Modelling Bd load against Jliv; handling zero-inflated dataset

MYC

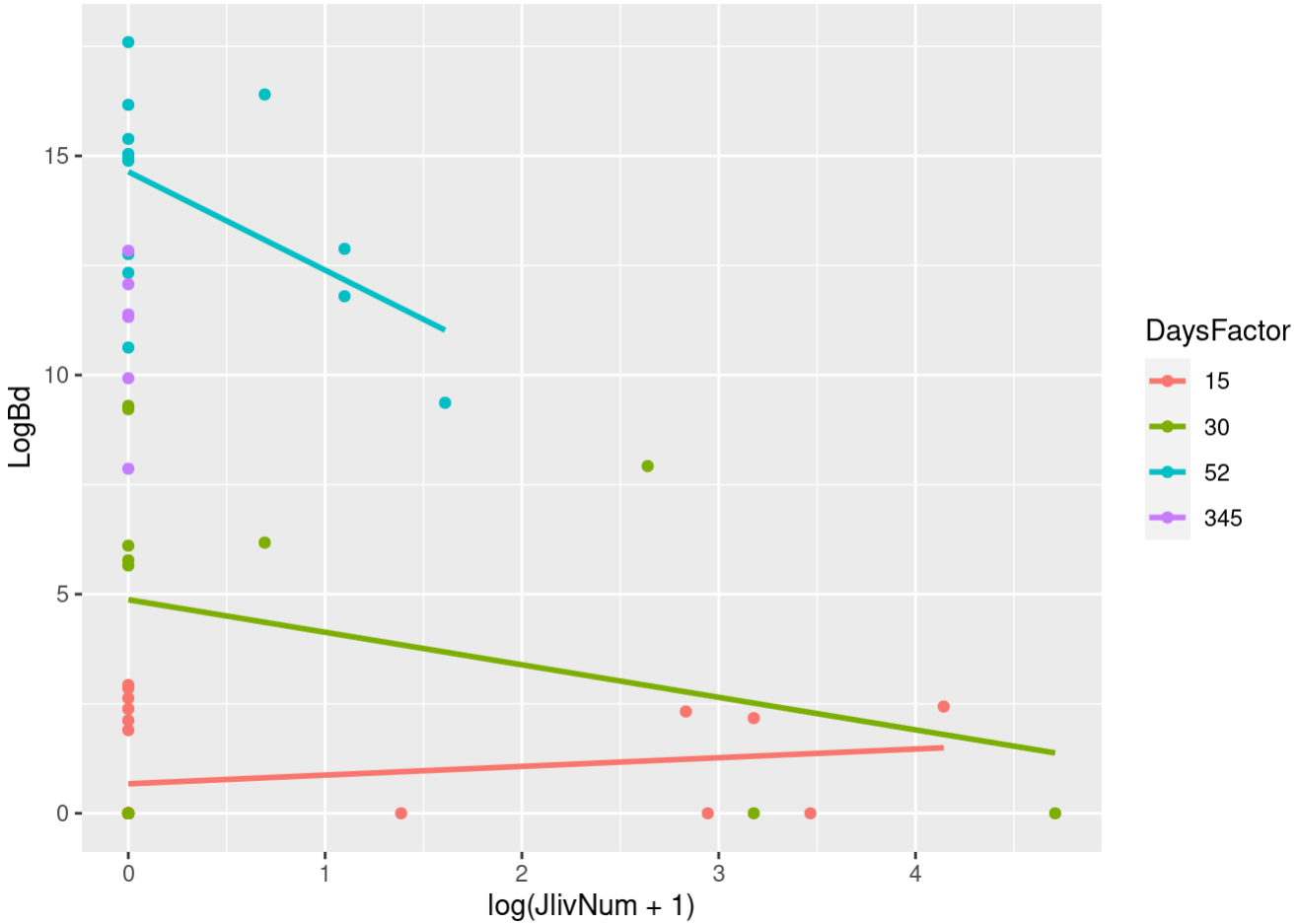
2022-10-18

```
# Load in dataset; adjust some Bd/Jliv numbers
dat <- read.csv("FigDataBd5.csv") %>%
  mutate(DaysFactor = factor(DaysSince)
         , DaysContinuous = DaysSince
         ,logJliv=log(JlivNum+1)
         ,Bd=round(exp(LogBd)-1))
```

