Chapter 9
Generalized linear models

{ch:glm}

Generalized linear models extend the familiar linear models of regression and ANOVA to include counted data, frequencies, and other data for which the assumptions of independent, normal errors are not reasonable. We rely on the analogies between ordinary and generalized linear models (GLMs) to develop visualization methods to display the fitted relations and check model assumptions.

> In one word, to draw the rule from experience, one must generalize; this is a necessity that imposes itself on the most circumspect observer.
>
> Henri Poincaré, *The Value of Science: Essential Writings of Henri Poincare*

In the modern history of statistics, most developments occur incrementally, with small additions to existing models and theory that extend their range and applicability to new problems and data. Occasionally, there is a major synthesis that unites a wide class of existing methods in a general framework and provides opportunities for far greater growth.

A prime example is the theory of generalized linear models, introduced originally by Nelder and Wedderburn (1972), that extended the familiar (classical) linear models for regression and ANOVA to include related models, such as logistic regression and logit models (described in Chapter 7) and loglinear models (described in Chapter 8) and other variations as "families" within a single general system.

This approach has proved attractive because it: (a) integrates many familiar statistical models in a general theory where they are just special cases; (b) provides the basis for extending these and developing new models within the same framework; (c) simplifies the implementation of these models in software, since the same algorithm can be used for estimation, inference and assessing model adequacy for all generalized linear models.

Section 9.1 gives a brief sketch of the GLM framework. The focus of this book is on visualization methods for categorical data, and the two important topics concern models and methods for binomial response data and for count data. The first of these, was described extensively in Chapter 7, with extensions to multinomial data (Section 7.5) and there is little to add here, except for changes in notation.

GLM models for count data, however, provide the opportunity to extend the scope of of these methods beyond what was covered in Chapter 8, and this topic is introduced in Section 9.2. The

GLM framework also provides the opportunity to deal with common problems of overdispersion (Section 9.3) and an overabundance of zero counts (Section 9.3), giving some new models and visualization methods that help to understand such data in greater detail. Section 9.5 illustrates other graphical methods for diagnostic model checking, some of which were introduced in earlier chapters. **TODO**: Complete this chapter overview.

9.1 Components of Generalized Linear Models

{glm:components}

The motivation for the ***generalized linear model*** (GLM) and its structure are most easily seen by considering the classical linear model,

$$y_i = \boldsymbol{x}_i^\mathsf{T} \boldsymbol{\beta} + \epsilon_i$$

where y_i is the response variable for case $i, i = 1, \dots n$, $\boldsymbol{x}_i$ is the vector of explanatory variables or regressors, $\boldsymbol{\beta}$ is the vector of model parameters, and the ϵ_i are random errors. In the classical linear model, the ϵ_i are assumed to (a) have constant variance, σ_ϵ^2, (b) follow a normal (Gaussian) distribution (conditional on $\boldsymbol{x}_i$), (c) be independent across observations.

Thus, Nelder and Wedderburn (1972) generalized this gaussian linear model to consist of the following three components, by relaxing assumptions (a) and (b) above:[1]

random component The conditional distribution of the $y_i \,|\, \boldsymbol{x}_i$, with mean $\mathcal{E}(y_i) = \mu_i$. Under classical assumptions, this is independent, normal with constant variance σ^2, i.e., $y_i \overset{\text{iid}}{\sim} N(\mu_i, \sigma^2)$. In the GLM, the probability distribution of the y_i can be any member of the ***exponential family***, including the normal, Poisson, binomial, gamma and others. Subsequent work has extended this framework to include multinomial distributions and some non-exponential families such as the negative binomial distribution.

