Comparing error distributions

## Comparing error distributions

The problem. For count response data of the sort we’re going to get in IID3 it’s inevitable that they’ll be overdispersed and a simple Poisson model will fail. In the past Steve and Roy have simply transformed the response using log(n+1), the +1 needed as you can’t log zero. A number of papers, mainly in the animal ecology literature, is very critical of log(n+1) amd says it should be avoided. Here the same dataset, from IID2, is pushed through several similar models and error distributions compared.

**Note**: When it comes to mixed models the non-INLA choice is primarily nlme, lme4 and glmmTMB. Both nlme and glmmTMB can handle correlation structures such as AR1. lme4 and glmmmTMB can handle several error structures. Only glmmTMB can handle multiple error structures and correlation structures in a mixed-effects framework. The following uses glmmTMB for all analyses, even where nlme would be fine, simply for ease of comparison of QQ plots. Model estimates for individual parameters are identical in glmmTMB and nlme for Gaussian models with no zero-inflation.

All these models are mixed-effects, with GP practice as random effect, as fixed-effects models meaningless for the data.

library(tidyr)  
library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':  
  
 filter, lag

The following objects are masked from 'package:base':  
  
 intersect, setdiff, setequal, union

library(lattice)  
library(ggplot2)  
library(gridExtra)

Attaching package: 'gridExtra'

The following object is masked from 'package:dplyr':  
  
 combine

library(DHARMa)

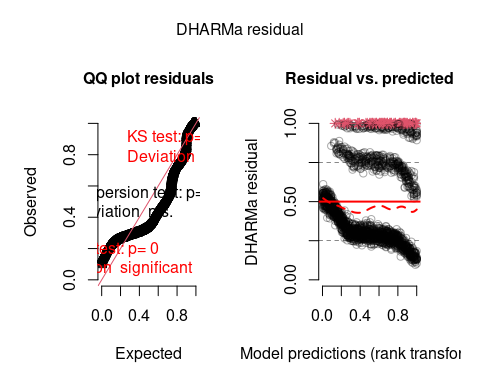
This is DHARMa 0.4.6. For overview type '?DHARMa'. For recent changes, type news(package = 'DHARMa')

library(glmmTMB)  
library(AICcmodavg)  
  
enum.data.tidy1 <- read.csv("data/Zero Inflated Case Data.csv")  
colnames(enum.data.tidy1)[1] <- "gp\_practice"  
glm.data <- enum.data.tidy1  
# glm.data <- enum.data.tidy1 %>%  
# group\_by(cont\_week) %>%  
# summarise(cases = sum(cases)) %>%  
# ungroup() %>%  
# mutate(sinwe=sin(cont\_week\*2\*pi/52),  
# coswe=cos(cont\_week\*2\*pi/52))

# Untransformed data

Gaussian, no transformation of response, no zero-inflation.

enum.glm1 <- glmmTMB(cases ~ week + coswe + sinwe + (1 | gp\_practice), zi=~0, data = glm.data)  
# summary(enum.glm1)  
enum.glm1.resid <- simulateResiduals(enum.glm1)  
plot(enum.glm1.resid)

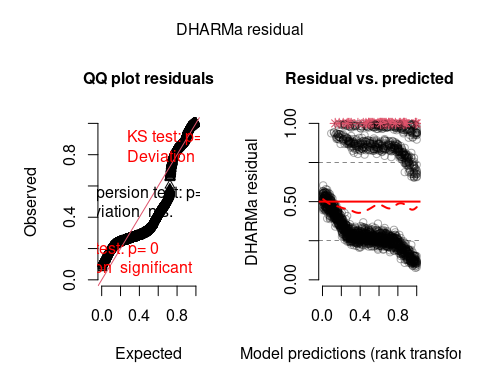


Significant deviation

# Log data cases + 1

Originally had planned to compare log(cases+1) vs a mixed-effects model with family=gaussian(link="log"). The latter works fine for a fixed-effects model via glm, but fails for lme4-based methods. Thus the following is only for log(cases+1

enum.glm2 <- glmmTMB(log(cases + 1) ~ week + coswe + sinwe + (1 | gp\_practice), zi=~0, data = glm.data)  
#summary(enum.glm1)  
enum.glm2.resid <- simulateResiduals(enum.glm2)  
plot(enum.glm2.resid)

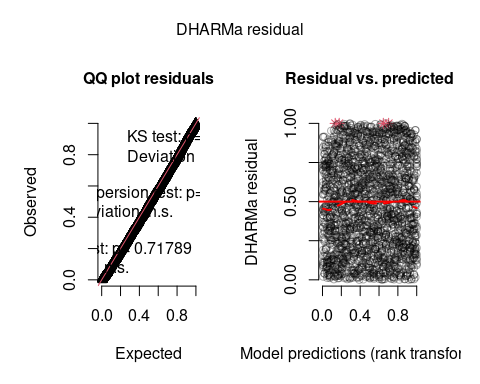


No obvious improvement over untransformed data.