Visual exploration

First, let's look at the data:

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



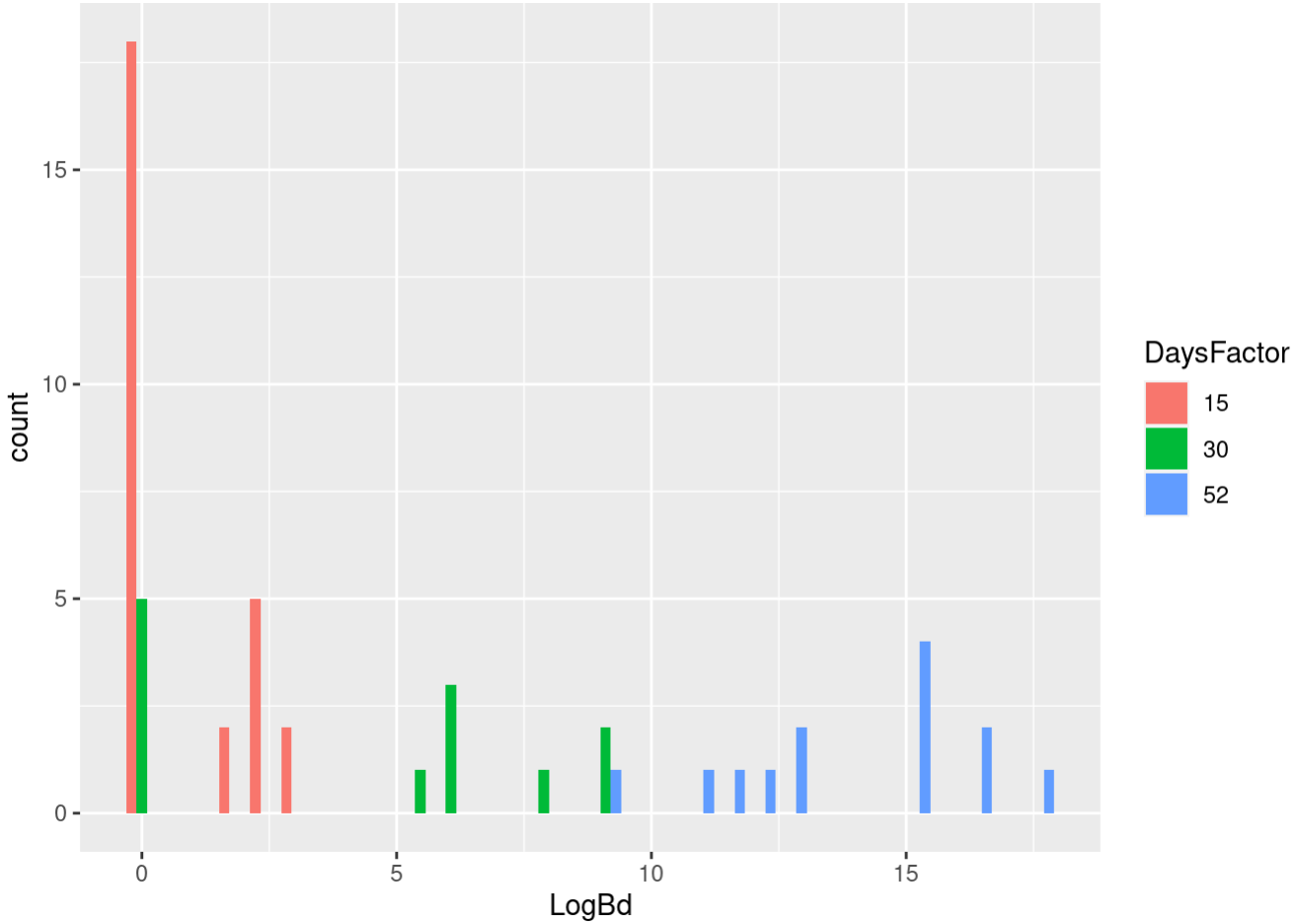
So it turns out that day 345 doesn't have any Jliv on any toads. This means that it is not actually useful for testing whether Jliv results in less Bd. Let's remove this day.

```
dat_filt <- dat %>% filter(DaysSince!=345)
```

