S632 Takehome - Exam 2

Erik Parker

March 3, 2018

1. Using only the methods and techniques learned so far in S632 (and, methods learned in 631 if you consider this appropriate), propose an appropriate model.

With the exception of the class instructor, I have not had any form of communication about this exam with any other individual (including other students, teaching assistants, instructors, etc.)

Signed: Erik Parker

We first saw from a summary of the data that there were many 0 counts of weeds, and that the distribution of observed humidity values is bimodal with a gap between 52 and 77%. This first suggested that the response might be zero inflated, and also that the treatment of humidity might be best as a categorical variable for high and low values or that a quadratic term for *Moist* might be necessary. First, to determine if a categorical variable for humidity would better predict the response, two models were compared. The reduced model had a *Moist* reduced to a two level variable with levels high and low, and the full model had the unaltered continuous *Moist* variable. A goodness of fit test comparing these two models found that we could safely reject the null hypothesis that the reduced model fit better than the full model.
Next, to determine if the the weird shape of the *Moist* variable needed a quadratic term to accurately represent it, a model with a quadratic term was compared to one without. It was seen, using a goodness of fit test, that we couldn't reject the null hypothesis that the model without the quadratic term adaquately fit the data. To then determine if the response was zero inflated, a variety of standard poisson models were compared to find the best among them in order to generate predicted values. Of these poisson models, it was determined using goodness of fit testing that the best one was a model regressing *Weeds* on *Moist*. Using this starting model, a predicted vs. observed plot was generated. This plot showed that there were many more observed zeroes (17) than expected (13). To account for this, the hurdle command was used to construct two models - one with only the regressor *Moist*, which earlier was shown to be the most influential individual regressor, and one with both *Moist* and *Days*. A likelihood ratio test comparing both models resulted in the conclusion that the reduced model, with only *Moist* was sufficient to explain the response. After this model was identified it was used to generate predicted values to draw another predicted vs. observed plot. The plot generated using this new model was seen to be much better - though the number of observed fields with one type of weed was still quite a bit higher than predicted. Despite this shortcoming, this model otherwise seems to be the best I am able to generate for these data. So the appropriate model here seems to be $hurdle(Weeds \sim Moist)$, as given in mz1.

2. Interpret at least 2 of the coefficient estimates.

First, I will interpret zero_Moist, which corresponds to the coefficient estimate for the hurdle (binomial) part of the model. This estimate tells us that for every unit increase in *Moist*, or every percentage increase in humidity, we see that the odds of finding at least one type of weed change by a factor of $e^{\hat{\beta}}$ - so here our odds increase by a factor of 1.031, or 3.1%. The coefficient estimate for count_Moist corresponds to the poisson part of the model, and here we see that when we assume there is already one type of weed present, for every unit increase in *Moist* the expected number of type of weeds increases by (1.0319-1)*100% = 3.19%.

3. Assume that your lawn is about 500ft^2 in size with 83% humidity level and you have not used any weed-control products for a whole year.

a) How many types of weeds would you expect to find?

Based on having a lawn with 83% humidity and no weed control product usage for a year, we would expect to see 3.373 types of weeds, so ~3 types of weeds, in a 500 sq ft lawn.

b) What is the probability you are not going to observe any weeds?

The probability of observing no weeds is 20.1% under these conditions.

c) What is the probability you may observe more than 3 types of weeds?

The probability of seeing more than three types of weeds under these conditions is 48.4%.

4. Write the R code necessary to obtain the coefficient estimates for the model glm(Weeds ~ Days*Moist, family = "poisson") without using glm and show that your results are equivalent to those using glm.

We can see in part 4 of the appendix that through calculating the estimated coefficients by "hand" using the Newton-Raphson method, we see values of -1.410263^{e+00}, $-3.877390e^{-03}$, $3.264673e^{-02}$, *and* $4.211656e^{-05}$ for the intercept, *Days*, *Moist* and the *Days:Moist* interaction respectively. The same exact coefficient estimates are seen from the summary of the poisson model constructed using the *glm()* command.