systematic component The idea that the predicted value of y_i itself is a linear combination of the regressors is replaced by that of a ***linear predictor***, η, that captures this aspect of linear models,

$$\eta_i = \boldsymbol{x}_i^\mathsf{T} \boldsymbol{\beta}$$

link function The connection between the mean of the the response, μ_i, and the linear predictor, η_i, is specified by the ***link function***, $g(\bullet)$, giving

$$g(\mu_i) = \eta_i = \boldsymbol{x}_i^\mathsf{T} \boldsymbol{\beta}$$

The link function $g(\bullet)$ must be both *smooth* and *monotonic*, meaning that it is one-to-one, so an inverse transformation, $g^{-1}(\bullet)$ exists,

$$\mu_i = g^{-1}(\eta_i) = g^{-1}(\boldsymbol{x}_i^\mathsf{T} \boldsymbol{\beta})$$

which allows us to obtain and plot the predicted values on their original scale. The link function captures the familiar idea that linear models are often estimated with a transformation of the response, such as $\log(y_i)$ for a frequency variable or $\text{logit}(y_i)$ for a binomial variable. The inverse function $g^{-1}(\bullet)$ is also called the ***mean function***.

link-funcs}

Table 9.1: Common link functions and their inverses used in generalized linear models

Link name	Function: $\eta_i = g(\mu_i)$	Inverse: $\mu_i = g^{-1}(\eta_i)$
identity	μ_i	η_i
square-root	$\sqrt{\mu_i}$	η_i^2
log	$\log_e(\mu_i)$	$\exp(\eta_i)$
inverse	μ_i^{-1}	η_i^{-1}
inverse-square	μ_i^{-2}	$\eta_i^{-1/2}$
logit	$\log_e \frac{\mu_i}{1-\mu_i}$	$\frac{1}{1+\exp(-\eta_i)}$
probit	$\Phi^{-1}(\mu_i)$	$\Phi^{-1}(\eta_i)$
log-log	$-\log_e[-\log_e(\mu_i)]$	$\exp[-\exp(-\eta_i)]$
comp. log-log	$\log_e[-\log_e(1-\mu_i)]$	$1-\exp[-\exp(\eta_i)]$

Some commonly used link functions are shown in Table 9.1. Some of these link functions have restrictions on the range of y_i to which they can be applied. For example, the square-root and log links apply only to non-negative and positive values respectively. The last four link functions in this table are for binomial data, where y_i represents the observed proportion of successes in n_i independent trials, and thus the mean μ_i represents the probability of success (symbolized by π_i in Chapter 7). Binary data are the special case where $n_i = 1$.

9.1.1 Variance functions

The GLM has the additional property that, for distributions in the exponential family, the conditional variance of $y_i \,|\, \eta_i$ is a known function, $\mathcal{V}(\mu_i)$ of the mean and possibly one other parameter called the ***scale parameter*** or ***dispersion parameter***, ϕ. Some commonly used distributions in the exponential family and their variance functions are shown in Table 9.2.

Table 9.2: Common distributions in the exponential family used with generalized linear models and their canonical link and variance functions

{tab:exp-families}

Family	Notation	Canonical link	Range of y	Variance function, $\mathcal{V}(\mu \,|\, \eta)$
Gaussian	$N(\mu, \sigma^2)$	identity: μ	$(-\infty, +\infty)$	ϕ
Poisson	$\text{Pois}(\mu)$	$\log_e(\mu)$	$0, 1, \ldots, \infty$	μ
Negative-Binomial	$\text{NBin}(\mu, \theta)$	$\log_e(\mu)$	$0, 1, \ldots, \infty$	$\mu + \mu^2/\theta$
Binomial	$\text{Bin}(n, \mu)/n$	$\text{logit}(\mu)$	$\{0, 1, \ldots, n\}/n$	$\mu(1-\mu)/n$
Gamma	$G(\mu, \nu)$	μ^{-1}	$(0, +\infty)$	$\phi\mu^2$
Inverse-Gaussian	$IG(\mu, \nu)$	μ^2	$(0, +\infty)$	$\phi\mu^3$

- In the classical Gaussian linear model, the conditional variance is constant, $\phi = \sigma_\epsilon^2$.

