Count Data Regression with Excess Zeros: A Flexible Framework Using the GLM Toolbox

Christian Kleiber, Achim Zeileis

http://eeecon.uibk.ac.at/~zeileis/

Count data regression with excess zeros

In practice: The basic Poisson regression model is often not flexible enough to capture count data observed in applications.

- *Overdispersion:* Variance is higher than the mean. Often addressed by adopting a negative binomial (NB) model.
- *Excess zeros:* (Far) more zeros observed than expected from Poisson (or NB) model.

Here: Focus on excess zeros. Poisson will be employed for simplicity but most ideas work analogously for NB.

Strategies:

- *Zero-inflation model:* Finite mixture model of a Poisson regression and a point mass at zero. Zeros can come from either component.
- *Hurdle model:* Two part model with a binary hurdle part and a zero-truncated count part. Only a single source of zeros and hence simpler to fit and interpret.

Hurdle count data models

Idea: Account for excess (or lack of) zeros by *two-part model*.

- Is y equal to zero or positive? “Is the hurdle crossed?”
- If $y > 0$, how large is y?

Formally:

- *Zero hurdle:* $f_{\text{zero}}(y; z, \gamma)$. Binary part given by count distribution right-censored at $y = 1$ (or simply Bernoulli variable).
- *Count part:* $f_{\text{count}}(y; x, \beta)$. Count part given by count distribution left-truncated at $y = 1$.

Combined: Probability density function for hurdle model,

$$
\begin{aligned}
f_{\text{hurdle}}&(y; x, z, \beta, \gamma) \\
&= \begin{cases} f_{\text{zero}}(0; z, \gamma), & y = 0, \\ \{1 - f_{\text{zero}}(0; z, \gamma)\} \cdot f_{\text{count}}(y; x, \beta) / \{1 - f_{\text{count}}(0; x, \beta)\}, & y > 0. \end{cases}
\end{aligned}
$$

Hurdle models as two GLMs

Estimation: Facilitated by properties that are not as well known as they deserve to be.

- Both parts of the hurdle model can be fitted separately.
- Each of the two parts is a GLM (or a straightforward extension thereof in case of NB).

Illustration: Poisson hurdle model.

Zero hurdle part: From Poisson with $\log(\lambda) = z^\top \gamma$.

$$
\begin{aligned}
\pi &= 1 - f_{\text{zero}}(0; z, \gamma) \\
&= 1 - \exp(-\lambda) \\
&= 1 - \exp(-\exp(z^\top \gamma)) \\
\log(-\log(1 - \pi)) &= z^\top \gamma
\end{aligned}
$$

Thus: Binary GLM with complementary log-log link.

Hurdle models as two GLMs

Zero-truncated count part: From Poisson with $\log(\lambda) = x^\top \beta$.

$$\begin{aligned}
\frac{f_{\text{count}}(y; x, \beta)}{1 - f_{\text{count}}(0; x, \beta)} &= \frac{\lambda^y \exp(-\lambda)}{y!\{1 - \exp(-\lambda)\}} \\
&= \exp\left\{y \log \lambda - \lambda - \log(1 - \exp(-\lambda)) - \log y!\right\}
\end{aligned}$$

Thus: Exponential family corresponding to a GLM.

However: The inverse link function is given by

$$\begin{aligned}
\text{E}(y | y > 0) &= \frac{\lambda}{1 - \exp(-\lambda)} \\
&= \frac{\exp(x^\top \beta)}{1 - \exp(-\exp(x^\top \beta))}
\end{aligned}$$

…and the link function has no closed form.

Hurdle models as two GLMs

Advantages:

- Theoretical properties of GLMs are inherited.
- Implementation can be carried out by standard GLM software with suitable families.
- Methods for GLMs and their extensions can be leveraged for hurdle models.

Implementation:

- A `"family"` object `ztpoisson()` in package *countreg*.
- Link function is computed numerically.

Illustration: Australian doctor visits

Description: Cross-section data with 5,190 observations originating from the 1977–1978 Australian Health Survey.

Source: Cameron & Trivedi (1986, *Journal of Applied Econometrics*).

Variables:

`visits`	Number of doctor visits in past 2 weeks.
`gender`	Factor indicating gender.
`health`	General health questionnaire score using Goldberg's method (GHQ-12).
`income`	Annual income (in 10,000 dollars).
`age`	Age (in 100 years).
...	

