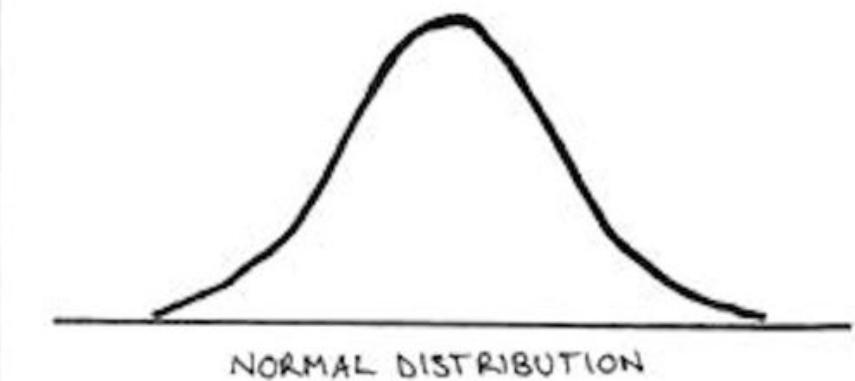


ZEROs & Model Diagnostics

BIO 599: Ecological Data Analysis



NORMAL DISTRIBUTION



PARANORMAL DISTRIBUTION

Frazee.

Reminders/Updates

1. Tuesday, Nov 11 no class
 2. Thursday, Nov 13: GLMM
 3. Tuesday, Nov 18: GLMM workshop
(week 10) due: Nov 21
 4. Final Project: Dec 5th w/ workshop
 5. Anonymous Feedback – please ☺



Outline for Today



01

Zero-truncated data

02

Zero-inflated data

03

Model diagnostics
with DHARMA

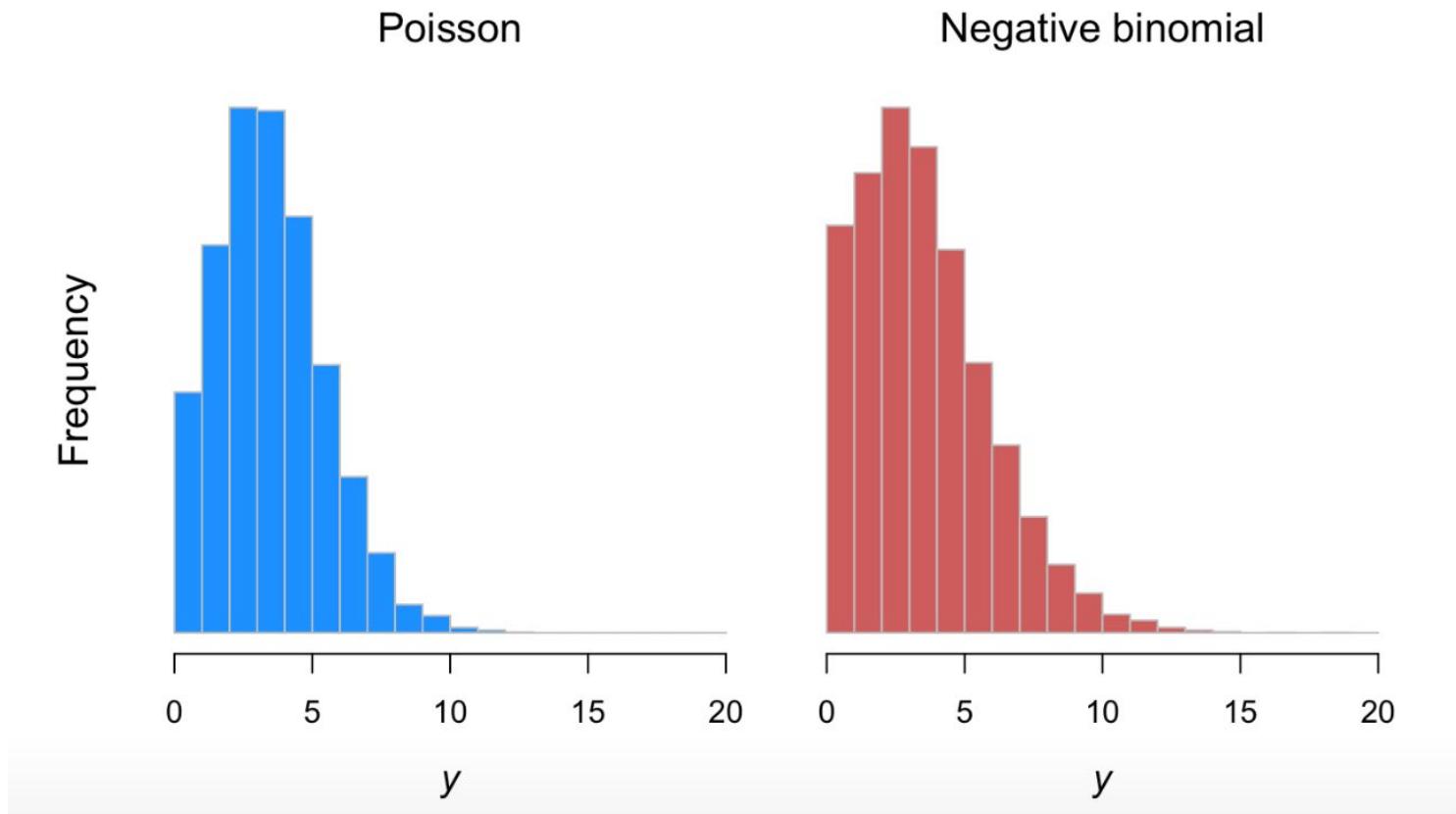
Zero-truncated data

Response variable with out zero.

- Zero truncated means the response variable cannot have a value of 0
- Zero-truncated data are not necessarily a problem
- Rather, an underlying assumption of Poisson or neg binomial may be the problem

Zero-truncated data

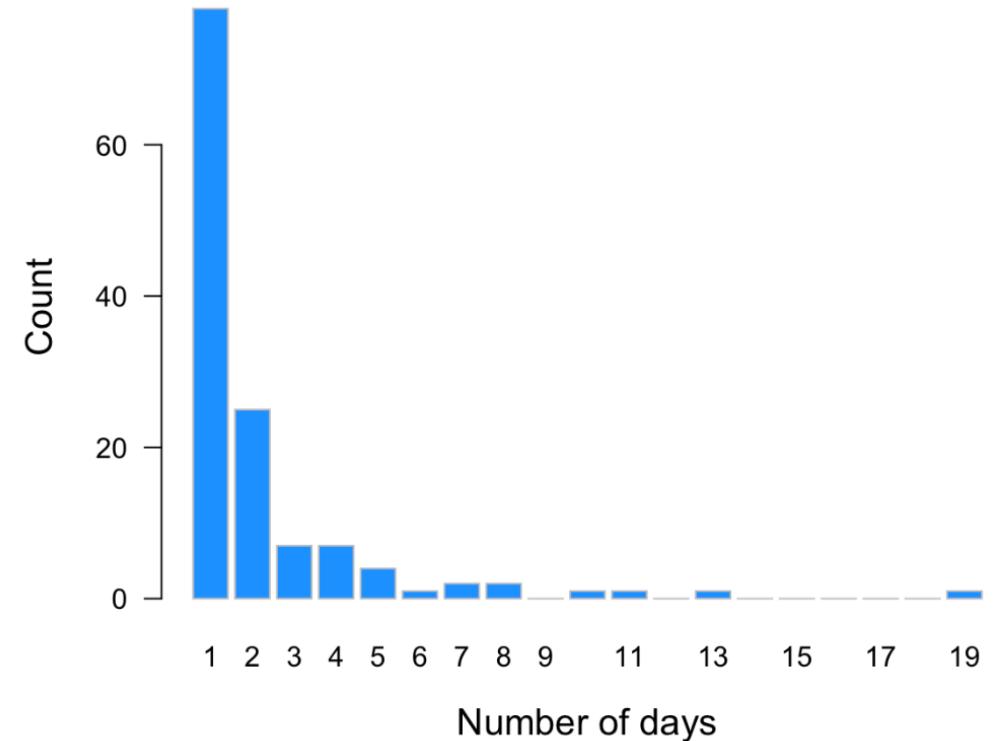
Need to adjust the probability functions for the Poisson and NB distributions to exclude the probability of a zero observation



Zero-truncated: example

- Response: number of days that carcasses from road-killed snakes stay on roads (mean=2.2, variance=6.5)
- Predictors: total rainfall (mm) and an indicator of where on the pavement the snake was killed (lane or shoulder)

When lane is ~~fixed~~
then mean = ~~the~~ dispersion.



Zero-truncated: example

Let's first consider a regular negative-binomial regression model with glmmTMB

```
smod_nb <- glmmTMB(N_days ~ Road_Loc + Tot_Rain, data = Snakes, family="nbinom2")
summary(smod_nb)
```

```
Family: nbinom2  ( log )
Formula:      N_days ~ Road_Loc + Tot_Rain
Data: Snakes

  AIC    BIC   logLik deviance df.resid
469.3  480.7 -230.6    461.3     126

Dispersion parameter for nbinom2 family (): 4.15

Conditional model:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.418101  0.107339  3.895 9.81e-05 ***
Road_LocV   0.453524  0.151408  2.995  0.00274 **
Tot_Rain    0.025127  0.005075  4.951 7.39e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

est. met ~
log. & exp to
get RR.

greater than 1
so overdispersion
prev.

nbinom1 is appropriate when the variance increases linearly with the mean.

nbinom2 is more flexible and is typically used when the variance increases more sharply with the mean***

Zero-truncated: example

Now let's fit a zero-truncated NB regression model with glmmTMB

```
smod_ztnb <- glmmTMB(N_days ~ Road_Loc + Tot_Rain, data = Snakes,  
                      family = "truncated_nbinom2")
```

```
Family: truncated_nbinom2 ( log )  
Formula: N_days ~ Road_Loc + Tot_Rain  
Data: Snakes  
  
      AIC      BIC      logLik deviance df.resid  
 351.9    363.4    -172.0     343.9      126  
  
Dispersion parameter for truncated_nbinom2 family (): 1.27e-08  
  
Conditional model:  
              Estimate Std. Error z value Pr(>|z|)  
(Intercept) -18.14719  9372.02750 -0.002 0.998455  
Road_LocV     1.14474    0.38568   2.968 0.002996 **  
Tot_Rain      0.07241    0.02150   3.368 0.000758 ***  
---  
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Zero-truncated: example

Differences between NB GLM and truncated NB GLM will become smaller if the observed values are further away from zero.

| | NB | NB SE | T-NB | T-NB SE |
|-------------|-------|-------|---------|---------|
| (Intercept) | 0.418 | 0.107 | -18.147 | 9372.0 |
| Road | 0.454 | 0.152 | 1.144 | 0.386 |
| Rain | 0.025 | 0.005 | 0.072 | 0.021 |

| | df <dbl> | AIC <dbl> |
|-----------|-------------|--------------|
| smod_nb | 4 | 469.2788 |
| smod_ztnb | 4 | 351.9488 |

Zero-truncated: summary



Zero-truncated models are designed for count data where zeros are not observed.