[1] The remaining assumption of independent observations is relaxed in ***generalized linear mixed models*** (GLMMs), in which random effects to account for non-independence are added to the linear predictor. This allows the modeling of correlated (responses of family members), clustered (residents in different communities) or hierarchical data (patients within hospitals within regions). See: McCulloch and Neuhaus (2005) ... **TODO**: other references?

- In the Poisson family, $\mathcal{V}(\mu_i) = \mu_i$ and the dispersion parameter is fixed at $\phi = 1$. In practice, it is common for count data to exhibit ***overdispersion***, meaning that $\mathcal{V}(\mu_i) > \mu_i$. One way to correct for this is to extend the GLM to allow the dispersion parameter to be estimated from the data, giving what is called the ***quasi-poisson*** family, with $\mathcal{V}(\mu_i) = \widehat{\phi}\mu_i$.

- Similarly, for binomial data, the variance function is $\mathcal{V}(\mu_i) = \mu_i(1 - \mu_i)/n_i$, with ϕ fixed at 1. Overdispersion often results from failures of the assumptions of the binomial model: supposedly independent observations may be correlated or clustered and the probability of success may not be constant, or vary with unmeasured or unmodeled variables.

- The gamma and inverse-Gaussian families are distributions useful for modeling a continuous and positive response variable with no upper bound (e.g., reaction time). They both have the property that conditional variance increases with the mean, and for the inverse-Gaussian, variance increases at a faster rate. Their dispersion parameters ϕ are simple functions of their intrinsic "shape" parameters, indicated as ν in the table.

The important points from this discussion are that the GLM together with the exponential family of distributions:

- provide for simple, linear relations between the response and the predictors via the link function and the linear predictor.

- allows a very flexible relationship between the mean and conditional variance to be specified in terms of a set of known families.

- incorporates a dispersion parameter ϕ that in some cases can be estimated or tested for departure from that entailed in a given family.

- has allowed further extensions of this framework outside the exponential family, ranging from simple adjustments for statistical inference ("quasi" families, adjusted "sandwich" covariances) to separate modeling of the variance relation to the predictors.

Further details of generalized linear models are beyond the scope of this book, but the interested reader should consult Fox (2008, §15.3) and Agresti (2013, Ch. 4) for a comprehensive treatment.

9.1.2 Hypothesis tests for coefficients

GLMs are fit using maximum likelihood estimation, and implemented in software using an iterative algorithm known as *iteratively weighted least squares* that generalizes the least squares method for classical linear models. This provides estimates $\widehat{\boldsymbol{\beta}}$ of the model coefficients for the predictors in $\boldsymbol{x}$, as well as an estimated asymptotic (large sample) variance matrix of $\widehat{\boldsymbol{\beta}}$, given by

{eq:varbeta}

$$\mathcal{V}(\widehat{\boldsymbol{\beta}}) = \phi(\boldsymbol{X}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{X}) \tag{9.1}$$

where $\boldsymbol{W}$ is a diagonal matrix of weights computed in the final iteration. In the standard Poisson GLM, the weight matrix is $\boldsymbol{W} = \mathrm{diag}\,(\widehat{\boldsymbol{\mu}})$ and $\phi = 1$ is assumed.

Asymptotic standard errors, $se(\widehat{\beta}_j)$, for the coefficients are then the square roots of the diagonal elements of $\mathcal{V}(\widehat{\beta})$, and tests of hypotheses regarding an individual coefficient, e.g.,

$H_0 : \beta_j = 0$, can be carried out using the Wald test statistic, $z_j = \widehat{\beta}_j / se(\widehat{\beta}_j)$. When the null hypothesis is true, z_j has a standard normal $N(0, 1)$ distribution, providing p-values for significance tests.[2]

9.1.3 Goodness-of-fit tests

{sec:glm-goodfit}

The basic ideas for testing goodness-of-fit were discussed in Section 8.3.2 in connection with loglinear models for contingency tables. As before, these assess the overall performance of a model in reproducing the data. The commonly used measures include the Pearson chi-square and likelihood-ratio deviance statistics, which can be seen as weighted sums of residuals. We re-state these test statistics here in the wider context of the GLM.

