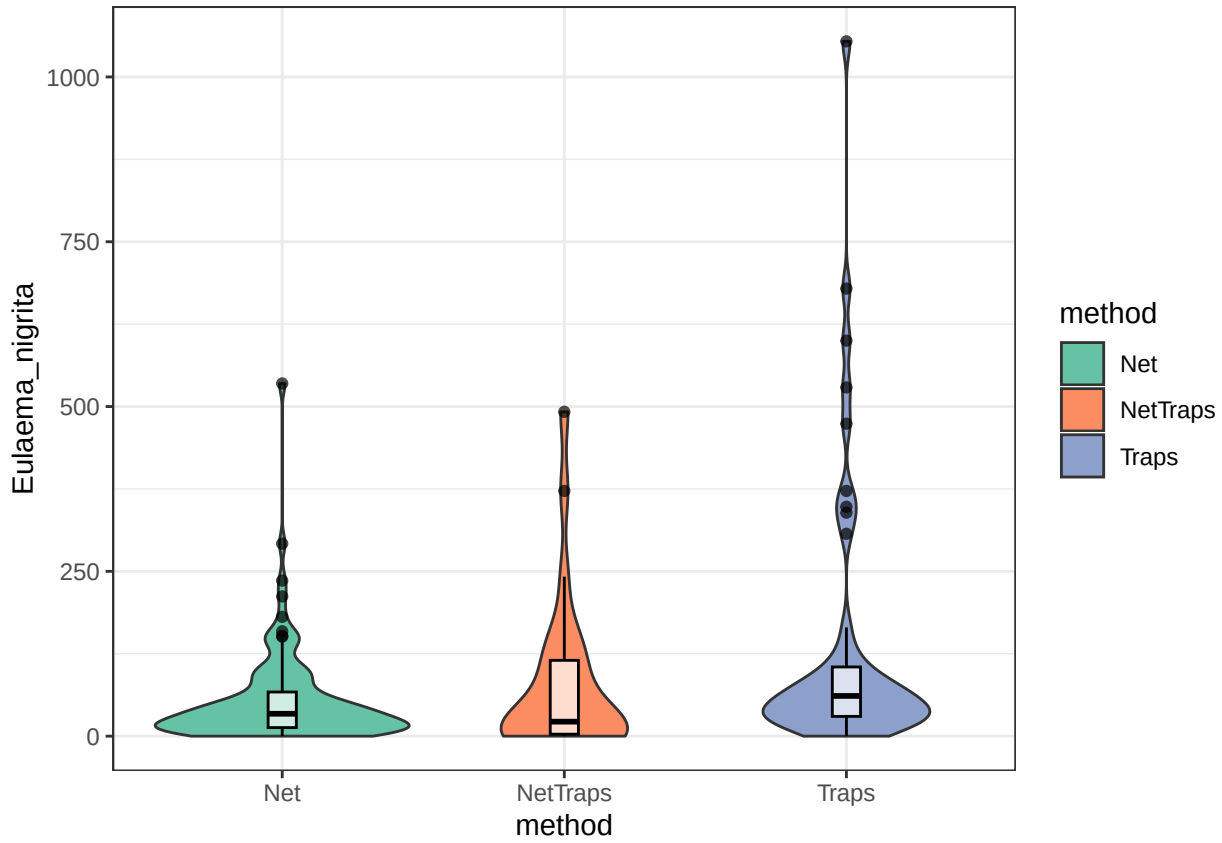
```r
sum(dat$method == "Net")    # 101
```

```
## [1] 101
```

```r
sum(dat$method == "Traps")  # 48
```

```
## [1] 48
```

```r
# Violin plot to see how the distribution looks like based on method.
dat %>%
  ggplot(aes(method, Eulaema_nigrita, fill = method)) +
  geom_violin() +
  geom_boxplot(width = 0.1, color = "black", fill = "white",
               alpha = 0.7) +
  scale_fill_brewer(palette="Set2") +
  theme_bw()
```



There are definitely a few outliers, specially one in **Traps**.