Now, let's look at Bd distribution.

```
dat_filt %>%
  ggplot() +
  geom_histogram(aes(x=LogBd, group=DaysFactor, fill=DaysFactor), position = "dodge")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



It also turns out that the data is not as zero-inflated as we might expect! A lot of the zeros are in the first sampling point, and the distribution within each day is actually pretty reasonable. The difference in Bd load between time points, and the fact that the relationship between Bd load and Jliv changes with each time point, makes me feel that we *should* include time. We can include time account for differences in Bd load at each time point; but then also to see if the slope between Jliv and Bd changes as time goes on.

Let's try three approaches:

- Linear Regression
- Zero-inflated negative binomial (ZINB)
- Quantile regression

In each model, we will include Jliv*Days (days as factor). In two-part models, we will also include DaysSince (numeric) as a predictor for Bd prevalence (0/1).

Linear Regression

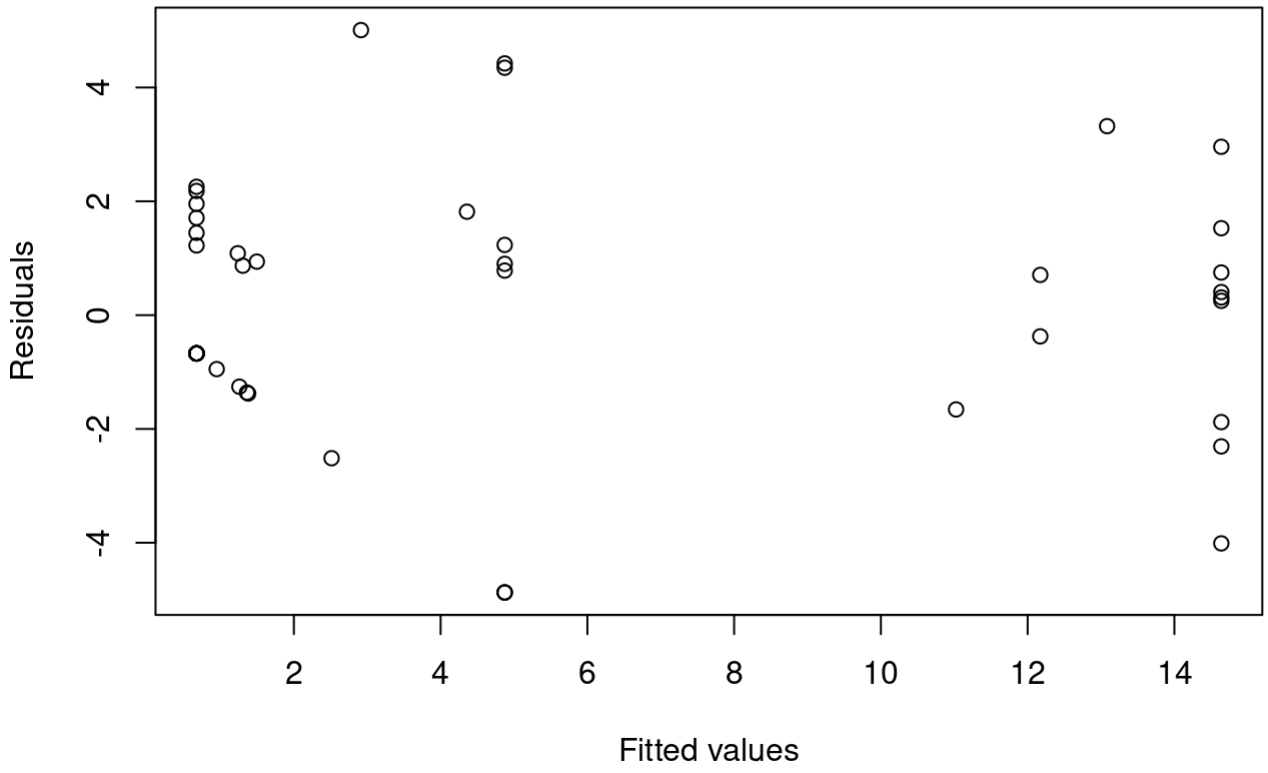
```
##### Normal regression #####
linear_ml <- lm(LogBd ~ logJliv*DaysFactor, data=dat_filt)
summary(linear_ml)
```