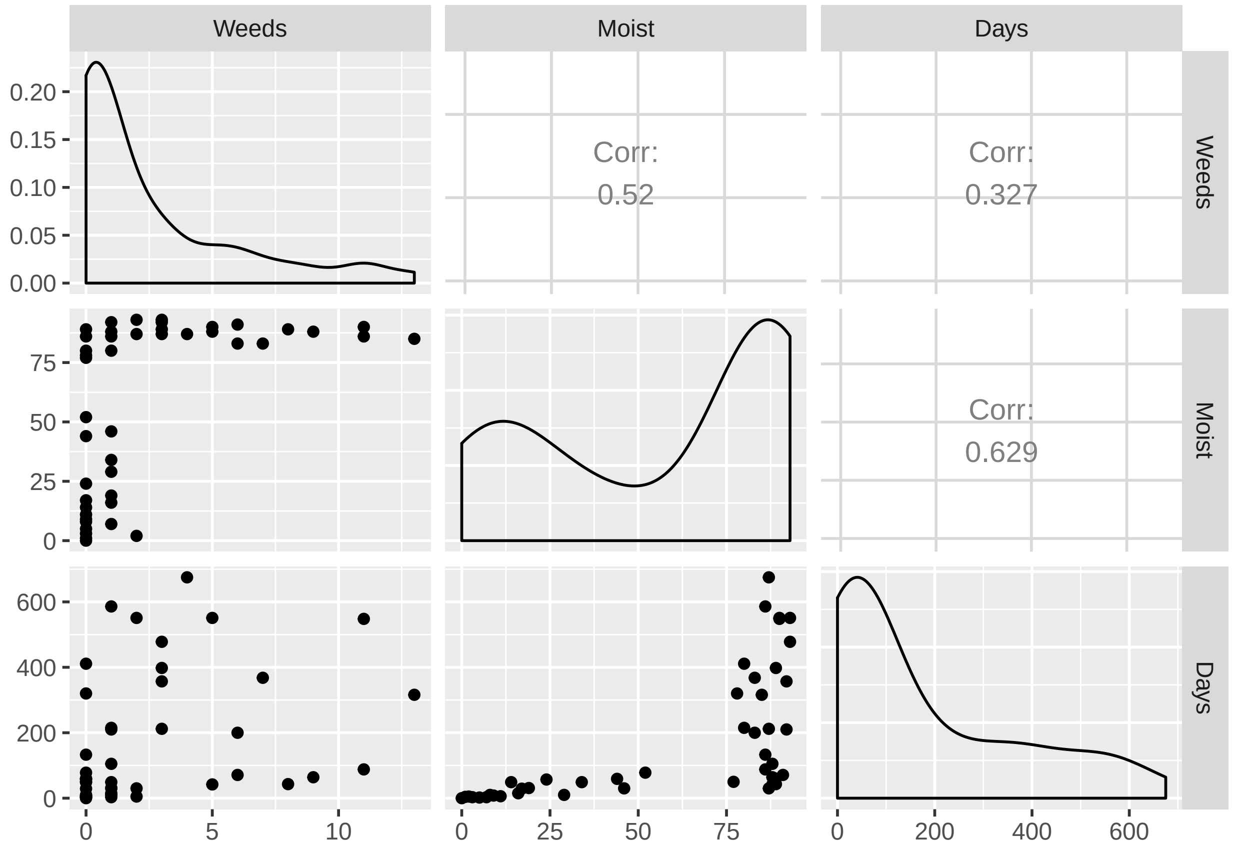
Appendix

1.

```r
rm(list = ls())
library(alr4)
library(ggplot2)
library(faraway)
library(pscl)
library(GGally)

weed <- read.table("takehome.txt")

summary(weed)
```

```
##      Weeds            Moist            Days      
##  Min.   : 0.000   Min.   : 0.00   Min.   :  0.0  
##  1st Qu.: 0.000   1st Qu.:17.00   1st Qu.: 29.0  
##  Median : 1.000   Median :80.00   Median : 60.0  
##  Mean   : 2.511   Mean   :57.73   Mean   :167.3  
##  3rd Qu.: 3.000   3rd Qu.:88.00   3rd Qu.:316.0  
##  Max.   :13.000   Max.   :93.00   Max.   :675.0
```

```r
# So, ranges of days and moist are quite large and both include zero. Also,
# lots of 0 counts of weeds. Probably should consider using a zero inflated
# poisson model.

ggpairs(weed)
```



```
# Moist looks almost bimodal, the lack of values from 52-77 is clear, but is
# it important?

weed$moistcat <- ifelse(weed$Moist < "52", "low", "high")

m1 <- glm(Weeds ~ ., data = weed, family = poisson)
m2 <- glm(Weeds ~ Moist * Days, data = weed, family = poisson)
m3 <- glm(Weeds ~ Moist, data = weed, family = poisson)
# m4 <- glm.nb(Weeds ~ Moist*Days, data = weed) m5 <- glm.nb(Weeds ~ Moist,
# data = weed)
m6 <- glm(Weeds ~ moistcat, data = weed, family = poisson)