Illustration: Australian doctor visits

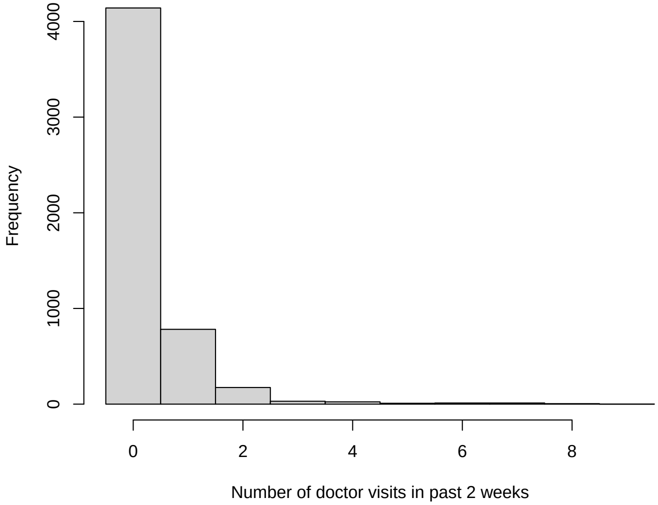


Illustration: Poisson hurdle model

Estimation: Dedicated `hurdle()` fitting function.

```
R> library("countreg")
R> dv0 <- hurdle(visits ~ gender + health + income + poly(age, 2),
+    data = DoctorVisits, dist = "poisson", zero.dist = "poisson")
R> summary(dv0)
```

```
Call:
hurdle(formula = visits ~ gender + health + income + poly(age, 2), data
    dist = "poisson", zero.dist = "poisson")

Pearson residuals:
    Min      1Q  Median      3Q     Max
-1.2743 -0.4528 -0.3638 -0.3148 14.7294

Count model coefficients (truncated poisson with log link):
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.03126    0.11716  -0.267  0.78960
genderfemale  -0.13488    0.08913  -1.513  0.13022
health         0.07588    0.01208   6.282 3.33e-10 ***
income        -0.45814    0.14332  -3.197  0.00139 **
poly(age, 2)1  1.98614    3.24091   0.613  0.53999
poly(age, 2)2 -8.16804    2.95390  -2.765  0.00569 **
```

GLM

```
Zero hurdle model coefficients (censored poisson with log link):
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -1.88034    0.08577 -21.924  < 2e-16 ***
genderfemale   0.26538    0.06884   3.855 0.000116 ***
health         0.14555    0.01104  13.179  < 2e-16 ***
income        -0.02763    0.10057  -0.275 0.783521
poly(age, 2)1 21.05519    2.34957   8.961  < 2e-16 ***
poly(age, 2)2  3.29889    2.33186   1.415 0.157155
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of iterations in BFGS optimization: 37
Log-likelihood: -3537 on 12 Df
```

Illustration: Two GLMs

Estimation: Standard `glm()` function with new `ztpoisson` family.

```
R> dv0z <- glm(factor(visits > 0) ~ gender + health + income +
+    poly(age, 2), data = DoctorVisits,
+    family = binomial(link = "cloglog"))
R> dv0c <- glm(visits ~ gender + health + income + poly(age, 2),
+    data = DoctorVisits, family = ztpoisson, subset = visits > 0)
```

Results: Essentially identical parameter estimates.

	hurdle-zero	glm-cloglog	hurdle-count	glm-ztpoisson
(Intercept)	-1.8803	-1.8804	-0.0313	-0.0313
genderfemale	0.2654	0.2654	-0.1349	-0.1349
health	0.1456	0.1456	0.0759	0.0759
income	-0.0276	-0.0276	-0.4581	-0.4582
poly(age, 2)1	21.0552	21.0554	1.9861	1.9859
poly(age, 2)2	3.2989	3.3001	-8.1680	-8.1689

Advantage: Can leverage tools such as the *effects* package.