```
##
## Call:
## lm(formula = LogBd ~ logJliv * DaysFactor, data = dat_filt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8735 -0.6733 -0.6733  1.2266  5.0086
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.6733     0.4879   1.380   0.1742
## logJliv                0.1984     0.3332   0.596   0.5544
## DaysFactor30           4.2002     0.9032   4.650 2.81e-05 ***
## DaysFactor52          13.9662     0.8867  15.751  < 2e-16 ***
## logJliv:DaysFactor30  -0.9408     0.5344  -1.761   0.0850 .
## logJliv:DaysFactor52  -2.4444     1.1876  -2.058   0.0452 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.259 on 46 degrees of freedom
## Multiple R-squared:  0.8678, Adjusted R-squared:  0.8534
## F-statistic: 60.39 on 5 and 46 DF,  p-value: < 2.2e-16
```

Okay! So this tells us that Bd load is not predicted by Jliv on Day 15 (logJliv; p=0.5544), but that as time goes on Jliv becomes a better (and eventually significant) predictor of Bd load (logJliv:DaysFactor52; p=0.0452). Also, There is an obvious effect of time (DaysFactor30, DaysFactor52) where Bd load is higher at later time points.

Let's look at residuals of model:

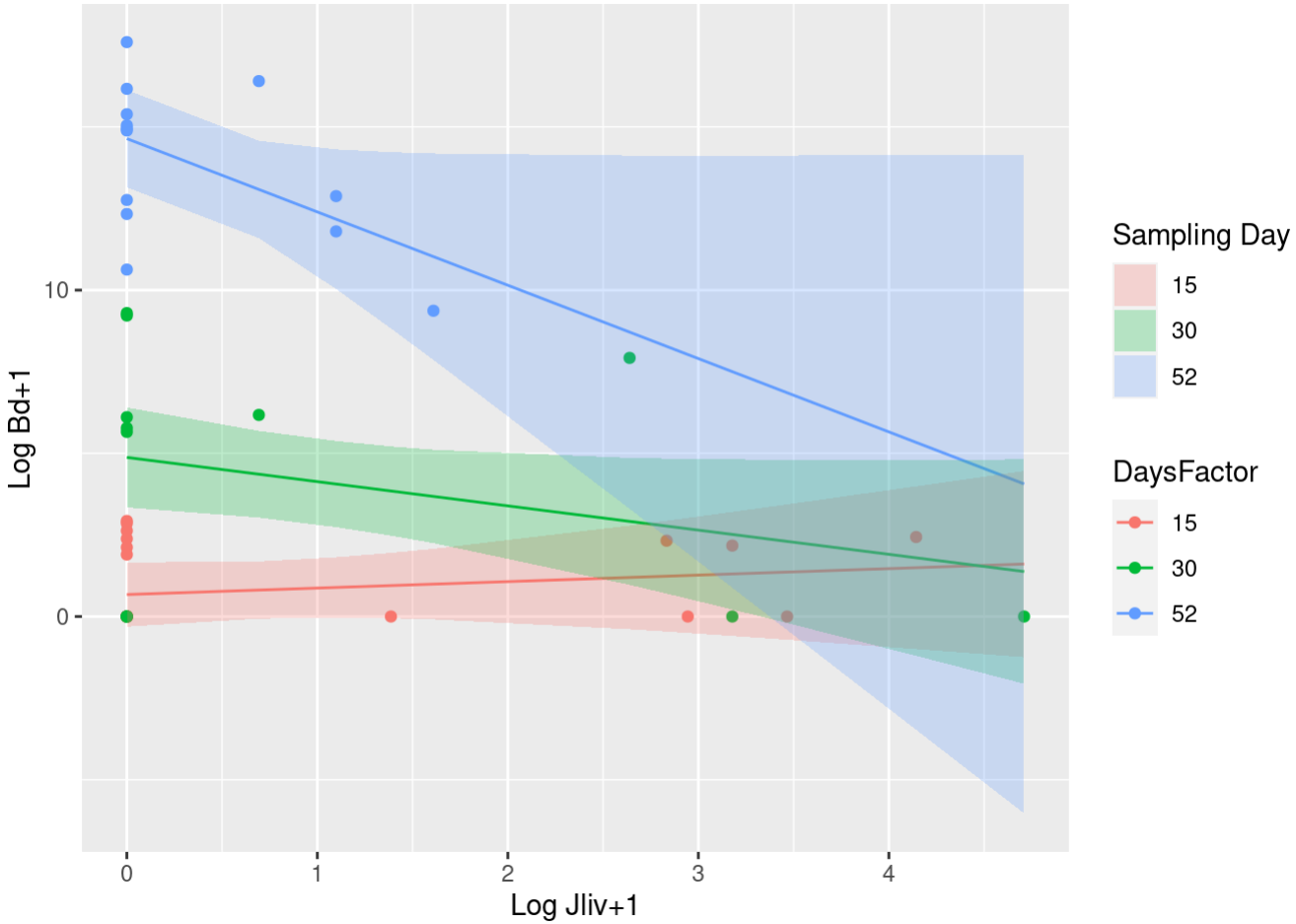
```
# Plotting residuals
plot(linear_ml$fitted.values, linear_ml$residuals, xlab="Fitted values", ylab="Residuals")
```



The fit is honestly not that bad. Most of the zeros are actually associated with day 15, and day 15 is generally low. So the data are not as heterskedastic as we'd feared. Let's overlay the model predictions + confidence intervals on the plot:

```
newdata <- data.frame(logJliv=rep(dat_filt$logJliv, 3)
                      , DaysFactor = rep(unique(dat_filt$DaysFactor),
                                         each=length(dat_filt$logJliv)))
linear_ml_predict <- predict(linear_ml, newdata = newdata,interval = "confidence")
LinearModelFit <- cbind(newdata, linear_ml_predict)

ggplot(dat_filt) +
  geom_point(aes(x=logJliv, y=LogBd, col=DaysFactor)) +
  geom_line(data=LinearModelFit, aes(x=logJliv, y=fit, col=DaysFactor)) +
  geom_ribbon(data=LinearModelFit, aes(x=logJliv, ymin=lwr, ymax=upr, fill=DaysFactor), alpha=0.25) +
  ylab("Log Bd+1") + xlab("Log Jliv+1") + labs(fill="Sampling Day")
```



Zero-inflated negative binomial

Now, let's try a model that accounts for the increased zeros in Bd count.

```
##### Zero inflated negative binomial #####
# logit part: time as continuous factor
# negbin part: Jliv and time, interacting

zinb_ml <- zeroinfl(Bd ~ logJliv*DaysFactor | DaysContinuous,
               data = dat_filt, dist = "negbin", link="log")
summary(zinb_ml)
```

```
##
## Call:
## zeroinfl(formula = Bd ~ logJliv * DaysFactor | DaysContinuous, data = dat_filt,
##     dist = "negbin", link = "log")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -0.6939 -0.4864 -0.3919  0.3148  3.4994
##
## Count model coefficients (negbin with log link):
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)           2.17512    0.54122   4.019 5.85e-05 ***