Zero-Truncated Poisson is used for equidispersed data, while Zero-Truncated Negative Binomial is used for overdispersed data.



Maximum Likelihood Estimation (MLE) is used to estimate these models.



Check model assumptions! (more on that later)

When Zeros are in
the response &
not predictors.



1. What are too many zeros?
2. Sources of Zeros
3. Examples: ZIP, ZINB, ZAP, ZANB

What are *too* Many Zeros?

An overabundance of 0's in the response data

More zeros than expected given the assumed statistical distribution = zero-inflation

A simple histogram or frequency plot with a large spike at zero gives an early warning of possible zero inflation.

BUT a lot of zeros does NOT mean that data are zero-inflated



Why do we need to account for “too many zeros”?

- Ignoring zero inflation can have two consequences:
 - The estimated parameters and standard errors may be biased
 - The excessive number of zeros can cause overdispersion.
- Before discussing techniques that can cope with all these zeros, we need to ask the question: Why do we have all these zeros?

Sources of zeros

1. structural errors: a species is not present because the habitat is not suitable.



2. design error: where poor experimental design or sampling practices result in zero observations



3. observer error: species look similar or are difficult to detect.



4. ‘species’ error: the habitat is suitable, but the site is not used.



Modeling zero-inflated data

Hurdle models (ZAP and ZANB)

1. Model for whether an observation is 0
2. A truncated count model for the non-zero observations

Zero-inflated mixture models (ZIP and ZINB)

1. A model for whether or not an observation is an inflated 0
2. A count model that also allows for the possibility of a 0

ZI = Zero-inflated

ZA = Zero-adjusted or altered

P = Poisson

NB = Negative Binomial

ZAP and ZANB

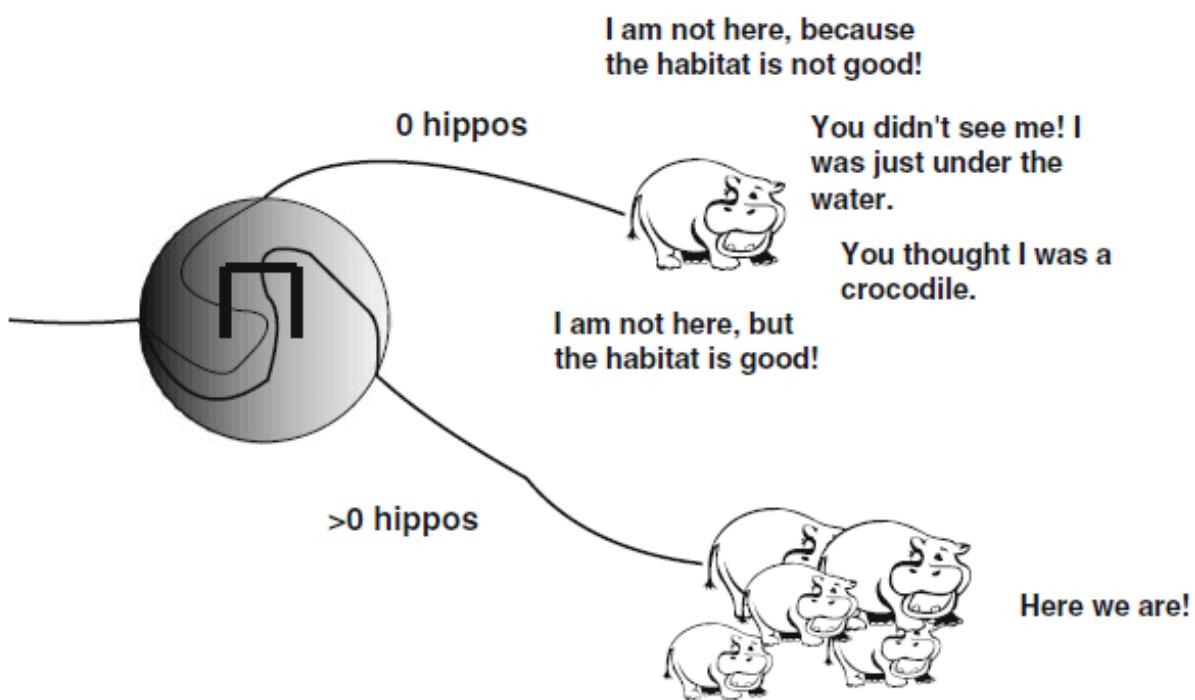


Fig. 11.4 Sketch of a two-part, or hurdle model. There are two processes; one is causing zeros versus non-zeros, the other process is explaining the non-zero counts. This is expressed with the hurdle in the circle; you have to cross it to get non-zero counts. The model does not make a distinction between the different types of zeros

Treat all the four different types of zeros the same

ZIP and ZINB

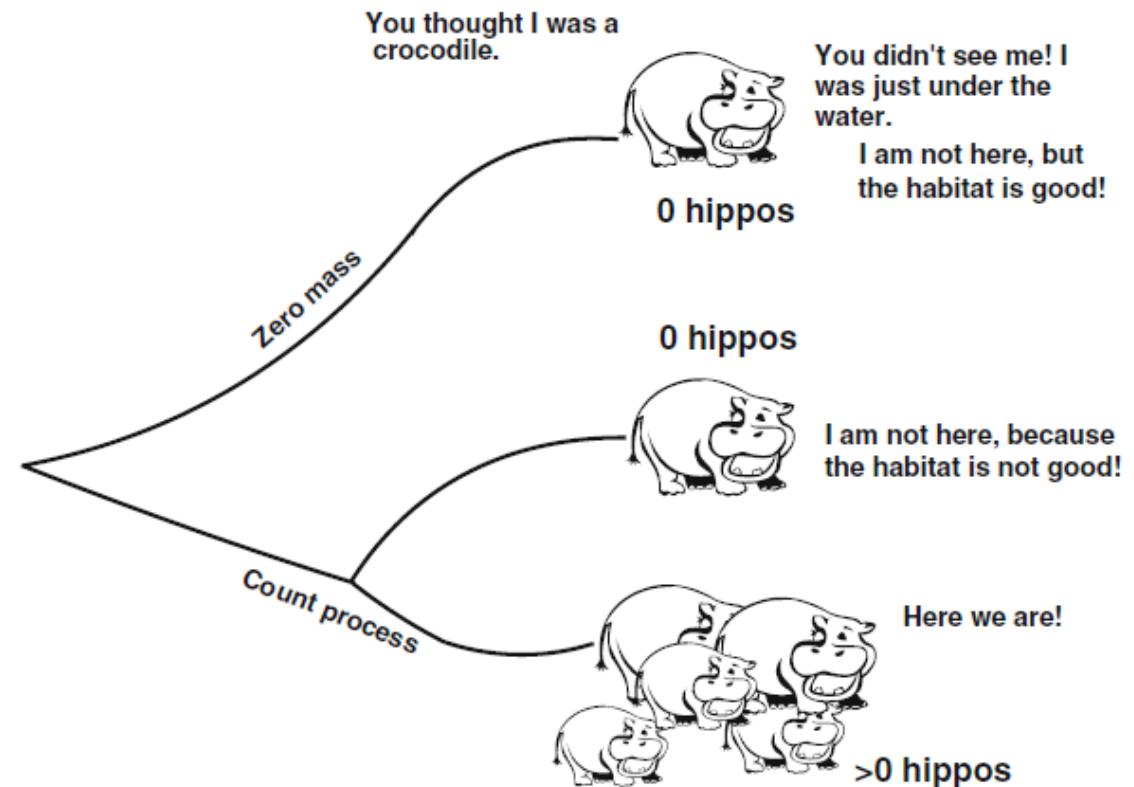


Fig. 11.5 Sketch of the underlying principle of mixture models (ZIP and ZINB). In counting hippos at sites, one can measure a zero because the habitat is not good (the hippos don't like the covariates), or due to poor experimental design and inexperienced observers (or experienced observers but difficult to observe species)

Separate the zeros into “true” and “extra” categories

What model should I use?



Zero-inflation models may be more elegant and informative if the same predictors are thought to contribute to the extra and real zeros or if you have additional predictors for the excess zeros.



Hurdle models can be useful in that they allow you to model the zeros and non-zeros with different predictors or different roles of the same predictors. Maybe one process leads to the zero/non-zero data and another leads to the non-zero magnitude.



Choose based on biological knowledge OR fit all models and choose based on AIC best-model.

Examples: Cod parasite

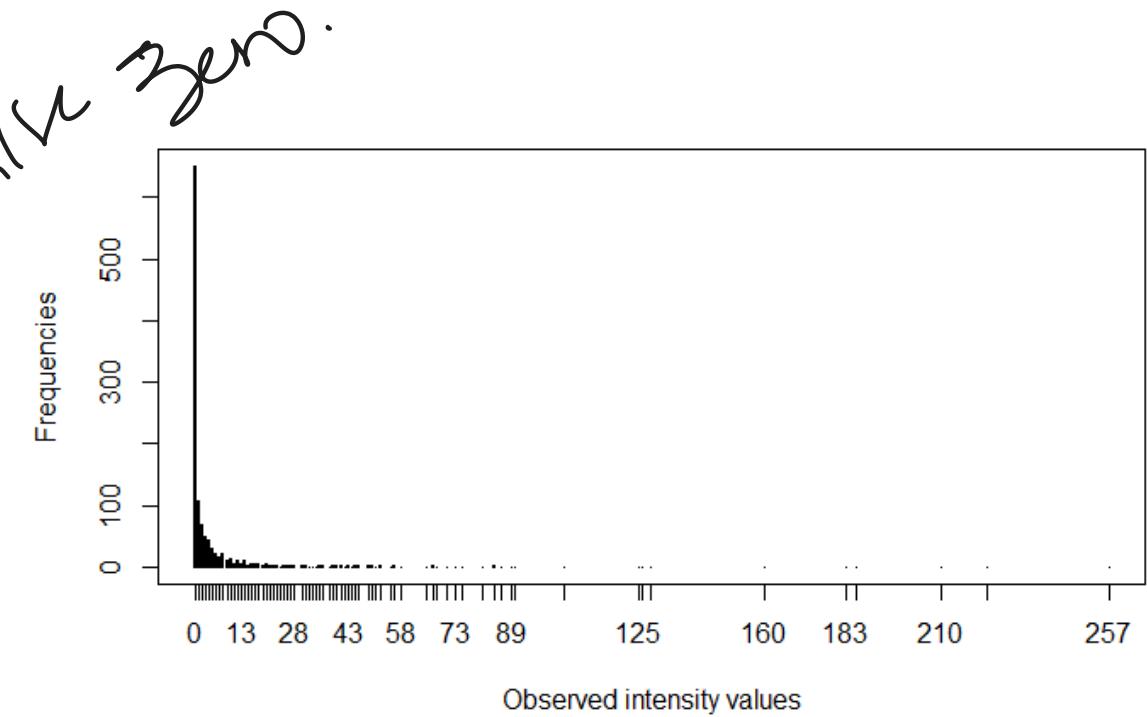
Zero in fish noted
because + the zero
is five
zero