Illustration: Effect displays

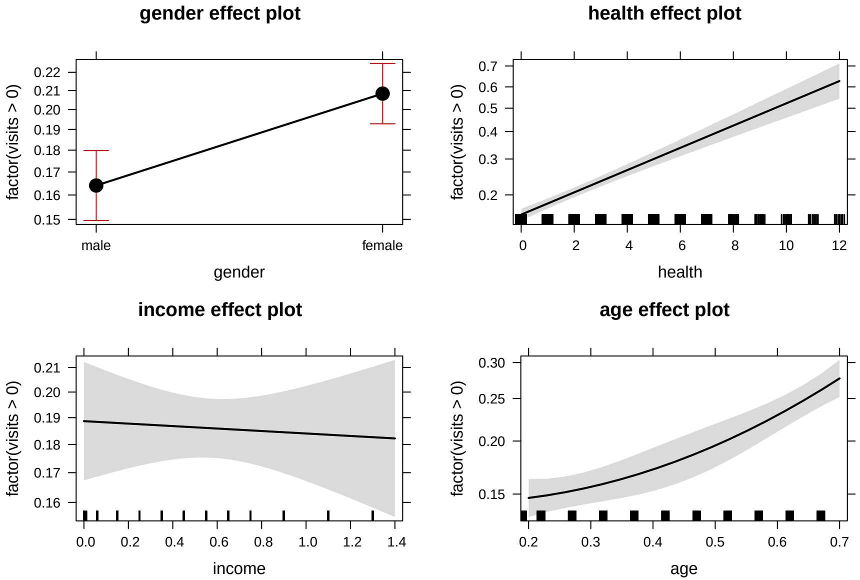


Illustration: Effect displays

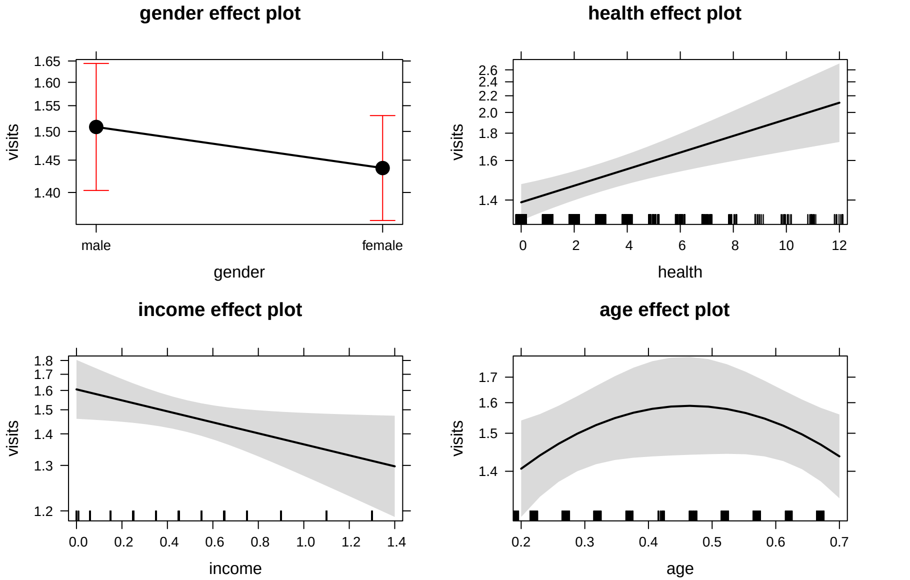


Illustration: GAMs

Extension: Generalized additive models.

- Package *mgcv* can use the `ztpoisson` family to estimate GAM versions of both parts.
- Needs some further derivatives of link and variance function in the `"family"` object.
- Computed either analytically or numerically.

Application: Use simple splines for numeric covariates.

```
R> library("mgcv")
R> dv1z <- gam(factor(visits > 0) ~ gender +
+    s(health, k = 5) + s(income, k = 5) + s(age, k = 5),
+    data = DoctorVisits, family = binomial(link = "cloglog"))
R> dv1c <- gam(visits ~ gender +
+    s(health, k = 5) + s(income, k = 5) + s(age, k = 5),
+    data = DoctorVisits, family = ztpoisson, subset = visits > 0)
```