## logJliv              -0.01669    0.27265  -0.061   0.9512
## DaysFactor30          6.09272    0.78489   7.762 8.33e-15 ***
## DaysFactor52         13.95398    0.74482  18.735  < 2e-16 ***
## logJliv:DaysFactor30 -0.22565    0.58250  -0.387   0.6985
## logJliv:DaysFactor52 -2.07250    1.03735  -1.998   0.0457 *
## Log(theta)           -0.61170    0.26814  -2.281   0.0225 *
##
## Zero-inflation model coefficients (binomial with log link):
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.37247    0.43878   0.849   0.3959
## DaysContinuous -0.05819    0.02286  -2.545   0.0109 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 0.5424
## Number of iterations in BFGS optimization: 13
## Log-likelihood:  -331 on 9 Df
```

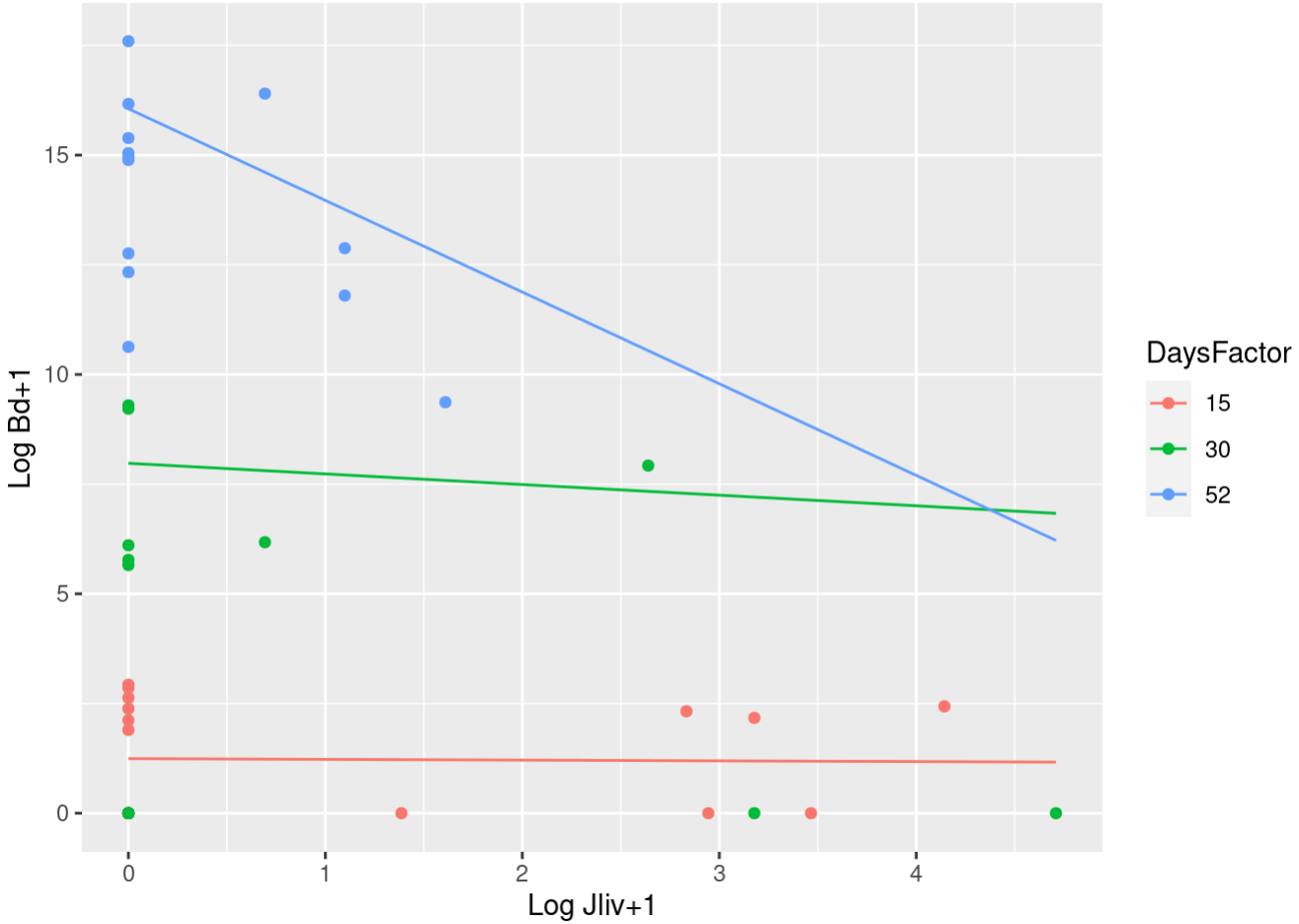
What we see here, is:

1. Bd prevalence (0/1) is higher as time goes on (DaysContinuous; p = 0.01)
2. Jliv has no effect on Bd load on Day 15 (logJliv; p=0.9512)
3. The effect of Jliv has a significant effect on Bd load (relative to day 15) on Day 52 (logJliv:DaysFactor52, p=0.0457)

This tells us the same thing as the linear model: that as time goes on, the “effect” of Jliv on Bd load becomes more pronounced. Below, I’ve plotted the model fit onto data.