Lets fit a model

As expected, we see a very high null deviance due to the overdispersion!

```
# Analysis
# Generate a complex model with all the variables
m_all <- glm(Eulaema_nigrita ~ ., data = dat, family = "poisson")
summary(m_all)  # Very high null deviance as we expected overdispersion!
```

```
##
## Call:
## glm(formula = Eulaema_nigrita ~ ., family = "poisson", data = dat)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.910e+01  3.987e-01  47.899  < 2e-16 ***
## methodNetTraps -5.366e-01  3.397e-02 -15.798  < 2e-16 ***
## methodTraps    -5.663e-01  3.023e-02 -18.734  < 2e-16 ***
## effort          4.714e-01  7.729e-03  60.996  < 2e-16 ***
## altitude       -2.789e-03  7.943e-05 -35.105  < 2e-16 ***
## MAT            -5.260e-02  1.437e-03 -36.596  < 2e-16 ***
## MAP            -1.506e-03  3.608e-05 -41.732  < 2e-16 ***
## Tseason        -1.443e-03  3.460e-05 -41.714  < 2e-16 ***
```

```
## Pseason         2.090e-02  6.250e-04  33.431  < 2e-16 ***
## forest.        -1.165e+00  3.819e-02 -30.504  < 2e-16 ***
## lu_het         -8.346e-02  3.092e-02  -2.699  0.00695 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 23280  on 177  degrees of freedom
## Residual deviance: 11295  on 167  degrees of freedom
## AIC: 12241
##
## Number of Fisher Scoring iterations: 6
```

Thus, we do a negative binomial regression

```
# Thus we do a negative binomial regression
nb_all <- glm.nb(Eulaema_nigrita ~ ., data = dat)
summary(nb_all) # Much better!
```

```
##
## Call:
## glm.nb(formula = Eulaema_nigrita ~ ., data = dat, init.theta = 1.125447701,
##     link = log)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    13.9840806  3.4268681   4.081 4.49e-05 ***
## methodNetTraps -0.2784872  0.2651483  -1.050 0.293577
## methodTraps    -0.5170301  0.2389399  -2.164 0.030476 *
## effort          0.4776507  0.0710479   6.723 1.78e-11 ***
## altitude       -0.0016500  0.0006958  -2.371 0.017721 *
## MAT            -0.0320050  0.0124180  -2.577 0.009957 **
## MAP            -0.0015272  0.0002255  -6.772 1.27e-11 ***
## Tseason        -0.0013532  0.0002512  -5.386 7.19e-08 ***
## Pseason         0.0194377  0.0042750   4.547 5.45e-06 ***
## forest.        -1.1685866  0.3143283  -3.718 0.000201 ***
## lu_het         -0.1059110  0.2495819  -0.424 0.671308
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.1254) family taken to be 1)
##
##     Null deviance: 393.88  on 177  degrees of freedom
## Residual deviance: 204.16  on 167  degrees of freedom
## AIC: 1776.8
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.125
##           Std. Err.:  0.116
##
##  2 x log-likelihood:  -1752.765
```

Lets see if we can remove some of the variables that are not significant. I see that `lu_het` is not significant as it is less than 2 SD from the mean. As well as for `methodNetTraps` however, i will keep it in the model for now.

```
nb_imp1 <- glm.nb(Eulaema_nigrita ~ . - lu_het, data = dat)
summary(nb_imp1)
```

```
##
## Call:
## glm.nb(formula = Eulaema_nigrita ~ . - lu_het, data = dat, init.theta = 1.124594029,
##     link = log)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    13.9878476  3.3969372   4.118 3.83e-05 ***
## methodNetTraps -0.2621045  0.2571603  -1.019 0.308096
## methodTraps    -0.5098481  0.2371276  -2.150 0.031547 *
## effort          0.4724685  0.0701487   6.735 1.64e-11 ***
## altitude       -0.0016764  0.0006955  -2.411 0.015928 *
## MAT            -0.0324577  0.0124017  -2.617 0.008865 **
## MAP            -0.0015241  0.0002256  -6.757 1.41e-11 ***
## Tseason        -0.0013552  0.0002488  -5.446 5.14e-08 ***
## Pseason         0.0197479  0.0042447   4.652 3.28e-06 ***
## forest.        -1.1284545  0.2966185  -3.804 0.000142 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.1246) family taken to be 1)
##
##     Null deviance: 393.60  on 177  degrees of freedom
## Residual deviance: 204.18  on 168  degrees of freedom
## AIC: 1774.9
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.125
##           Std. Err.:  0.116
##
##  2 x log-likelihood:  -1752.932
```