Response: Intensity (count of parasites on fish)

Explanatory: 3 years, 4 area, and the length of the fish

Source of zeros:

1. Experimental error: Fish not exposed to the parasite
2. Observer error: Difficult to detect
3. Fish that were in contact with the parasite but not infected (TRUE zeros)



ZIP: example

```
library(glmmTMB)
zip1 <- glmmTMB(Intensity ~ fArea*fYear + Length, ziformula = ~ fArea*fYear + Length,
                 data = ParasiteCod2, family = "poisson")
```

- **ziformula** uses a **logistic regression model** (binomial dist) to estimate the probability that an observation is from the zero-inflation process
- **ziformula** allows the following formulae specifications
 - ~ 1 (the zero-inflation probability is constant for all observations)
 - ~ any combination of predictors or other variables you collects

ZIP: example

| Predictors | Intensity | | | Zero-Inflated Model |
|--------------------------|-----------------------|------------------------------------------|---------------|--------------------------------------------------|
| | Incidence Rate Ratios | CI | p | |
| Count Model | | | | |
| (Intercept) | 52.21 | 46.34 – 58.84 | <0.001 | (Intercept) 1.01 0.56 – 1.82 0.977 |
| fArea [2] | 1.37 | 1.19 – 1.57 | <0.001 | fArea [2] 3.75 2.14 – 6.58 <0.001 |
| fArea [3] | 0.75 | 0.65 – 0.87 | <0.001 | fArea [3] 4.22 2.61 – 6.83 <0.001 |
| fArea [4] | 2.28 | 2.08 – 2.49 | <0.001 | fArea [4] 0.74 0.43 – 1.25 0.260 |
| fYear [2000] | 1.15 | 0.99 – 1.32 | 0.065 | fYear [2000] 0.68 0.34 – 1.33 0.259 |
| fYear [2001] | 0.40 | 0.27 – 0.59 | <0.001 | fYear [2001] 12.79 5.40 – 30.26 <0.001 |
| Length | 0.96 | 0.96 – 0.97 | <0.001 | Length 0.99 0.98 – 1.00 0.064 |
| fArea [2] × fYear [2000] | 0.55 | 0.42 – 0.71 | <0.001 | fArea [2] × fYear [2000] 1.07 0.39 – 2.91 0.902 |
| fArea [3] × fYear [2000] | 2.30 | 1.88 – 2.83 | <0.001 | fArea [3] × fYear [2000] 0.43 0.18 – 1.04 0.060 |
| fArea [4] × fYear [2000] | 1.31 | 1.11 – 1.53 | 0.001 | fArea [4] × fYear [2000] 0.42 0.13 – 1.36 0.148 |
| fArea [2] × fYear [2001] | 2.64 | 1.75 – 3.99 | <0.001 | fArea [2] × fYear [2001] 0.07 0.03 – 0.21 <0.001 |
| fArea [3] × fYear [2001] | 2.75 | 1.81 – 4.16 | <0.001 | fArea [3] × fYear [2001] 0.07 0.03 – 0.19 <0.001 |
| fArea [4] × fYear [2001] | 2.44 | 1.66 – 3.59 | <0.001 | fArea [4] × fYear [2001] 0.09 0.03 – 0.25 <0.001 |
| | | Observations | 1191 | |
| | | R ² / R ² adjusted | 0.686 / 0.682 | |

ZIP: example

- the ZIP model only deals with zero inflation, not directly with overdispersion in the non-zero count data.
- If the overdispersion in a Poisson GLM is caused by the excessive number of zeros, then the ZIP will take care of the overdispersion, and we are finished.
- But if the overdispersion is not caused by the zeros, then the ZIP is not the appropriate model either!

```
> check_overdispersion(zip1)
# Overdispersion test

  dispersion ratio =   6.170
                    p-value = < 0.001

overdispersion detected.
```

ZINB

```
library(glmmTMB)
zinbl <- glmmTMB(Intensity ~ fArea*fYear + Length, ziformula = ~ fArea*fYear + Length,
                   data = ParasiteCod2, family = "nbinom2")

library(lmtest)
lrtest(zip1,zinbl)
AIC(zip1,zinbl)
```

| | df <dbl> | AIC <dbl> |
|-------|-------------|--------------|
| Zip1 | 26 | 13687.097 |
| Zinbl | 27 | 4954.896 |

Zero-inflated negative binomial is a much better model!

ZIP and ZINB

Model selection of the explanatory variables:

1. Drop length from count model
2. Drop Area * Year from the count model
3. Drop length from the logistic model
4. Drop Area * Year from the logistic model

No further terms could be dropped

| Dropped term | df | AIC | Likelihood ratio test | |
|---------------------------------|----|----------|-----------------------|------------------------|
| None | 27 | 4954.897 | | |
| Length from μ_i | 26 | 4994.993 | $X^2 = 42.096$ | (df = 1, $p < 0.001$) |
| Area \times Year from μ_i | 21 | 4957.146 | $X^2 = 14.249$ | (df = 6, $p = 0.026$) |
| Length from π_i | 26 | 4965.019 | $X^2 = 12.122$ | (df = 1, $p < 0.001$) |
| Area \times Year from π_i | 21 | 4961.751 | $X^2 = 18.853$ | (df = 6, $p = 0.004$) |

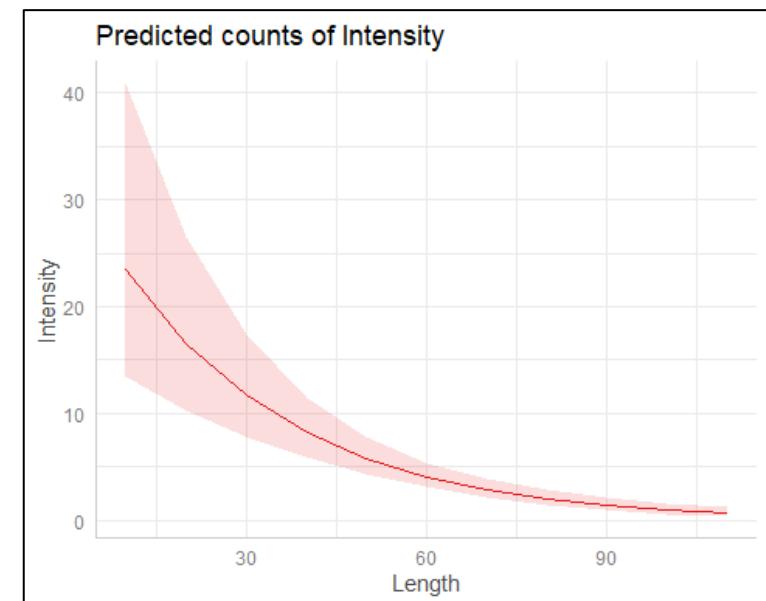
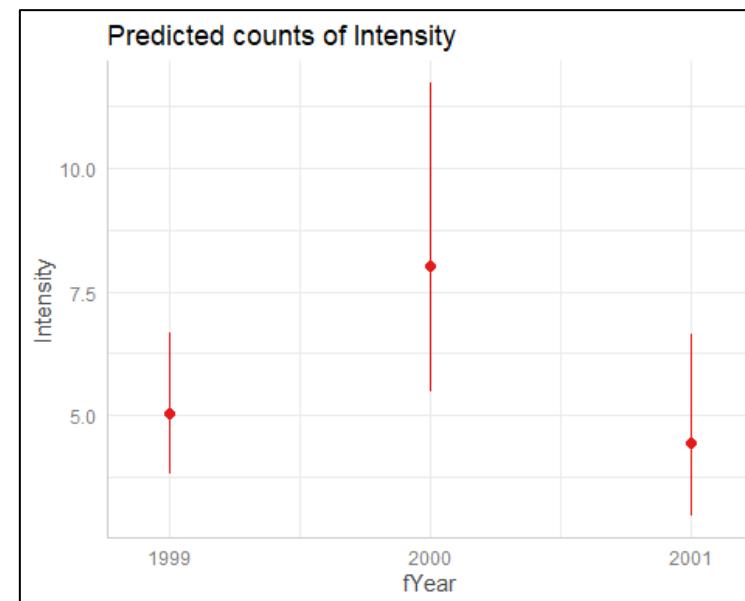
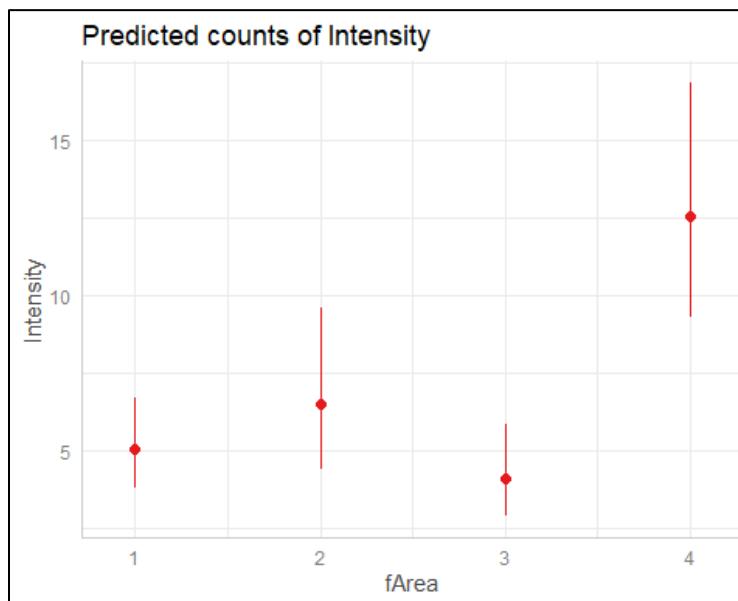
5. Test whether Length, Area, or Year can be dropped from the count model and length and the Area \times Year interaction from the logistic model.