m7 <- glm(Weeds ~ Moist + I(Moist^2), data = weed, family = poisson)


pchisq(deviance(m1), df.residual(m1), lower = FALSE)
```

```
## [1] 1.528071e-08
```

```
pchisq(deviance(m2), df.residual(m2), lower = FALSE)
```

```
## [1] 1.659392e-08
```

```
pchisq(deviance(m3), df.residual(m3), lower = FALSE)
```

```
## [1] 4.11955e-08
```

```
pchisq(deviance(m6), df.residual(m6), lower = FALSE)
```

```
## [1] 1.247024e-11
```

```r
pchisq(deviance(m7), df.residual(m7), lower = FALSE)
```

```
## [1] 3.351605e-08
```

```r
pchisq(deviance(m3) - deviance(m1), df.residual(m3) - df.residual(m1), lower = FALSE)
```

```
## [1] 0.9376794
```

```r
pchisq(deviance(m1) - deviance(m2), df.residual(m1) - df.residual(m2), lower = FALSE)
```

```
## [1] 0
```

```r
pchisq(deviance(m3) - deviance(m2), df.residual(m3) - df.residual(m2), lower = FALSE)
```

```
## [1] 0.8277112
```

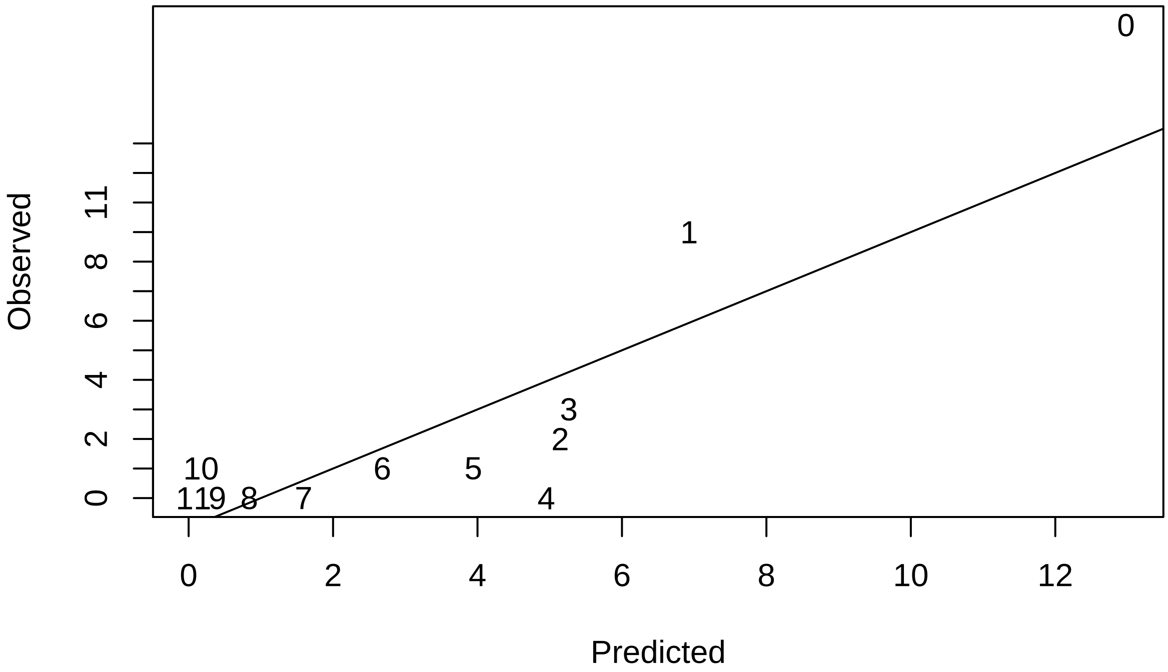
```r
pchisq(deviance(m6) - deviance(m3), df.residual(m6) - df.residual(m3), lower = FALSE)
```

```
## [1] 0
```

```r
pchisq(deviance(m3) - deviance(m7), df.residual(m3) - df.residual(m7), lower = FALSE)
```

```
## [1] 0.333436
```