Now, i remove the ones that have very small effect size

```
nb_imp2 <- glm.nb(Eulaema_nigrita ~ . - lu_het - altitude - MAP - Tseason,
                  data = dat)
summary(nb_imp2)
```

```
##
## Call:
## glm.nb(formula = Eulaema_nigrita ~ . - lu_het - altitude - MAP -
##     Tseason, data = dat, init.theta = 0.83832615, link = log)
##
## Coefficients:
```

```
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -0.270391   1.015359  -0.266 0.790007
## methodNetTraps -0.069003   0.253088  -0.273 0.785126
## methodTraps    -0.035586   0.248781  -0.143 0.886258
## effort          0.314427   0.074942   4.196 2.72e-05 ***
## MAT             0.011952   0.003950   3.026 0.002481 **
## Pseason         0.017720   0.004312   4.110 3.96e-05 ***
## forest.        -1.248200   0.327025  -3.817 0.000135 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.8383) family taken to be 1)
##
##     Null deviance: 299.24  on 177  degrees of freedom
## Residual deviance: 209.50  on 171  degrees of freedom
## AIC: 1829.2
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.8383
##           Std. Err.:  0.0832
##
##  2 x log-likelihood:  -1813.1810
```

The `method` is not significant anymore, lets remove it also

```
nb_imp3 <- glm.nb(Eulaema_nigrita ~ effort+MAT+Pseason+forest., data = dat)
summary(nb_imp3)
```

```
##
## Call:
## glm.nb(formula = Eulaema_nigrita ~ effort + MAT + Pseason + forest.,
##     data = dat, init.theta = 0.8381012847, link = log)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.272920   0.981854  -0.278  0.78104
## effort       0.306060   0.060476   5.061 4.17e-07 ***
## MAT          0.012081   0.003796   3.183  0.00146 **
## Pseason      0.017534   0.004280   4.097 4.18e-05 ***
## forest.     -1.255141   0.312063  -4.022 5.77e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.8381) family taken to be 1)
##
##     Null deviance: 299.16  on 177  degrees of freedom
## Residual deviance: 209.52  on 173  degrees of freedom
## AIC: 1825.3
##
## Number of Fisher Scoring iterations: 1
##
```

```
##
##               Theta:  0.8381
##           Std. Err.:  0.0832
##
##  2 x log-likelihood:  -1813.2540
```