ZINB

```
zinb2 <- glmmTMB(Intensity ~ fArea+fYear + Length, ziformula = ~ fArea*fYear + Length,
                    data = ParasiteCod2, family = "nbinom2")
```

```
plot(qapredict(zinb2))
```

| Predictors | Intensity | | |
|--------------------|-----------------------|---------------|--------|
| | Incidence Rate Ratios | CI | p |
| Count Model | | | |
| (Intercept) | 33.34 | 17.57 – 63.26 | <0.001 |
| fArea [2] | 1.29 | 0.82 – 2.02 | 0.270 |
| fArea [3] | 0.82 | 0.55 – 1.22 | 0.326 |
| fArea [4] | 2.49 | 1.70 – 3.64 | <0.001 |
| fYear [2000] | 1.59 | 1.13 – 2.23 | 0.007 |
| fYear [2001] | 0.88 | 0.63 – 1.22 | 0.443 |
| Length | 0.97 | 0.96 – 0.98 | <0.001 |
| (Intercept) | 1.45 | 1.36 – 1.57 | |

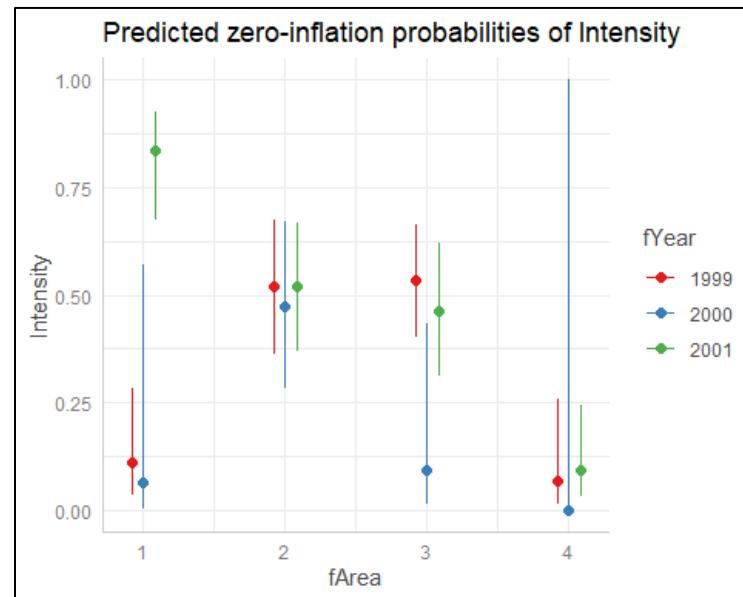
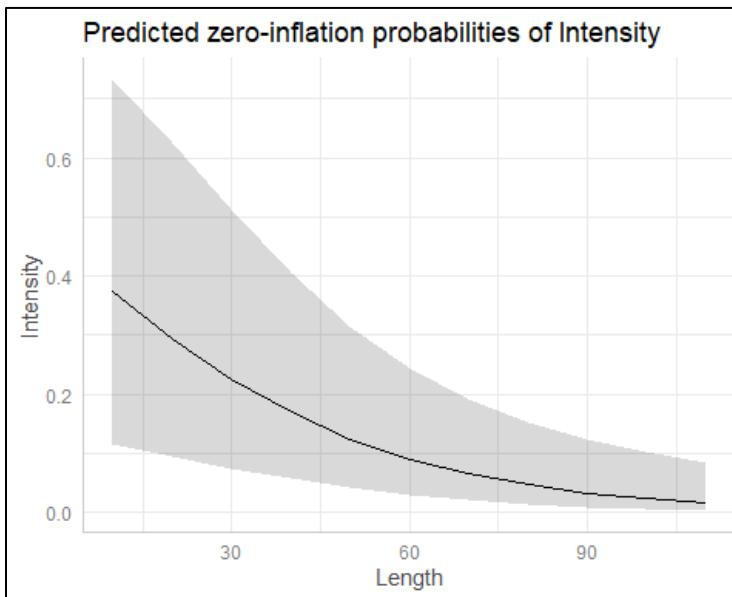


$$\text{Percent change} = (\text{IRR}-1) * 100$$

ZINB

```
zinb2 <- glmmTMB(Intensity ~ fArea+fYear + Length, ziformula = ~ fArea*fYear + Length,
                    data = ParasiteCod2, family = "nbinom2")
```

```
plot(ggpredict(zinb2, terms=c("fArea", "fYear"), type="zi_prob"))
plot(ggpredict(zinb2, terms=c("Length"), type="zi_prob"))
```



| Predictors | Intensity | | |
|------------------------------------------|-----------------------|---------------|--------|
| | Incidence Rate Ratios | CI | p |
| Count Model | | | |
| (Intercept) | 33.34 | 17.57 – 63.26 | <0.001 |
| fArea [2] | 1.29 | 0.82 – 2.02 | 0.270 |
| fArea [3] | 0.82 | 0.55 – 1.22 | 0.326 |
| fArea [4] | 2.49 | 1.70 – 3.64 | <0.001 |
| fYear [2000] | 1.59 | 1.13 – 2.23 | 0.007 |
| fYear [2001] | 0.88 | 0.63 – 1.22 | 0.443 |
| Length | 0.97 | 0.96 – 0.98 | <0.001 |
| (Intercept) | 1.45 | 1.36 – 1.57 | |
| Zero-Inflated Model | | | |
| (Intercept) | 0.85 | 0.16 – 4.56 | 0.853 |
| fArea [2] | 8.83 | 2.47 – 31.53 | 0.001 |
| fArea [3] | 9.34 | 2.79 – 31.24 | <0.001 |
| fArea [4] | 0.60 | 0.10 – 3.49 | 0.570 |
| fYear [2000] | 0.55 | 0.03 – 11.33 | 0.698 |
| fYear [2001] | 40.73 | 9.91 – 167.39 | <0.001 |
| Length | 0.96 | 0.94 – 0.99 | 0.002 |
| fArea [2] × fYear [2000] | 1.50 | 0.06 – 35.07 | 0.800 |
| fArea [3] × fYear [2000] | 0.16 | 0.00 – 5.89 | 0.322 |
| fArea [4] × fYear [2000] | 0.00 | 0.00 – Inf | 0.993 |
| fArea [2] × fYear [2001] | 0.02 | 0.00 – 0.13 | <0.001 |
| fArea [3] × fYear [2001] | 0.02 | 0.00 – 0.09 | <0.001 |
| fArea [4] × fYear [2001] | 0.03 | 0.00 – 0.30 | 0.002 |
| Observations | 1191 | | |
| R ² / R ² adjusted | 0.917 / 0.916 | | |

ZAP and ZANB

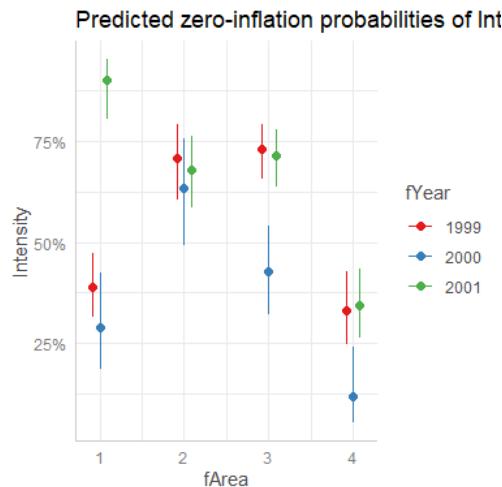
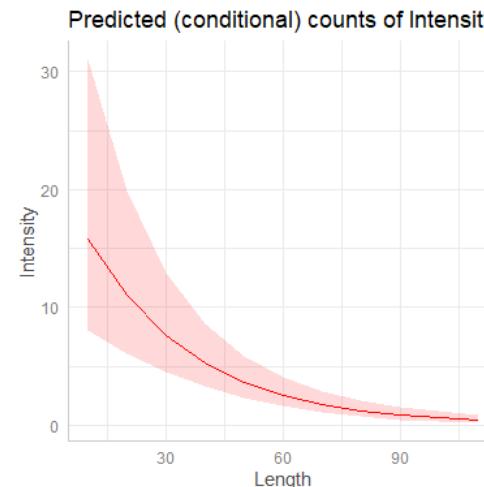
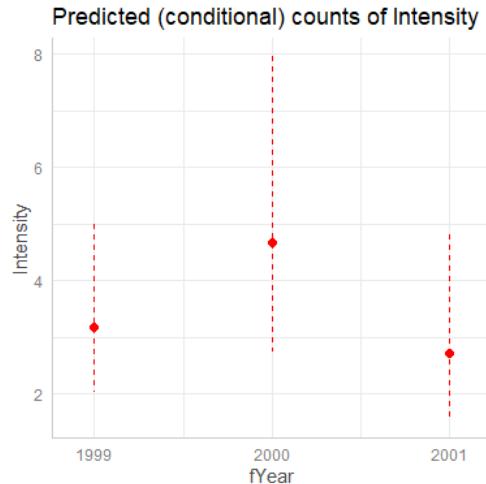
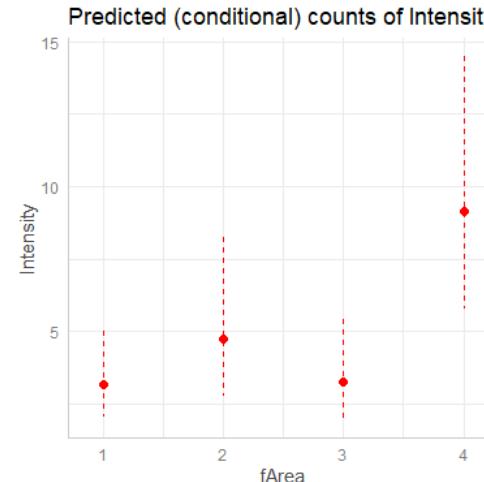
```
Hurd1 <- glmmTMB(Intensity ~ fArea*fYear + Length, ziformula = ~ fArea*fYear + Length,  
                    data = ParasiteCod2, family = "truncated_poisson")  
Hurd2 <- glmmTMB(Intensity ~ fArea*fYear + Length, ziformula = ~ fArea*fYear + Length,  
                    data = ParasiteCod2, family = "truncated_nbom2")  
AIC(Hurd1, Hurd2)
```

- Ziformula = the zero vs non-zero values
- family “truncated_x” because the count data cannot include zeros

| | df <dbl> | AIC <dbl> |
|-------|-------------|--------------|
| Hurd1 | 26 | 13687.576 |
| Hurd2 | 27 | 4937.085 |