Let $y_i, i = 1, 2, \dots, n$ be the response and $\widehat{\mu}_i = g^{-1}(\boldsymbol{x}_i^\mathsf{T} \widehat{\boldsymbol{\beta}})$ the fitted mean using the estimated coefficients, having estimated variance $\widehat{\omega}_i = \mathcal{V}(\widehat{\mu}_i \,|\, \eta_i)$ as in Table 9.2. Then the normalized squared residual for observation i is $(y_i - \widehat{\mu}_i)^2 / \widehat{\omega}_i$, and the Pearson statistic is

$$X_P^2 = \sum_{i=1}^{n} \frac{(y_i - \widehat{\mu}_i)^2}{\widehat{\omega}_i} \ . \qquad (9.2)$$

{eq:pearson}

In the GLM for count data, the main focus of this chapter, the Poisson family sets $\omega = \mu$ with the dispersion parameter fixed at $\phi = 1$.

The ***residual deviance*** statistic, as in logistic regression and loglinear models is defined as twice the difference between the maximum possible log-likelihood for the *saturated model* that fits perfectly and maximized log-likelihood for the fitted model. The deviance can be defined as

$$D(\boldsymbol{y}, \widehat{\boldsymbol{\mu}}) \equiv 2[\log \mathcal{L}(\boldsymbol{y}; \boldsymbol{y}) - \log \mathcal{L}(\boldsymbol{y}; \widehat{\boldsymbol{\mu}})]$$

For classical linear models under normality, the deviance is simply the residual sum of squares, $\sum_i^n (y_i - \widehat{\mu}_i)$. This has led to the deviance being taken in the GLM framework as a generalization of the sum of squares used in ANOVA, and hence, an anlogous ***analysis of deviance*** to carry out tests for individual terms in GLMs, or to compare nested models.

In R, `anova(mod)` for the "glm" object `mod` gives *sequential* (“Type I”) tests of successive terms in a model, while `Anova()` in the car package gives the more generally useful “Type II” (and “Type III”) *partial* tests, that assess the additional contribution of each term above all others, taking marginality into account.

For Poisson models with a log link giving $\boldsymbol{\mu} = \exp(\boldsymbol{x}^\mathsf{T} \boldsymbol{\beta})$, the deviance takes the form[3]

$$D(\boldsymbol{y}, \widehat{\boldsymbol{\mu}}) = 2 \sum_{i=1}^{n} \left[y_i \log_e \left(\frac{y_i}{\widehat{\mu}_i} \right) - (y_i - \widehat{\mu}_i) \right] \qquad (9.3)$$

{eq:pois-deviance}

For a GLM with p parameters, both the Pearson and residual deviance statistics follow approximate χ^2_{n-p} distributions with $n - p$ degrees of freedom.

[2] Wald tests are sometimes carried out using z^2, which has an equivalent χ^2_1 distribution with 1 degree of freedom.

[3] In the context of the loglinear models discussed in Section 8.3.2, this is also referred to as the likelihood-ratio G^2 statistic.

9.2 GLMs for count data

{sec:glm-co

The prototypical GLM for count data, where the response y_i takes on non-negative values $0, 1, 2, \ldots$, uses the Poisson family with the log link. We used this model extensively throughout all of Chapter 8. There the focus was on the special case of the loglinear model applied largely to contingency tables, where the loglinear model could be seen as a fairly direct extension of ANOVA models for a quantitative response applied to the log of cell frequency.

The advantage there was that models for two-way, three-way and by implication n-way tables could be discussed and illustrated using notation and graphs that separated the parameters and effects for one-way terms ("main effects"), two-way terms ("simple associations") and higher-way terms ("conditional associations").