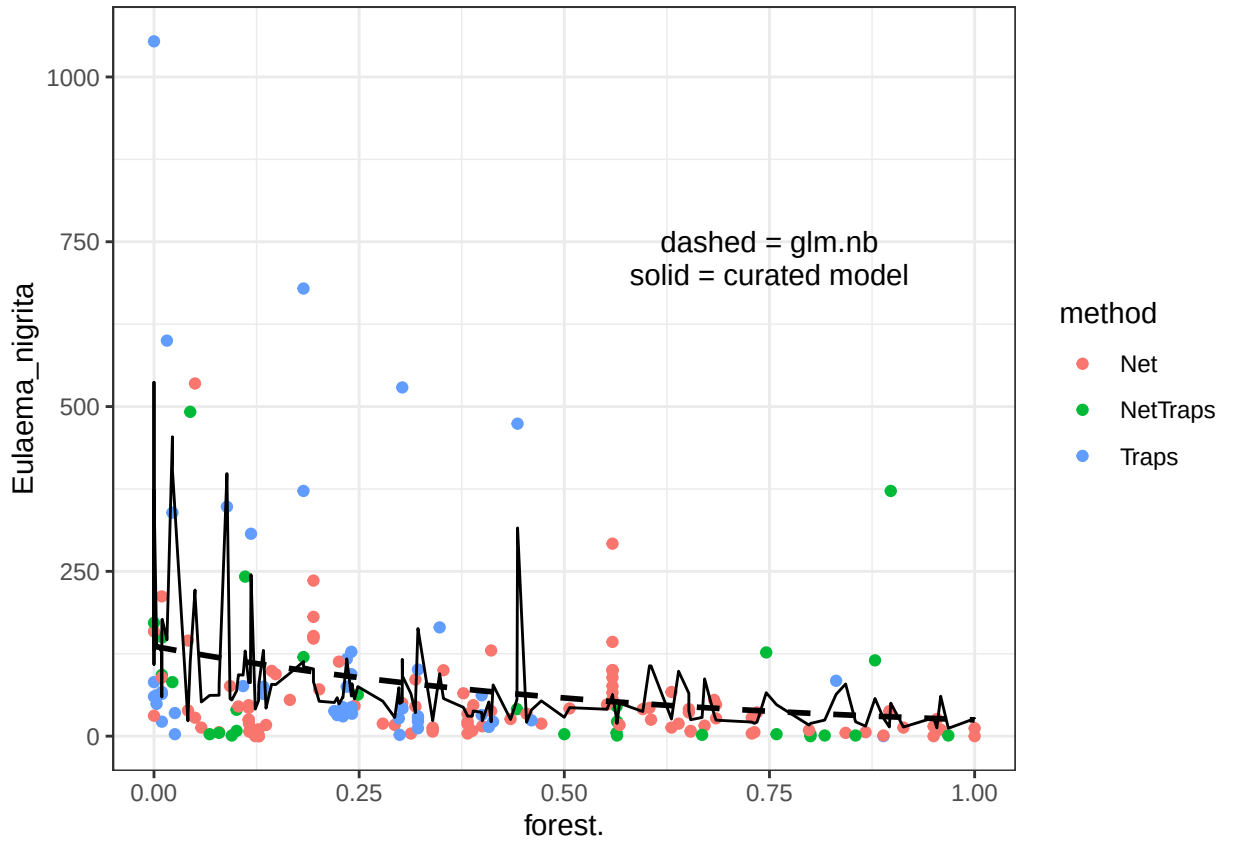
```
df_models <- data.frame(
  model = c("m_all", "nb_all", "nb_imp1", "nb_imp2", "nb_imp3"),
  AIC = c(NA, 1776.8, 1774.9, 1829.2, 1825.3),
  Theta = c(NA, 1.125, 1.125, 0.8383, 0.8381),
  Res.dev = c(11295, 204, 204, 210, 210),
  deg.free = c(167, 167, 168, 171, 173)
)

df_models
```

```
##     model    AIC  Theta Res.dev deg.free
## 1   m_all     NA     NA   11295      167
## 2  nb_all 1776.8 1.1250     204      167
## 3 nb_imp1 1774.9 1.1250     204      168
## 4 nb_imp2 1829.2 0.8383     210      171
## 5 nb_imp3 1825.3 0.8381     210      173
```

Lets now plot the model against the data. As `forest.` has the biggest effect size, lets plots the data against it.