ZANB: Final model

```
Hurd3 <- glmmTMB(Intensity ~ fArea+fYear + Length, ziformula = ~ fArea*fYear,
                   data = ParasiteCod2, family = "truncated_nbino2")
```



| Predictors | Intensity | | |
|------------------------------------------|-----------------------|---------------|--------|
| | Incidence Rate Ratios | CI | p |
| Count Model | | | |
| (Intercept) | 22.81 | 10.68 – 48.73 | <0.001 |
| fArea [2] | 1.50 | 0.89 – 2.52 | 0.127 |
| fArea [3] | 1.02 | 0.65 – 1.61 | 0.926 |
| fArea [4] | 2.89 | 1.86 – 4.48 | <0.001 |
| fYear [2000] | 1.47 | 0.99 – 2.18 | 0.057 |
| fYear [2001] | 0.86 | 0.58 – 1.27 | 0.439 |
| Length | 0.96 | 0.95 – 0.97 | <0.001 |
| (Intercept) | 1.23 | 1.13 – 1.40 | |
| Zero-Inflated Model | | | |
| (Intercept) | 0.64 | 0.46 – 0.89 | 0.009 |
| fArea [2] | 3.76 | 2.15 – 6.57 | <0.001 |
| fArea [3] | 4.20 | 2.61 – 6.77 | <0.001 |
| fArea [4] | 0.77 | 0.45 – 1.31 | 0.331 |
| fYear [2000] | 0.64 | 0.33 – 1.25 | 0.193 |
| fYear [2001] | 14.05 | 6.01 – 32.84 | <0.001 |
| fArea [2] × fYear [2000] | 1.12 | 0.41 – 3.02 | 0.828 |
| fArea [3] × fYear [2000] | 0.43 | 0.18 – 1.04 | 0.062 |
| fArea [4] × fYear [2000] | 0.43 | 0.14 – 1.38 | 0.156 |
| fArea [2] × fYear [2001] | 0.06 | 0.02 – 0.18 | <0.001 |
| fArea [3] × fYear [2001] | 0.07 | 0.02 – 0.18 | <0.001 |
| fArea [4] × fYear [2001] | 0.08 | 0.03 – 0.21 | <0.001 |
| Observations | 1191 | | |
| R ² / R ² adjusted | 0.952 / 0.952 | | |

ZINB vs. ZANB results

Count part:

p-values and corresponding terms give similar conclusions but different magnitudes

Binomial part:

length is not significant in the ZANB (it is removed)
p-values and corresponding terms give similar conclusions but different magnitudes

| | df <dbl> | AIC <dbl> |
|-------|-------------|--------------|
| Zinb2 | 21 | 4957.145 |
| Hurd3 | 20 | 4936.624 |

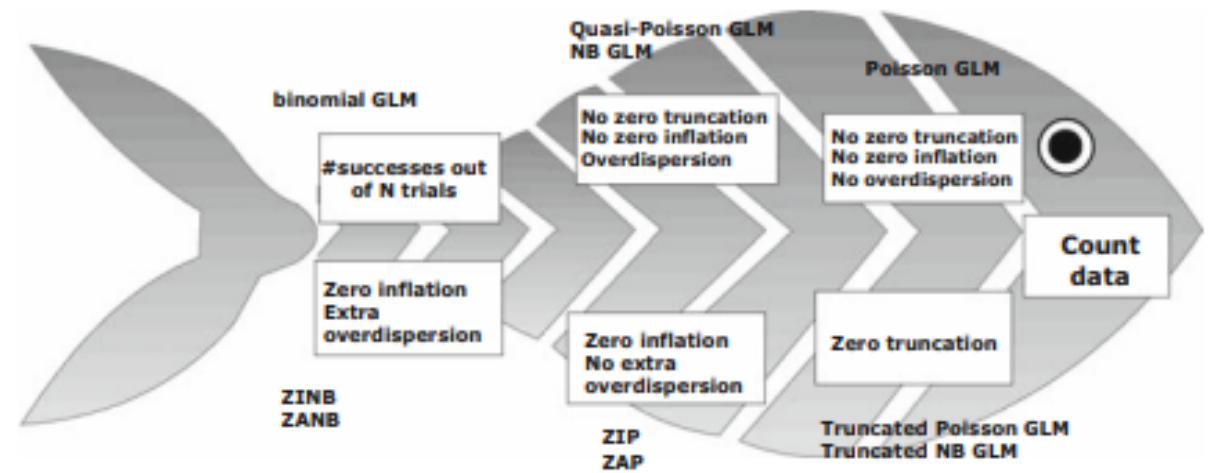
Comparing Poisson, Quasi-Poisson, NB, ZIP, ZINB, ZAP and ZANB GLMs

1. Common Sense

- Small overdispersion = quasi-poisson
- Large overdispersion = NB
- Zero-inflation = ZIP, ZAP
- Zero-inflation and overdispersion = ZINB, ZANB

2. Model Validation

3. Information Criteria



What about continuous data with many Zeros?

Tweedie
distribution
(first choice!)

ZAG: zero-
adjusted
gamma

Summary

Zero-inflated and Zero-adjusted (hurdle) models:

- ZI models address excessive zero counts by assuming a mixture of two processes: one generating excess zeros and another for the real zeros and counts.
- ZA models tackle excess zeros by modeling two separate components: one for the probability of observing zero counts (hurdle component) and another for the distribution of non-zero counts (count component).
- Choosing depends on the nature of excess zeros, with model selection guided by information criteria

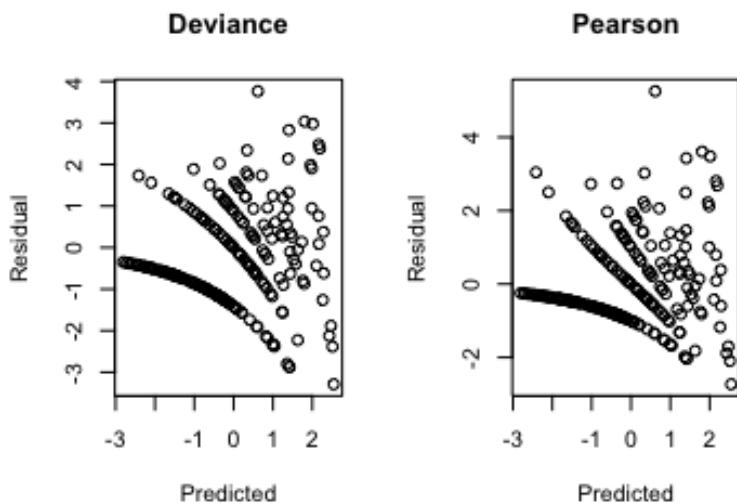
Model diagnostics with DHARMa

GLM residual plots

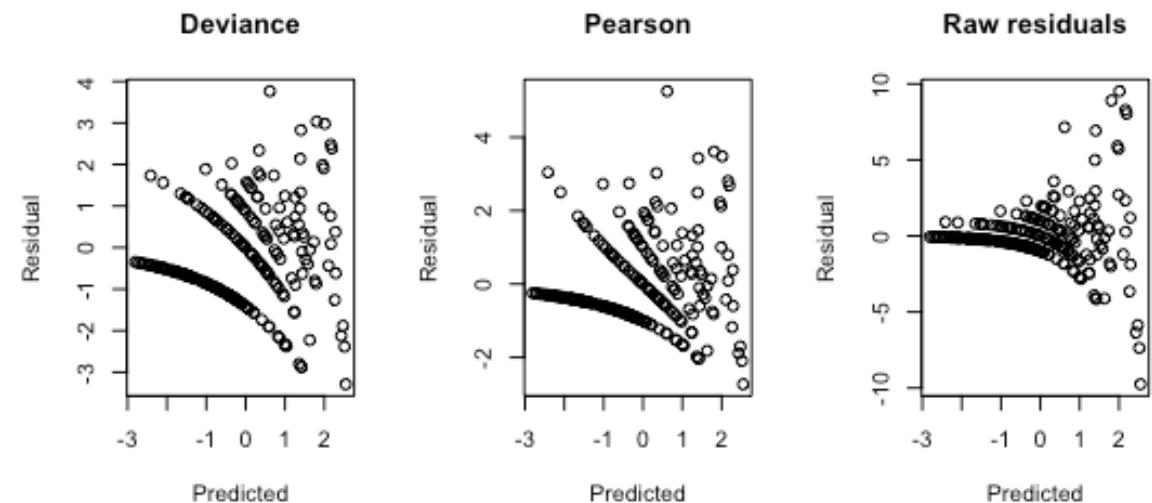
- Last week, we plotted the deviance residuals
- Deviance is a measure of how well a model fits the data compared to a saturated model (a model that perfectly fits the data).
- Deviance residuals are used to assess the contribution of each observation to the total deviance of the model
- Somewhat useful for detecting outliers/influence observations and overall model fit....BUT

The interpretation of residuals for GLMs is often problematic.

Conventional residual plots



Conventional residual plots



Why??

The expected distribution of the data (aka predictive distribution) changes with the fitted values.

Solution: simulated residuals (DHARMa)

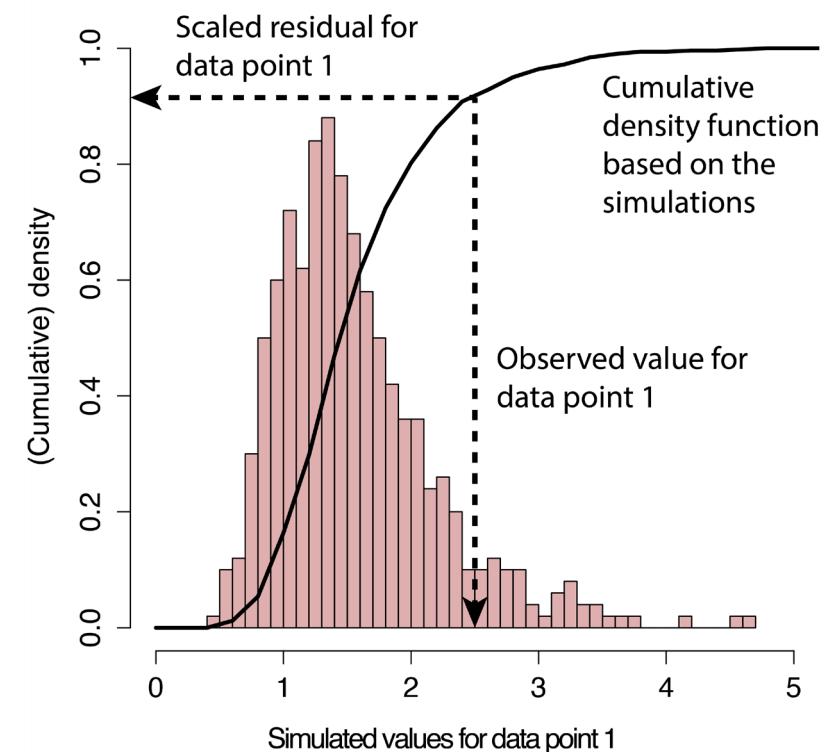
- allow for model diagnostics that doesn't depend on assumptions about the distribution of the residuals
- Uses a simulation-based approach to create readily interpretable scaled (quantile) residuals for fitted generalized linear (mixed) models
- Residuals are standardized to values between 0 and 1 and can be interpreted as intuitively as residuals from a linear regression.