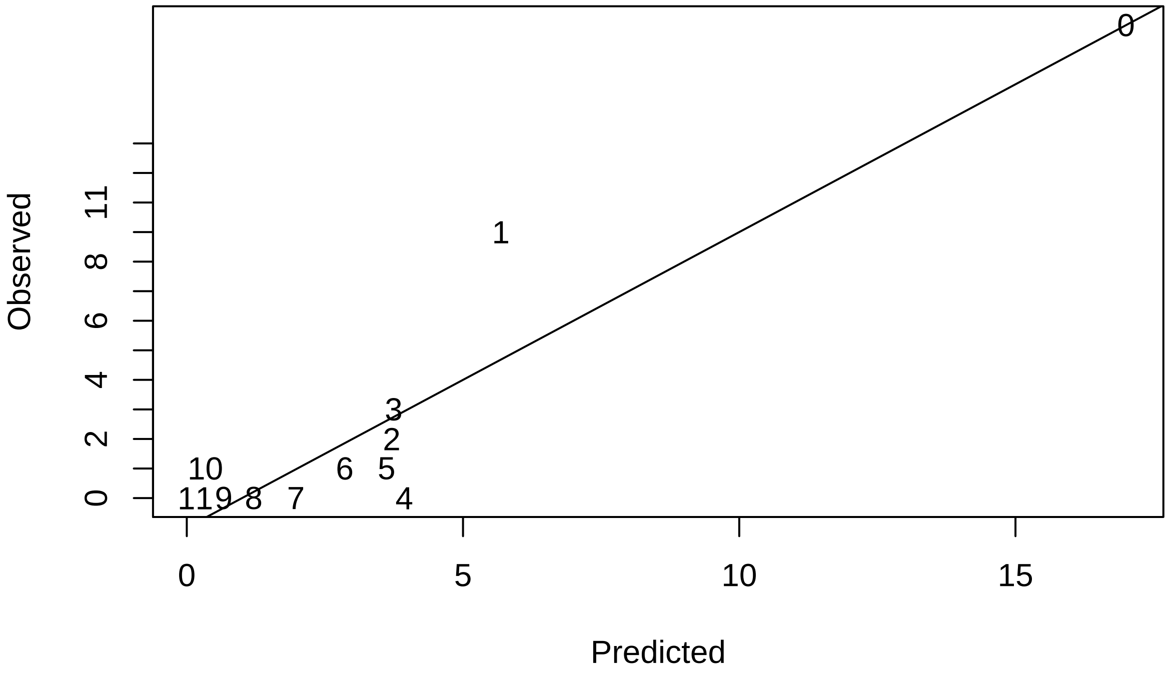
```r
# So, none of these models appear to fit particularly well, really low
# p-values from the goodness of fit tests. But, of the three models, we see
# that the one with only Moist (m3) seems to be the best, according to these
# tests. Additionally, we see from the summary command that m3 has the
# lowest AIC of these models.  Also, from the fourth test above we can
# reject the null hypothesis that the reduced model, with our factor
# *moistcat*, fits better than the model with the full *Moist* regressor.
# Also, the addition of the quadratic term for Moist^2 does not seem to
# significantly improve the model, as seen in the fifth test above versus
# the reduced model with just *Moist*.  So, of all the basic poisson models,
# we see that the model with *Moist* alone fits the best.  Will now try some
# tests to see why the deviance is still so high in m3.

pcount <- colSums(predprob(m3)[, 1:13])
ocount <- table(weed$Weeds)[1:13]
plot(pcount, ocount, type = "n", xlab = "Predicted", ylab = "Observed")
text(pcount, ocount, 0:12)
abline(0, 1)
```