The disadvantage is that these models as formulated there do not easily accommodate general quantitative predictors and were limited to the log link and the Poisson family. For example, the models discussed in Section 8.6 for ordinal variables allow one or more table factors to be assigned quantitative scores or have such scores estimated from the data, as in RC() models (Section 8.6.2). Yet, the contingency table approach for loglinear models breaks down if there are continuous predictors, and count data often exhibits features that make the equivalent Poisson regression model unsuitable or incomplete. We consider some extended models here.

{ex:phdpubs1}

Example 9.1: Publications of PhD candidates

In Example **??** we considered the distribution of the number of publications by PhD candidates in their last three years of study, but without taking any available predictors into account. For these data, a simple calculation shows why the Poisson distribution is unsuitable (for the marginal distribution), because the variance is 2.19 times the mean.

```
data("PhdPubs", package="vcdExtra")
with(PhdPubs, c(mean=mean(articles), var=var(articles),
                ratio=var(articles)/mean(articles)))

##  mean   var ratio
## 1.693 3.710 2.191
```

The earlier example showed rootograms (in Figure **??**) of the number of articles, but here it is useful to consider some more basic exploratory displays. A basic barplot of the frequency distribution of number of articles published is shown in the left panel of Figure 9.1. A quick look indicates that the distribution is highly skewed and there is a large number of counts of zero.

Another problem is that the frequencies of 0–2 articles account for over 75% of the total, so that the frequencies of the larger counts get lost in the display. The rootogram corrects for this by plotting frequency on the square-root scale. However, because we are contemplating a model with a log link, the same goal can be achieved by ploting log of frequency, as shown in the right panel of Figure 9.1. To accommodate the zero frequencies, the plot shows log(Frequency+1), avoiding errors from `log(0)`. It can be seen that log frequency decreases steadily up to 7 articles and then levels off approximately.

These plots are produced as shown below. The frequency distribution of `articles` can be tabulated by `table()`, but there is a subtle wrinkle here: By default, `table()` excludes the values of `articles` that do not occur in the data (zero frequencies). To include all values in the entire range, it is necessary to treat `articles` as a factor with levels `0:19`.

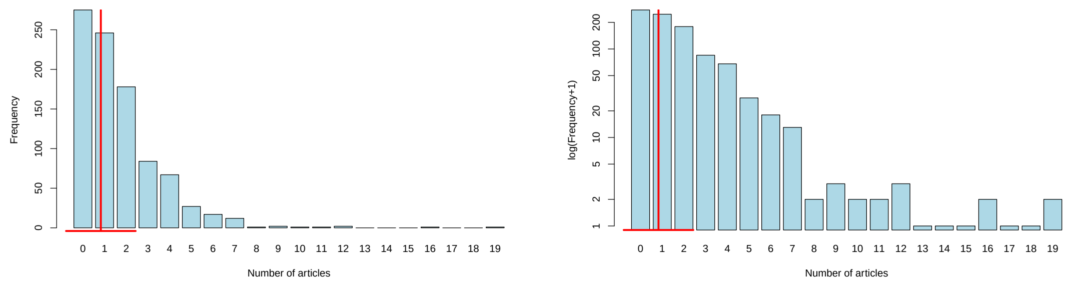

Figure 9.1: Barplots showing the frequency distribution of number of publications by PhD candidates. Left: raw scale; Right: a log scale makes the smaller counts more visible. The vertical red lines show the mean and horizontal lines show mean ±1 standard deviation.

bs-barplot}

```
art.fac <- factor(PhdPubs$articles, levels=0:19)  # include zero frequencies
art.tab <- table(art.fac)
art.tab

## art.fac
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
## 275 246 178  84  67  27  17  12   1   2   1   1   2   0   0   0   1   0   0   1
```

Then, the basic plot on the frequency scale is created using `barplot()`, and some annotations showing the mean and a one standard deviation interval can be added using standard plotting tools.

```
barplot(art.tab, xlab="Number of articles", ylab="Frequency",
        col="lightblue")
abline(v=mean(PhdPubs$articles), col="red", lwd=3)
ci <- mean(PhdPubs$articles)+c(-1,1) * sqrt(var(PhdPubs$articles))
lines(x=ci, y=c(-4, -4), col="red", lwd=3, xpd=TRUE)
```