DHARMa stands for "Diagnostics for HierArchical Regression Models" - which, strictly speaking, would make DHARM. But in German, Darm means intestines; plus, the meaning of DHARMa in Hinduism makes the current abbreviation so much more suitable for a package that tests whether your model is in harmony with your data:

DHARMA basic steps

1. **Simulate the response** from the model's estimated parameters and distribution
2. **Calculate the empirical cumulative distribution function** for each simulated observation
3. **Residual is defined** as the value of the empirical density function at the value of the observed data,

A residual of 0 means that all simulated values are larger than the observed value, and a residual of 0.5 means half of the simulated values are larger than the observed value



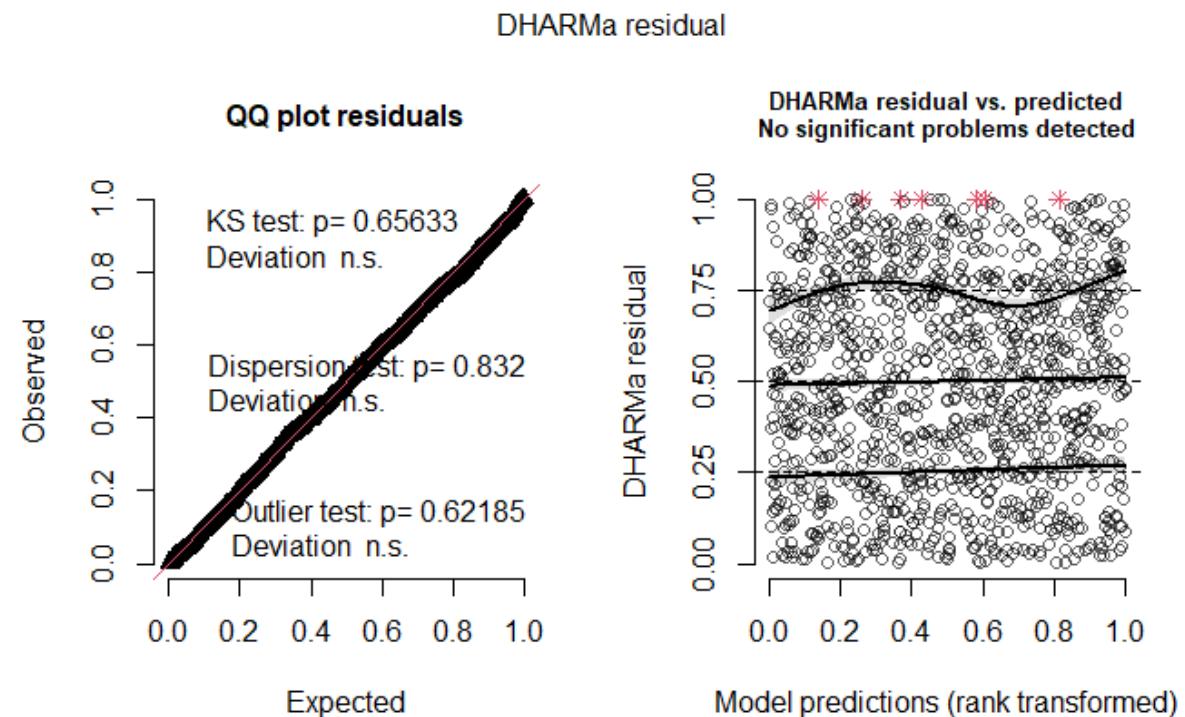
Workflow in DHARMA

1. Simulate Residuals

```
simulationOutput <- simulateResiduals(fittedModel = Hurd3)
```

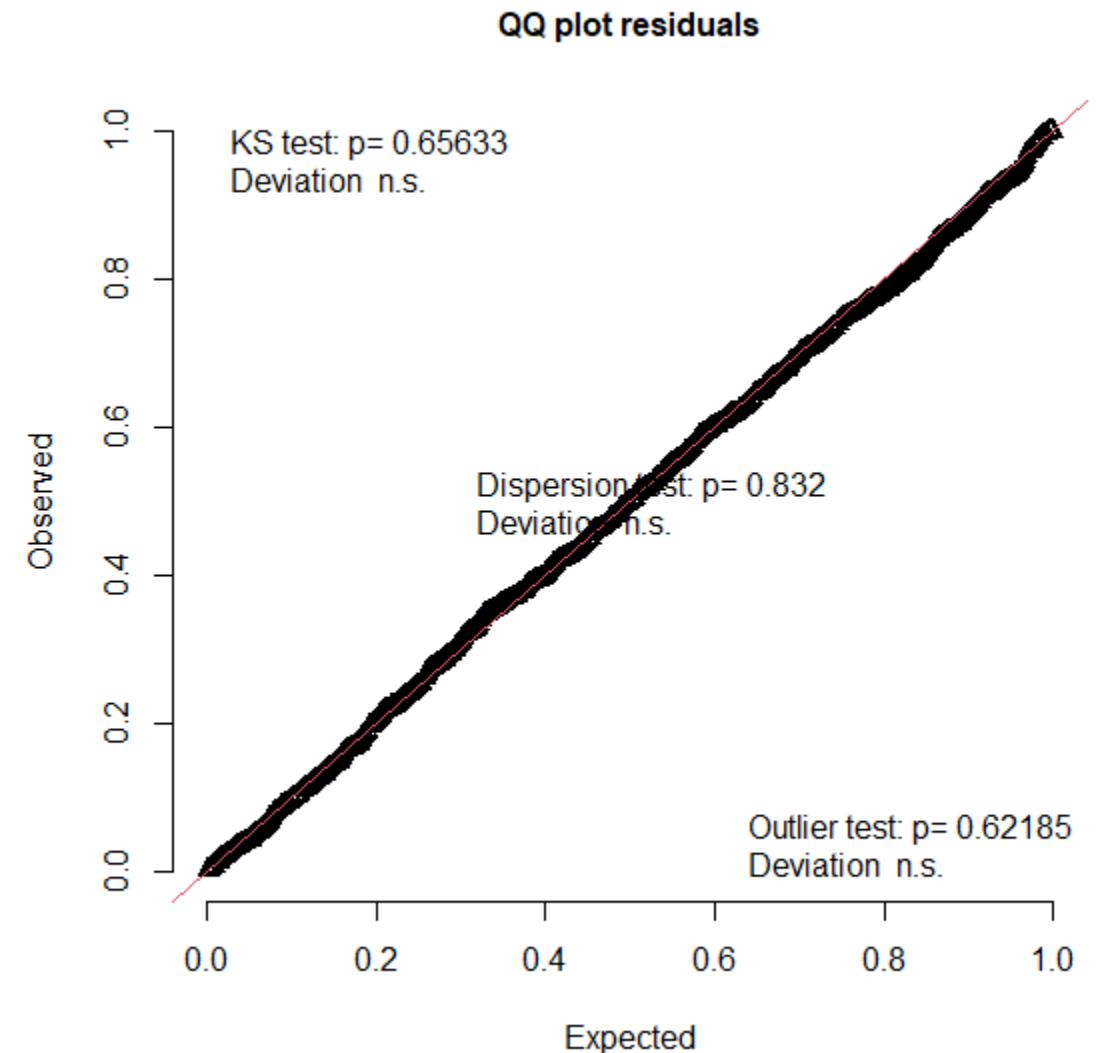
2. Plot the scaled residuals

```
plot(simulationOutput)
```