# Poisson errors

The official way of doing it:

enum.glm3 <- glmmTMB(cases ~ week + coswe + sinwe + (1 | gp\_practice), zi=~0, data = glm.data, family = "poisson")  
enum.glm3.resid <- simulateResiduals(enum.glm3)  
#summary(enum.glm3)  
plot(enum.glm3.resid)



Steve definitely won’t like this, but Poisson errors are surprisingly good in this random-effects model. Note that Poisson error is dreadful on fixed-effect.

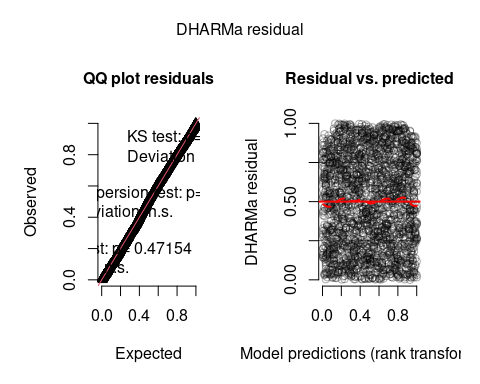
# Negative binomial

Some authors recommend negative binomial rather than simple Poisson, which if I’ve understood the literature properly (and I might not have done!) is actually a mixture distribution, allowing the Poisson mean to vary randomly following a Gamma distribution:

enum.glm4 <- glmmTMB(cases ~ week + coswe + sinwe + (1 | gp\_practice), zi=~0, data = glm.data, family = "nbinom2")  
enum.glm4.resid <- simulateResiduals(enum.glm4)  
summary(enum.glm4)

Family: nbinom2 ( log )  
Formula: cases ~ week + coswe + sinwe + (1 | gp\_practice)  
Data: glm.data  
  
 AIC BIC logLik deviance df.resid   
 3290.1 3324.3 -1639.1 3278.1 2195   
  
Random effects:  
  
Conditional model:  
 Groups Name Variance Std.Dev.  
 gp\_practice (Intercept) 0.8122 0.9012   
Number of obs: 2201, groups: gp\_practice, 31  
  
Dispersion parameter for nbinom2 family (): 3.05   
  
Conditional model:  
 Estimate Std. Error z value Pr(>|z|)   
(Intercept) -0.255875 0.181245 -1.412 0.158   
week -0.029646 0.002229 -13.302 < 2e-16 \*\*\*  
coswe 0.257072 0.054469 4.720 2.36e-06 \*\*\*  
sinwe -0.401133 0.057573 -6.967 3.23e-12 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

plot(enum.glm4.resid)



QQ plot diagnostics fairly similar to Poisson, although AIC is lower at 3313.5576099 for Poisson, compared to 3290.1419543 for negative binomial.

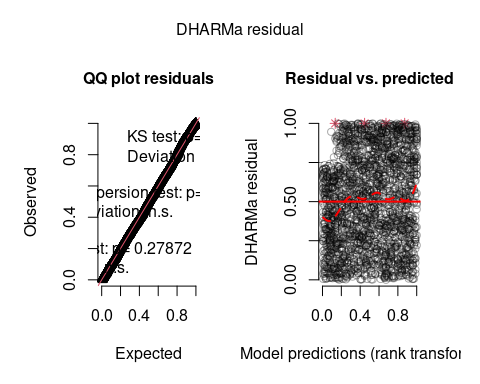
# Zero-inflated models

As far as I can tell, there are two main types, zero-adjusted (hurdle) models, and zero-inflated models. In theory it’s the latter we want: essentially a binomial (case of iid or not) then a Poisson (how many iid). Again requires glmmTMB. Using cont\_week only as the zero inflation parameter.

## First a hurdle (zero-adjusted) model

This is a truncated-Poisson hurdle model

enum.glm5 <- glmmTMB(cases ~ week + coswe + sinwe + (1 | gp\_practice),  
 zi=~1,  
 data = glm.data, family = truncated\_poisson)  
enum.glm5.resid <- simulateResiduals(enum.glm5)  
#summary(enum.glm5)  
plot(enum.glm5.resid)

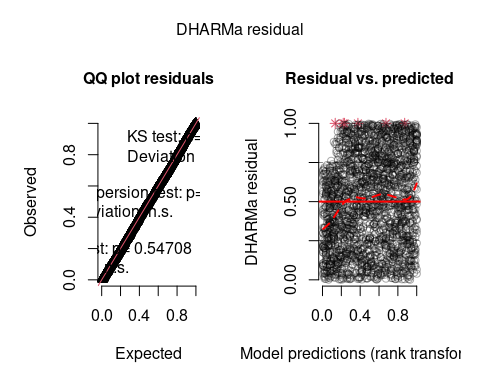


and a truncated negative binomial

enum.glm6 <- glmmTMB(cases ~ week + coswe + sinwe + (1 | gp\_practice),  
 zi=~1,  
 data = glm.data, family = truncated\_nbinom2)