Similarly, the plot on the log scale in the right panel of Figure 9.1 is produced with `barplot()`, but using `art.tab+1` to start frequency at one and `log="y"` to scale the vertical axis to log.

```
barplot(art.tab+1, ylab="log(Frequency+1)", xlab="Number of articles",
        col="lightblue", log="y")
```

Other useful exploratory plots for count data include boxplots of the response (on a log scale) and scatterplots against continuous predictors, where jittering the response is often necessary to avoid overplotting and a smooth nonparametric curve can show possible non-linearity. The `log="y"` option is again handy, and the formula method allows adding a start value to the response. Figure 9.2 illustrates these ideas, for the factor `married` and the covariate `mentor`.

```
boxplot(articles+1 ~ married, data=PhdPubs, log="y", varwidth=TRUE,
        ylab="log(articles+1)", xlab="married", cex.lab=1.25)
plot(jitter(articles+1) ~ mentor, data=PhdPubs, log="y",
      ylab="log(articles+1)", cex.lab=1.25)
lines(lowess(PhdPubs$mentor, PhdPubs$articles+1), col="red", lwd=2)
```

It can be seen that the distribution of articles for married and non-married are quite similar, except that for the married students there are quite a few observations with a large number of publications. The relationship between log(articles) and mentor publications seems largely linear

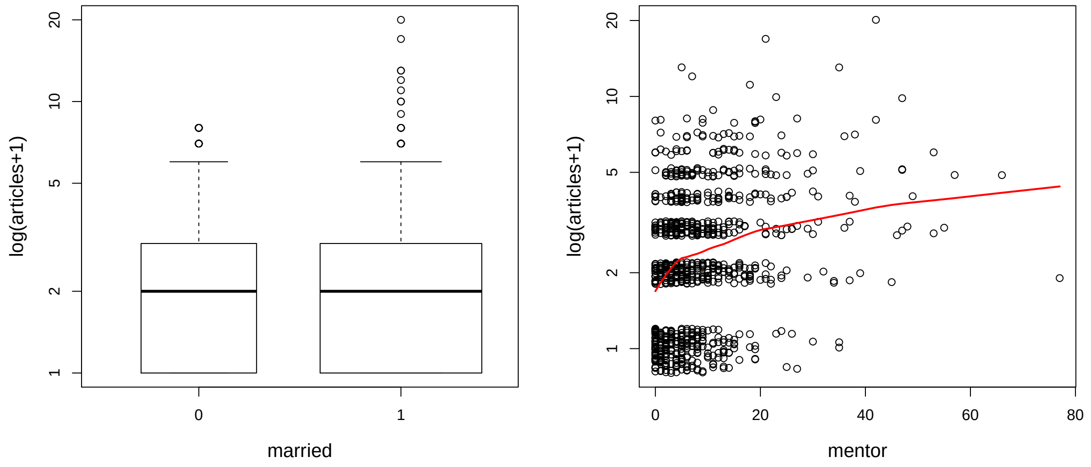

Figure 9.2: Exploratory plots for the number of articles in the PhdPubs data. Left: boxplots for married (1) vs. non-married (0); right: jittered scatterplot vs. mentor publicationswith a lowess smoothed curve.|fig:phdpubs-logplots

except possibly at the very low end. The large number of zero counts at the lower left corner stands out; this would not be seen without jittering.