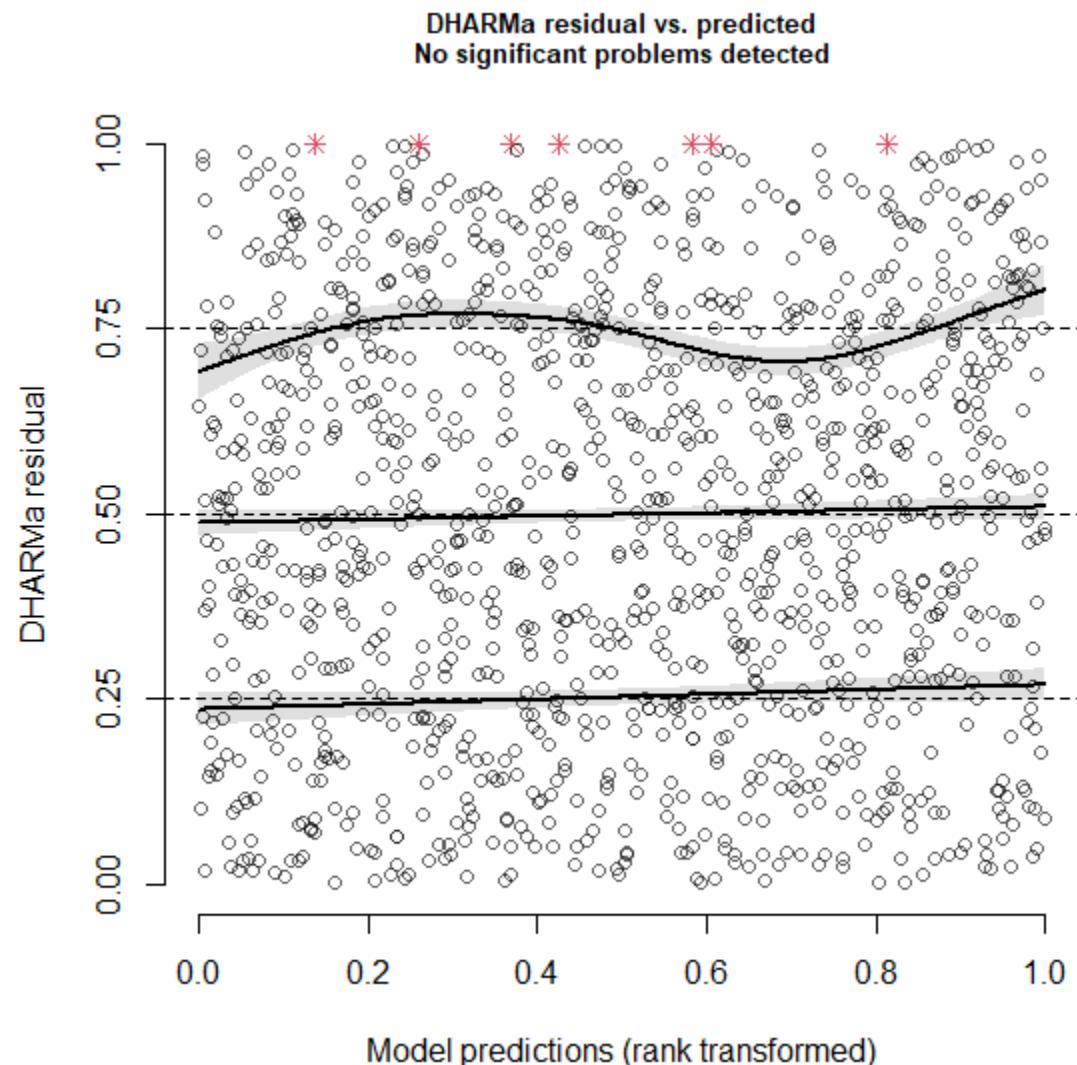
QQ plot

- If your data fits the model expect: Straight line on QQ plot of uniform distribution
- Visual check of uniform distribution (expect points to match line)
- Tests Uniformity with Kolmogorov-Smirnov (KS) test (Uniform distribution)
- Tests for Over/Underdispersion with Dispersion Test
- Tests for more Outliers than expected with Outlier test



Residuals vs. Predicted

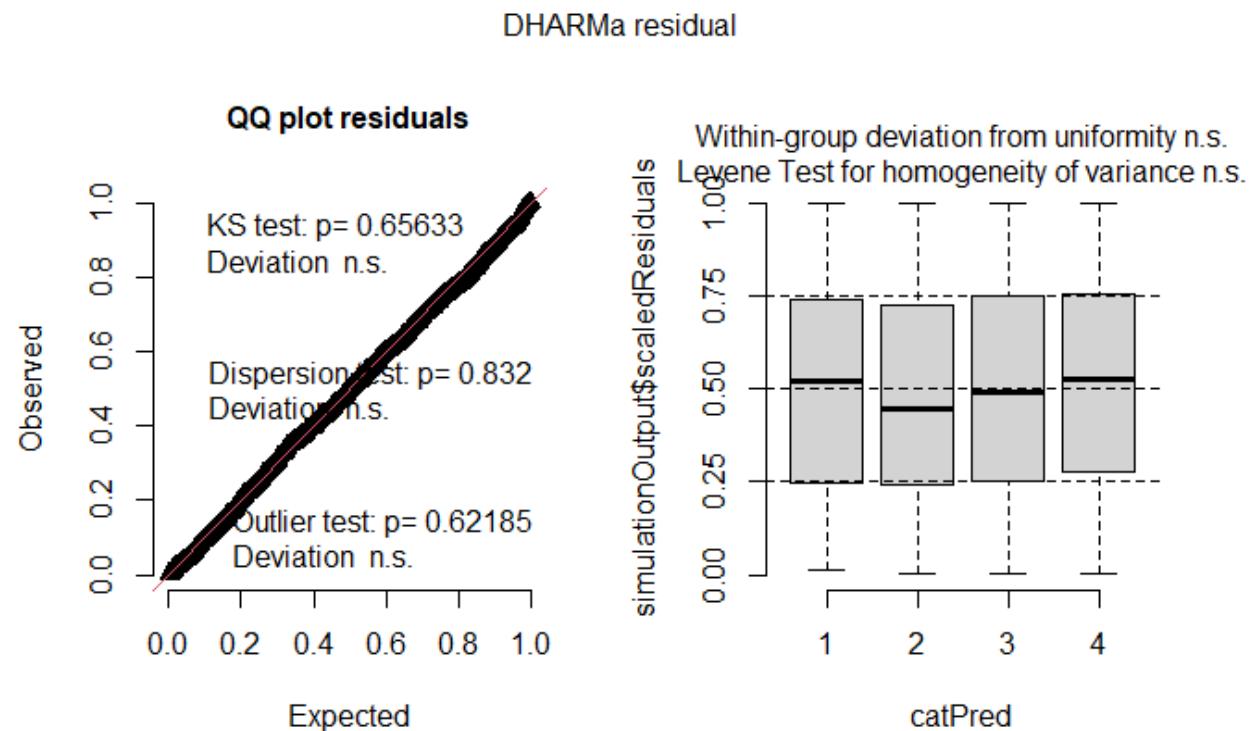
- If your data fits the model expect no patterns between residuals and model predictions
- Check distribution of residuals (visually and with quantile tests)
- Dotted lines show expected quantiles
- Black lines show simulated quantiles (want straight lines)
- Outliers show up as red stars



Workflow in DHARMA

3. Plot Residuals against predictors

```
plot(simulationOutput, form = ParasiteCod2$fArea)
```



Recognizing misspecification problems

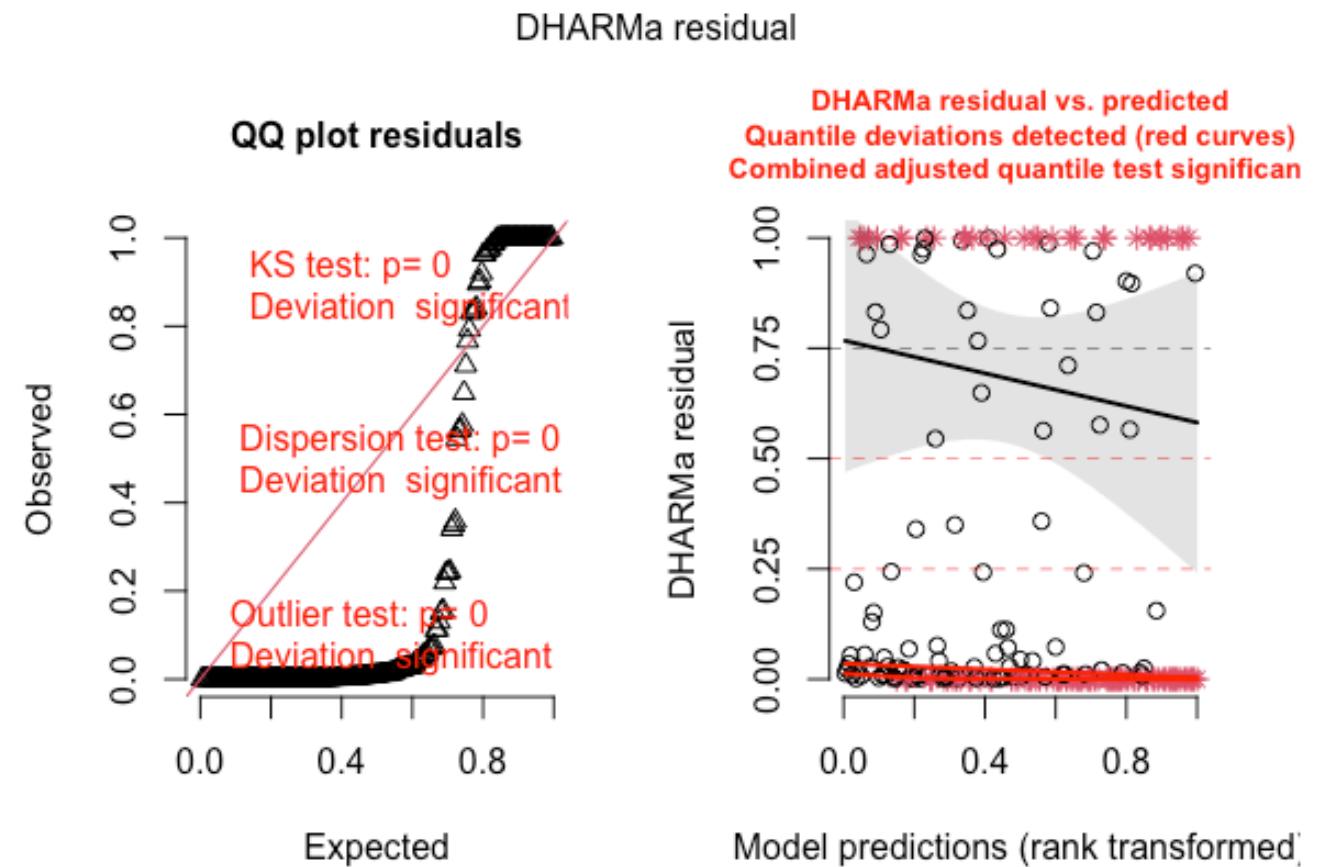
1. No residual pattern does not “prove” that the model is correct
2. Once a residual effect is statistically significant, look at the magnitude to decide if there is a problem.
3. A residual pattern does not mean that model is unusable

Important conclusion:

DHARMA only flags a difference between the observed and expected data - the user has to decide whether this difference is actually a problem for the analysis!