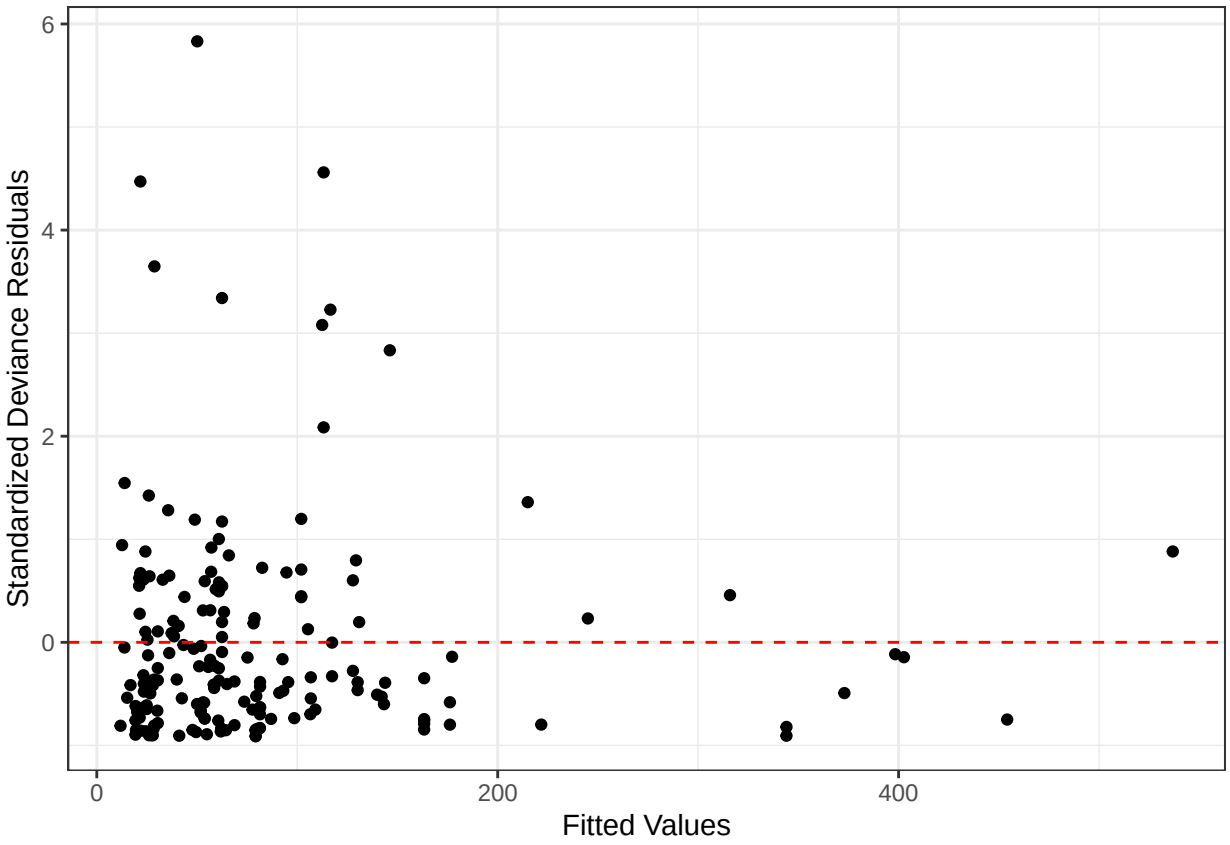
```
# Plotting the model against the data using ggplot and fitting the nb_imp3 model as a regression line
fit_m <- fitted(nb_imp3)

dat %>%
  mutate(fitted = fitted(nb_imp3)) %>%
  ggplot(aes(forest., Eulaema_nigrita)) +
  geom_point(aes(color = method)) +
  geom_line(aes(forest., fitted), color = "black") +
  geom_smooth(method = "glm.nb", se = FALSE, linetype = "dashed",
              color = "black") +
  theme_bw() +
  annotate("text", x = 0.75, y = 750, label = "dashed = glm.nb") +
  annotate("text", x = 0.75, y = 700, label = "solid = curated model")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



The model fits well overall, however there is something going on in the beginning as the residuals are greater. The model doesnt handle the first values well as the residuals are greater there.

```
# Plotting the residuals using ggplot
dat %>%
  mutate(residuals = resid(nb_imp3, type = "pearson")) %>%
  ggplot(aes(fitted(nb_imp3), residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(x = "Fitted Values", y = "Standardized Deviance Residuals") +
  theme_bw()
```


Standardized Deviance Residuals
6
4
2
0
0
200
400
Fitted Values

Result