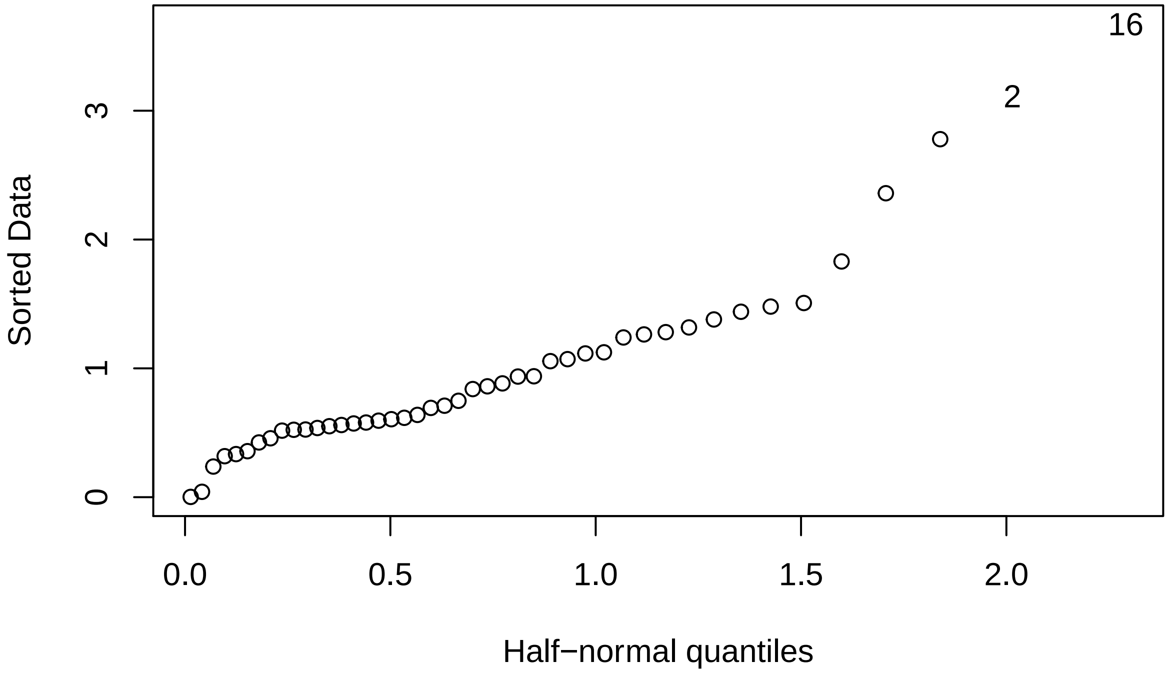
```
# The observed number of zeros (17) is much higher than the expected (~13),
# so it seems like going with a zero inflated poisson model would be a good
# choice here. But, the number of 1's seems to be higher than expected as
# well. Maybe the probability of observing 0 or 1 type of weeds is different
# than the other amounts and so 1 should be the hurdle?


mz1 <- hurdle(formula = Weeds ~ Moist, data = weed)
mz2 <- hurdle(formula = Weeds ~ Moist + Days, data = weed)
# mz3 <- zeroinfl(formula = Weeds ~ Moist, data = weed)
mz4 <- hurdle(formula = Weeds ~ Moist, data = weed, level = 1)

lrt <- 2 * (mz2$loglik - mz1$loglik)

1 - pchisq(lrt, 2)
```

```
## [1] 0.4866434
```