To start analysis, we fit the Poisson model using all predictors— `female`, `married`, `kid5`, `phdprestige`, and `mentor`. As recorded in `PhdPubs`, `female` and `married` are both dummy (0/1) variables, and it it slightly more convenient for plotting purposes to make them factors.

```
PhdPubs <- within(PhdPubs, {
  female <- factor(female)
  married <- factor(married)
})
```

The model is fit as shown below and summarized using `summary()`, but with abbreviated output.

```
phd.pois <- glm(articles ~ ., data=PhdPubs, family=poisson)
summary(phd.pois)


...
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26562    0.09962    2.67   0.0077 **
## female1     -0.22442    0.05458   -4.11  3.9e-05 ***
## married1     0.15732    0.06125    2.57   0.0102 *
## kid5        -0.18491    0.04012   -4.61  4.0e-06 ***
## phdprestige  0.02538    0.02527    1.00   0.3153
## mentor       0.02523    0.00203   12.43  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1817.4  on 914  degrees of freedom
## Residual deviance: 1633.6  on 909  degrees of freedom
## AIC: 3313
...
```

Significance tests for the individual coefficients show that all are significant, except for phdprestige. We ignore this here, and continue to interpret and extend the full main effects model.[4]

The estimated coefficients $\boldsymbol{\beta}$ for the predictors are shown below. Recall that using the log link means, for example, that being married increases the log of the expected number of articles published by 0.157, holding all other predictors constant. Each additional child of age 5 or less decreases this by 0.185.

```
round(cbind(beta=coef(phd.pois),
            expbeta=exp(coef(phd.pois)),
            pct=100*(exp(coef(phd.pois))-1)),3)

##                beta expbeta     pct
## (Intercept)   0.266   1.304  30.425
## female1      -0.224   0.799 -20.102
## married1      0.157   1.170  17.037
## kid5         -0.185   0.831 -16.882
## phdprestige   0.025   1.026   2.570
## mentor        0.025   1.026   2.555
```

It is somewhat easier to interpret the exponentiated coefficients, $\exp(\boldsymbol{\beta})$ as multiplicative effects on the expected number of articles and convert these to percentage change, again holding other predictors constant. For example, expected publications by married candidates are 1.17 times that of non-married, a 17% increase, while each additional child multiplies articles by 0.831, a 16.88% decrease.

Alternatively, we recommend visual displays for model interpretation, and effect plots do well in most cases, as shown in Figure 9.3. For a Poisson GLM, an important feature is that the response is plotted on the log scale, so that effects in the model appear as linear functions, while the values of the response (numer of articles) are labeled on their original scale, facilitating interpretation. The confidence bands and error bars give 95% confidence intervals around the fitted efffects.

```
library(effects)
plot(allEffects(phd.pois), band.colors="blue", lwd=3,
     ylab="Number of articles", main="")
```

In Figure 9.3 we can see the decrease in published articles with number of young children, but also that the confidence band gets wider with increasing children. The predicted effect here of number of publications by the student's mentor is more dramatic, particularly for those whose mentor were truly prolific.

You should note that the panels for the predictors in Figure 9.3 are scaled individually for the range of the fitted main effects. This is often a sensible default and all predictors except mentor give a similar range here. To make all of these plots strictly comparable, provide a ylim argument, giving the range of the response on the log scale, as below (but not shown here).

[4] It is usually less harmful to include a non-significant predictor, (which in any case may be a variable useful to control, as phdprestige here), than to omit a potentially important predictor, or worse— to fail to account for an important interaction.