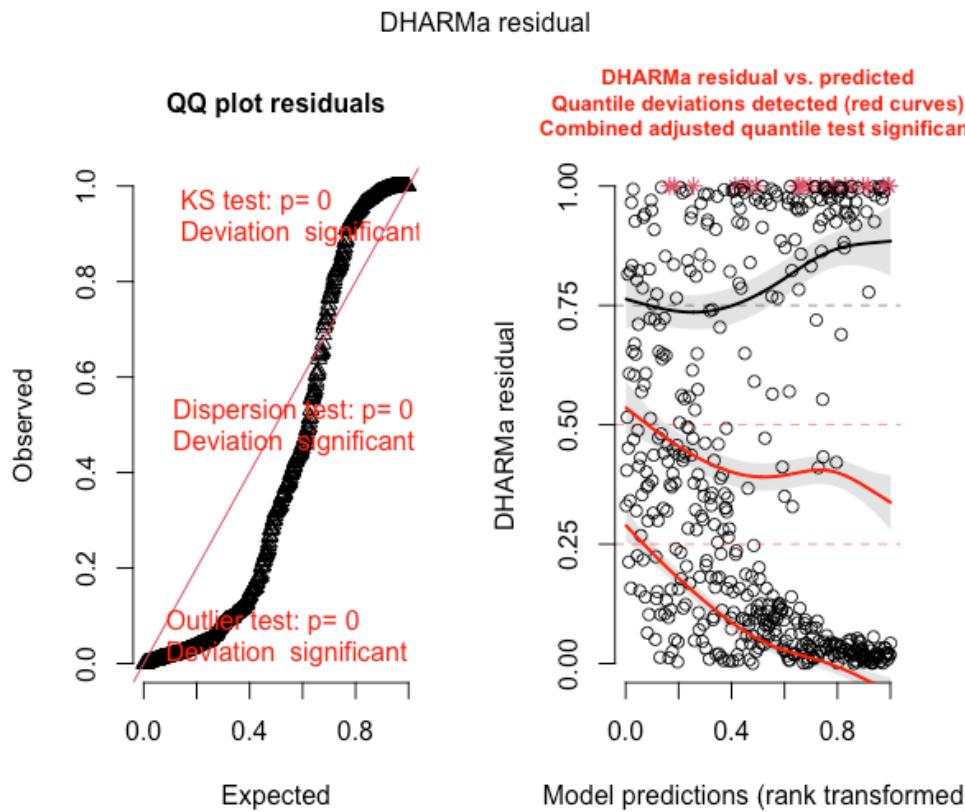
Overdispersion

- residual variance is larger than expected under the fitted model
- more residuals around 0 and 1, which means that more residuals are in the tail of distribution than would be expected under the fitted model.
- check your model for misfit by plotting residuals against all predictors

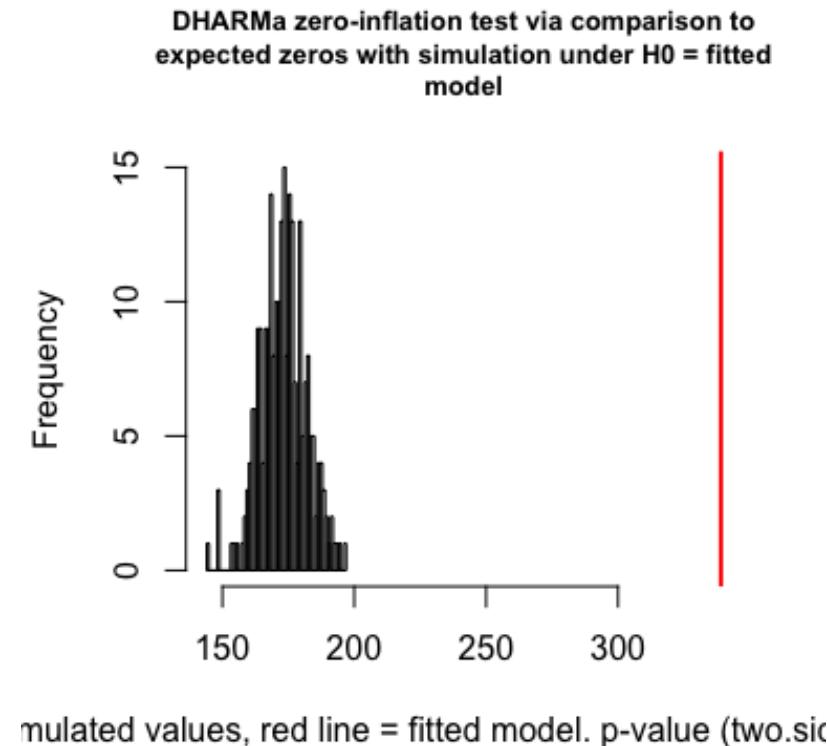


Zero-inflation

- Will look like overdispersion

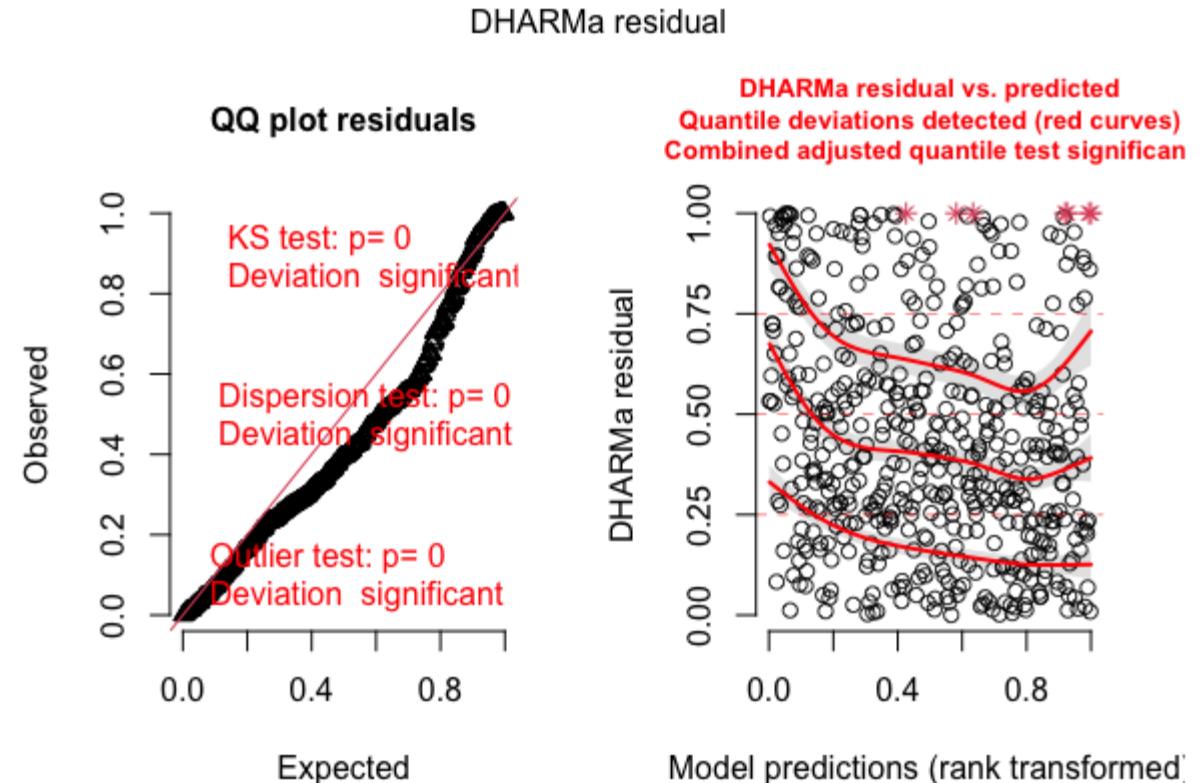


- compare the distribution of expected zeros in the data against the observed zero:
`testZeroInflation(simulationOutput)`



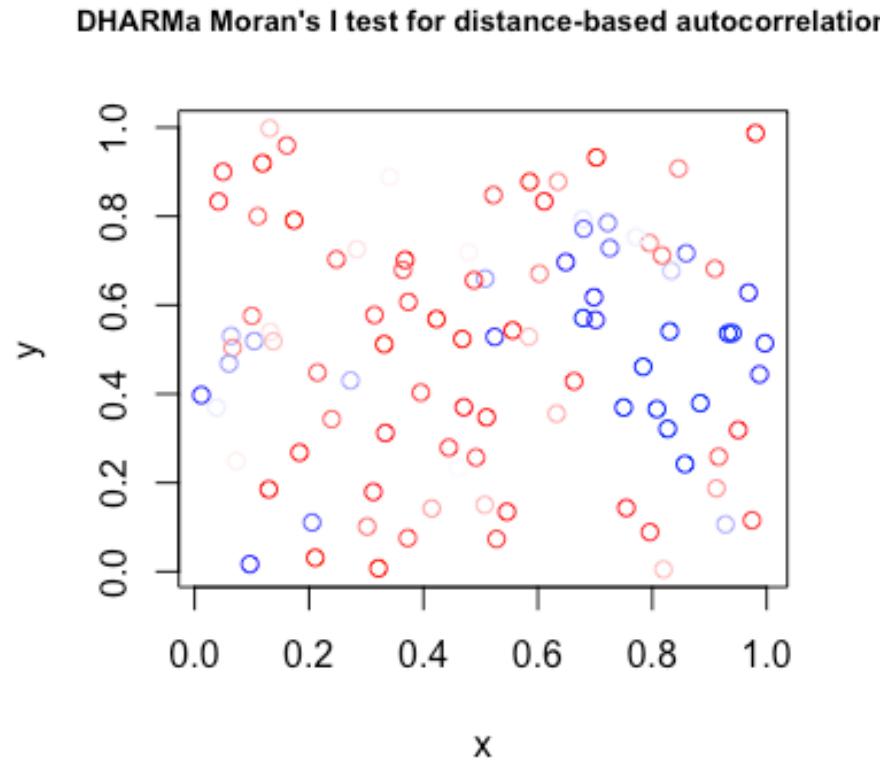
Heteroskedasticity

- Heteroscedasticity means that there is a systematic dependency of the dispersion / variance on another variable in the model.
- Options:
 - Quasi-likelihood model
 - Negative-binomial
 - Add observation-level random effect



Residual correlation structures (temporal, spatial, phylogenetic)

- `testTemporalAutocorrelation`: based on the Durbin-Watson test.
- `testSpatialAutocorrelation`: based on Moran's I, can also be used for generic distance functions.
- `testPhylogeneticAutocorrelation`: based on Moran's I test from `Moran.I` function on package `ape`.



To support the visual inspection of the residuals, the DHARMA package provides a number of specialized goodness-of-fit tests on the simulated residuals:

- `testUniformity()` - tests if the overall distribution conforms to expectations
- `testOutliers()` - tests if there are more simulation outliers than expected
- `testDispersion()` - tests if the simulated dispersion is equal to the observed dispersion
- `testQuantiles()` - fits a quantile regression or residuals against a predictor (default predicted value), and tests if this conforms to the expected quantile
- `testCategorical(simulationOutput, catPred = testData$group)` tests residuals against a categorical predictor
- `testZeroInflation()` - tests if there are more zeros in the data than expected from the simulations
- `testGeneric()` - test if a generic summary statistics (user-defined) deviates from model expectations
- `testTemporalAutocorrelation()` - tests for temporal autocorrelation in the residuals
- `testSpatialAutocorrelation()` - tests for spatial autocorrelation in the residuals. Can also be used with a generic distance function, for example to test for phylogenetic signal in the residuals

Summary: Model diagnostics with DHARMa



Generates simulated residuals by refitting the model with randomly simulated data based on the model's fitted parameters.



Plots the simulated residuals against fitted values, predictors, and other covariates to visualize potential issues. The goal is to check whether the residuals are uniformly distributed, with no systematic patterns.



Statistical tests like `testDispersion()` for overdispersion, `testQuantiles()` for heteroskedasticity, and `testZeroInflation()` for models with excess zeros, compare observed residuals to expected values, providing p-values to assess model fit and highlight specific areas of concern.