Warning in finalizeTMB(TMBStruc, obj, fit, h, data.tmb.old): Model  
convergence problem; false convergence (8). See vignette('troubleshooting'),  
help('diagnose')

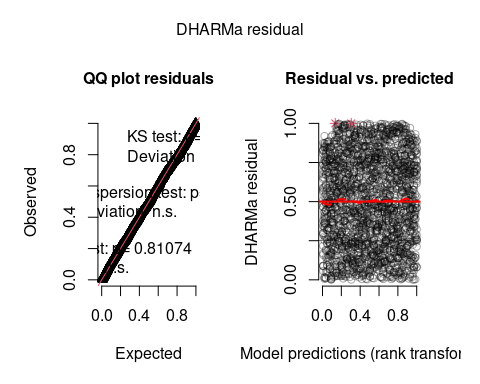
enum.glm6.resid <- simulateResiduals(enum.glm6)  
#summary(enum.glm6)  
plot(enum.glm6.resid)



## Second, zero-inflated (mixture) models

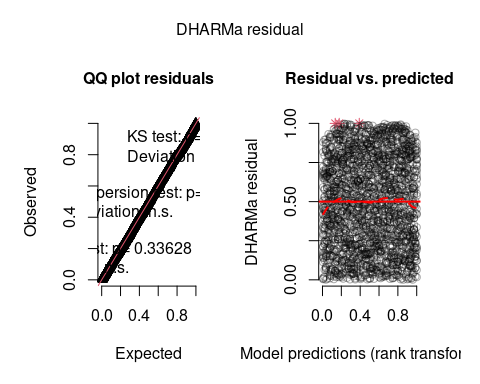
Zero-inflated Poisson

enum.glm7 <- glmmTMB(cases ~ week + coswe + sinwe + (1 | gp\_practice),  
 zi=~1,  
 data = glm.data, family = poisson)  
enum.glm7.resid <- simulateResiduals(enum.glm7)  
#summary(enum.glm7)  
plot(enum.glm7.resid)



and zero-inflated negative binomial:

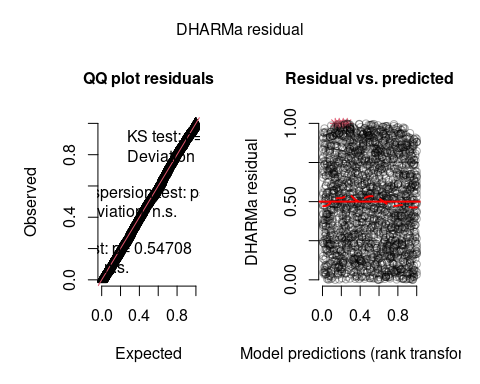
enum.glm8 <- glmmTMB(cases ~ week + coswe + sinwe + (1 | gp\_practice),  
 zi=~1,  
 data = glm.data, family = nbinom2)  
enum.glm8.resid <- simulateResiduals(enum.glm8)  
#summary(enum.glm8)  
plot(enum.glm8.resid)



# The form of the data

The “denominator” of our number of cases actually varies each week, as the number of GP practices changes. I guess that strictly-speaking this should be included as an offset. We also ignore any autocorrelation in the data. For example, including the number of GP practices as an offset, this needs to be logged, as Poisson by default uses a log-link function:

enum.glm9 <- glmmTMB(cases ~ week + coswe + sinwe + offset(log(practice\_recruitment)) + (1 | gp\_practice),  
 zi=~1,  
 data = glm.data, family = nbinom2)  
enum.glm9.resid <- simulateResiduals(enum.glm9)  
#summary(enum.glm9)  
plot(enum.glm9.resid)



# Conclusions

The Poisson models (of different forms) perform surprisingly well, and better than log(cases+1), which wasn’t anticipated. It is difficult to compare all the QQ plots, so some final insigts can be gained by comparing AIC scores:

Model selection based on AICc:  
  
 K AICc Delta\_AICc AICcWt Cum.Wt LL  
negbinom 6 3290.18 0.00 0.57 0.57 -1639.07  
zinbinom 7 3291.14 0.96 0.35 0.93 -1638.54  
zipoiss 6 3294.24 4.06 0.07 1.00 -1641.10  
poisson 5 3313.58 23.40 0.00 1.00 -1651.78  
zinbinom\_off 7 3326.01 35.83 0.00 1.00 -1655.98  
hrdlpoiss 6 3729.96 439.78 0.00 1.00 -1858.96  
hrdlnbinom 7 3731.97 441.79 0.00 1.00 -1858.96

Overall the negative binomial, and zero-inflated negative binomial have lowest AIC.