```
newdata <- data.frame(logJliv=rep(dat_filt$logJliv, 3)
                      , DaysFactor = rep(unique(dat_filt$DaysFactor),
                                         each=length(dat_filt$logJliv))
                      , DaysContinuous = rep(unique(dat_filt$DaysContinuous),
                                         each=length(dat_filt$logJliv)))
zinb_ml_predict <- predict(zinb_ml, newdata = newdata)
ZINBModelFit <- cbind(newdata, fit=log(zinb_ml_predict))

ggplot(dat_filt) +
  geom_point(aes(x=logJliv, y=LogBd, col=DaysFactor)) +
  geom_line(data=ZINBModelFit, aes(x=logJliv, y=fit, col=DaysFactor)) +
  ylab("Log Bd+1") + xlab("Log Jliv+1") + labs(fill="Sampling Day")
```



Quantile Regression

Finally, let's try a quantile regression, which uses the "limits" of the data to define the relationship between Bd and Jliv. This is useful because it treats zeros as just "part of the data below the quantile line", and doesn't assume anything about its distribution. Here, we use a "median", which should be similar to the effect of "mean" (a classic linear regression)

```
### Quantile Regression #####
# using a quantile of 90
quantreg_ml <- rq(LogBd ~ logJliv*DaysFactor, data=dat_filt, tau = 0.50)
summary(quantreg_ml, se="boot") # Use bootstrapping to estimate confidence
```

```
##
## Call: rq(formula = LogBd ~ logJliv * DaysFactor, tau = 0.5, data = dat_filt)
##
## tau: [1] 0.5
##
## Coefficients:
##                      Value    Std. Error t value  Pr(>|t|)
## (Intercept)           0.00000  0.25694    0.00000  1.00000
## logJliv               0.58783  0.32784    1.79303  0.07955
## DaysFactor30          5.77455  2.50145    2.30848  0.02552
## DaysFactor52         14.95013  0.90937   16.44012  0.00000
## logJliv:DaysFactor30 -1.81398  1.51912   -1.19410  0.23856
## logJliv:DaysFactor52 -3.45694  1.52516   -2.26661  0.02816
```

We see the same story here: Jliv is associated with less Bd, but this effect is really only statistically significant at later time points (logJliv:DaysFactor52; p=0.036). Early on, there isn't an effect of Jliv (logJliv; p = 0.068) but it's hard to say whether this is because there isn't enough Bd yet to see a difference, or if Jliv impacts long-term outcomes for Bd infection. The plot is below:

```
newdata <- data.frame(logJliv=rep(dat_filt$logJliv, 3)
                      , DaysFactor = rep(unique(dat_filt$DaysFactor),
                                         each=length(dat_filt$logJliv)))
quantreg_ml_predict <- predict(quantreg_ml, newdata = newdata, interval = "confidence")
```

```
## Warning in summary.rq(object, cov = TRUE, ...): 1 non-positive fis
```

```
quantregModelFit <- cbind(newdata, quantreg_ml_predict)

ggplot(dat_filt) +
  geom_point(aes(x=logJliv, y=LogBd, col=DaysFactor)) +
  geom_line(data=quantregModelFit, aes(x=logJliv, y=fit, col=DaysFactor)) +
  geom_ribbon(data=quantregModelFit, aes(x=logJliv, ymin=lower, ymax=higher, fill=DaysFactor), alpha=0.25) +
  ylab("Log Bd+1") + xlab("Log Jliv+1") + labs(fill="Sampling Day")
```