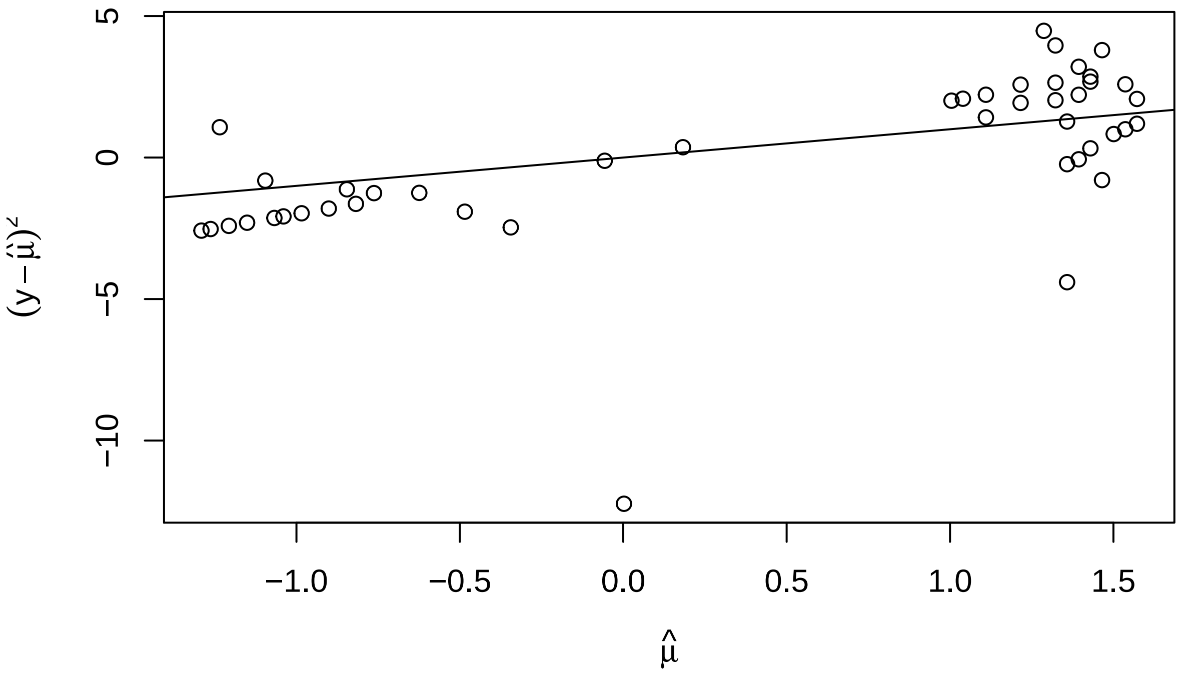
```
# So, can't reject null hypothesis that the reduced model, mz1, adaquately
# explains the response.

pcount <- colSums(predprob(mz1)[, 1:13])
plot(pcount, ocount, type = "n", xlab = "Predicted", ylab = "Observed")
text(pcount, ocount, 0:12)
abline(0, 1)
```



```
# So fitting the hurdle model makes this look much better, but the 1 count
# still has more observations than predicted.

halfnorm(residuals(mz1))
```



```
# No outliers

plot(log(fitted(mz1)), log((weed$Weeds - fitted(mz1))^2), xlab = expression(hat(mu)),
    ylab = expression((y - hat(mu))^2))
abline(0, 1)
```



```
# Mean seems pretty close to the variance here, not perfect, but it's okay.
```