Illustration: GAMs

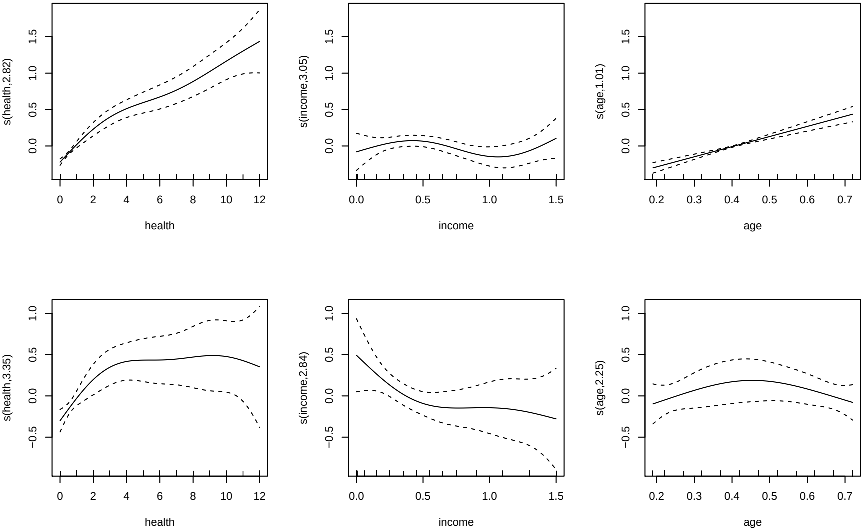


Illustration: Boosting

Extension: Boosting for GLMs or GAMs.

- Does not require the GLM framework, just an additive predictor and the score function of the model.
- Implemented in `MBztpoisson()` and `MBbinomial()` families for package *mboost*.
- Can be used for shrinkage and variable selection but requires selecting a tuning parameter `mstop`.

Application: Boosted GLMs.

```
R> library("mboost")
R> dv3c <- glmboost(visits ~ gender + health + income +
+    poly(age, 2), data = subset(DoctorVisits, visits > 0),
+    family = MBztpoisson(), control = boost_control(mstop = 1000))
R> set.seed(0)
R> dv3c_cv <- cvrisk(dv3c)
R> mstop(dv3c) <- mstop(dv3c_cv)
```

Illustration: Boosting

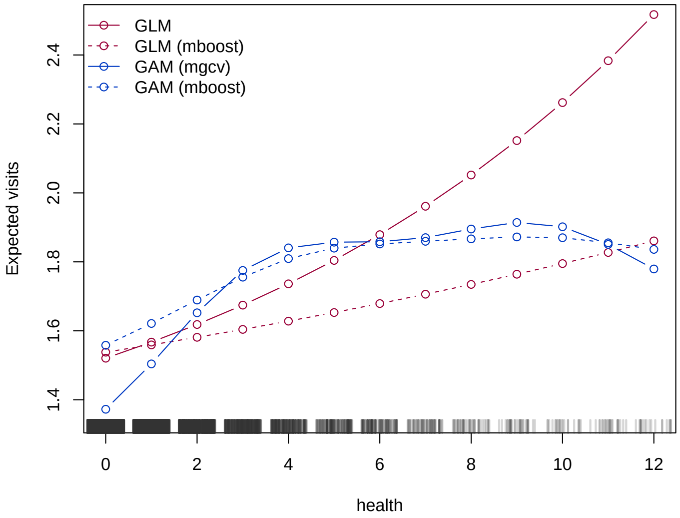
Results: 33 iterations, most coefficients not selected at all.

```
                    glm glmboost
(Intercept)   -0.0313  -0.0691
genderfemale  -0.1349   0.0000
health         0.0759   0.0340
income        -0.4582   0.0000
poly(age, 2)1  1.9859   0.0000
poly(age, 2)2 -8.1689   0.0000
```

Analogously: `gamboost` with boosted B-splines, 17 iterations.

Comparison: Health effect displays for males with average income/age.

Illustration: Comparison



Summary

- Hurdle models are easy to fit and interpret.
- They can be regarded as combining two GLMs.
- Paves the way for GLM-based extensions.
- Numerical computations might have to use approximations of the link and variance function.

References

Zeileis A, Kleiber C (2015). *countreg: Count Data Regression.* R package version 0.1-5/r104. URL https://R-Forge.R-project.org/projects/countreg/

Zeileis A, Kleiber C, Jackman S (2008). “Regression Models for Count Data in R.” *Journal of Statistical Software*, **27**(8), 1–25. doi:10.18637/jss.v027.i08