2.

```
summary(mz1)
```

```
##
## Call:
## hurdle(formula = Weeds ~ Moist, data = weed)
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.5072 -0.8604 -0.5369  0.5250  3.6711
##
## Count model coefficients (truncated poisson with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.181626   0.807622  -1.463 0.143442
## Moist        0.031395   0.009248   3.395 0.000687 ***
## Zero hurdle model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.169869   0.626352  -1.868  0.06180 .
## Moist        0.030743   0.009953   3.089  0.00201 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 7
## Log-likelihood: -90.13 on 4 Df
```

```
exp(coef(mz1))
```

```
## count_(Intercept)       count_Moist  zero_(Intercept)        zero_Moist
##         0.3067794         1.0318932         0.3104076         1.0312202
```

3.

```
newlawn <- data.frame(Moist = 83, Days = 365)

predict(mz1, newdata = newlawn, type = "response")
```

```
##        1
## 3.373474
```

```
predict(mz1, newdata = newlawn, type = "prob")
```

```
##           0         1         2        3         4         5          6
## 1 0.2007238 0.0529464 0.1099807 0.152302 0.1581816 0.1314305 0.09100286
##            7          8          9          10          11           12
## 1 0.05400913 0.02804707 0.01294658 0.005378552 0.002031341 0.0007032535
##             13
## 1 0.0002247392
```

```
# Also, can look at the probability coming from the 'zero'' part of the
# model 1 - predict(mz1, newdata = newlawn, type = 'zero') When the hurdle
# model is used, type = zero gives the probability of observing a non-zero
# count, based on the zero hurdle component - so this number is the
# probability of seeing a non-zero number of types of weeds. So 1- the
# predict command here is the probability of seeing a zero-number of types
# from the zero component of the hurdle model alone.

weedtable <- predict(mz1, newdata = newlawn, type = "prob")

sum(weedtable[5:14])
```

```
## [1] 0.4839556
```

4.

```
X = model.matrix(Weeds ~ Days * Moist, weed)
y = as.numeric(weed$Weeds)

beta_t = rep(0, 4)
mu = exp(X %*% beta_t)
beta_t1 = beta_t + solve(t(X) %*% diag(c(mu)) %*% X) %*% t(X) %*% (y - mu)

i.count = 1
while (sum((beta_t1 - beta_t)^2) > 1e-06) {
    beta_t = beta_t1
    mu = exp(X %*% beta_t)
    beta_t1 = beta_t + solve(t(X) %*% diag(c(mu)) %*% X) %*% t(X) %*% (y - mu)
    i.count = i.count + 1
    print(c(i.count, beta_t1))
}
```

```
## [1]  2.0000000000 -1.1477665706 -0.0253648885  0.0396922061  0.0002862089
## [1]  3.0000000000 -1.2400295662 -0.0163585743  0.0340464642  0.0001850857
## [1]  4.000000e+00 -1.345957e+00 -7.149163e-03  3.234398e-02  8.028466e-05
## [1]  5.000000e+00 -1.404853e+00 -4.078056e-03  3.259026e-02  4.451837e-05
## [1]  6.000000e+00 -1.410236e+00 -3.878250e-03  3.264639e-02  4.212704e-05
## [1]  7.000000e+00 -1.410263e+00 -3.877390e-03  3.264673e-02  4.211656e-05
```

```
beta_t1
```

```
##                      [,1]
## (Intercept) -1.410263e+00
## Days        -3.877390e-03
## Moist        3.264673e-02
## Days:Moist   4.211656e-05
```

```
m.rep <- glm(Weeds ~ Days * Moist, family = "poisson", data = weed)
summary(m.rep)$coef[, 1]
```

```
##   (Intercept)          Days         Moist    Days:Moist
## -1.410263e+00 -3.877390e-03  3.264673e-02  4.211656e-05
```