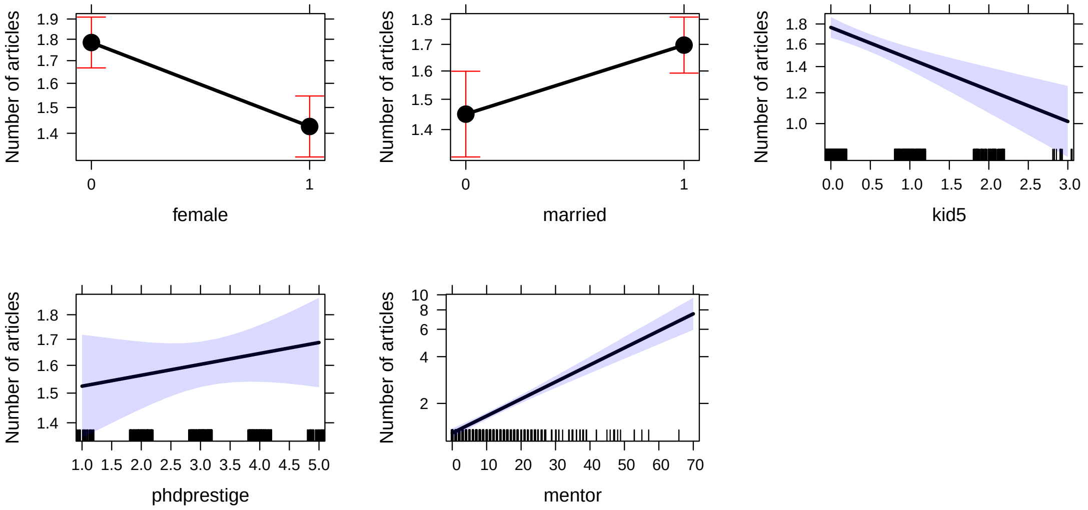

Figure 9.3: Effect plots for the predictors in the Poisson regression model for the PhdPubs data. Jittered values of the continuous predictors are shown at the bottom as rug-plots. fig:phdpubs1-effpois

```
plot(allEffects(phd.pois), band.colors="blue", ylim=c(0,log(10)))
```

All of the above is useful, but still leaves aside the question of how well the Poisson model fits the data. The output from `summary(phd.pois)` above showed that the Poisson model fits quite badly. The residual deviance of 1633.6 with 909 degrees of freedom is highly significant.

△

{ex:crabs1}

Example 9.2: Mating of horseshoe crabs

Brockmann (1996) studied the mating behavior of female horseshoe crabs in the Gulf of Mexico. In the mating season, crabs arrive on the beach in female/male pairs to lay and fertilize eggs. However, unattached males, called "satellites," also come to the beach, crowd around the nesting couples and compete with attached males for fertilizations, contributing to reproductive success. Some females are ignored by satellite males, and some attract more satellites than others, and the question is: what factors contribute to to the number of satellites for each female? Or, perhaps better, how do unattached males choose among available females? This is another example in which zero counts may require special treatment.

The data, given in `CrabSatellites` in the countreg package, give the response variable, `satellites` for 173 females. Possible predictors are the female's `color` and `spine` condition, given as ordered factors, as well as her `weight` and carapace (shell) `width`.

```
data("CrabSatellites", package = "countreg")
str(CrabSatellites)

## 'data.frame': 173 obs. of  5 variables:
##  $ color     : Ord.factor w/ 4 levels "lightmedium"<..: 2 3 3 4 2 1 4 2 2 2 ...
##  $ spine     : Ord.factor w/ 3 levels "bothgood"<"onebroken"<..: 3 3 3 2 3 2 3 3 1 3 ...
##  $ width     : num  28.3 26 25.6 21 29 25 26.2 24.9 25.7 27.5 ...
##  $ weight    : num  3.05 2.6 2.15 1.85 3 2.3 1.3 2.1 2 3.15 ...
##  $ satellites: int  8 4 0 0 1 3 0 0 8 6 ...
```

Agresti (2013, §4.3) analyses the number of satellites using count data GLMs, and in his Chapter 5, describes separate logistic regression models for the binary outcome of one or more satellites vs. none. Later in this chapter (Section 9.4) we consider hurdle and zero-inflated models for count data. These have the advantage of modeling the zero counts together with a model for the positive counts.

A useful overview plot of the data is shown using `gpairs()` in Figure 9.4. You can see that the distribution of *satellites* is quite positively skewed, with many zero counts. *width* and *weight* are highly corrrelated (0.89), and both relate to the size of the female. Their scatterplots in the first row show that larger females attract more satellites. The categorical ordered factors *spine* condition and *color* are strongly associated, with the lightest colored crabs having the best conditions.

```
library(vcd)
library(gpairs)
gpairs(CrabSatellites[,5:1],
       diag.pars = list(fontsize=16),
       mosaic.pars = list(gp=shading_Friendly, gp_args=list(interpolate=1:4)))
```

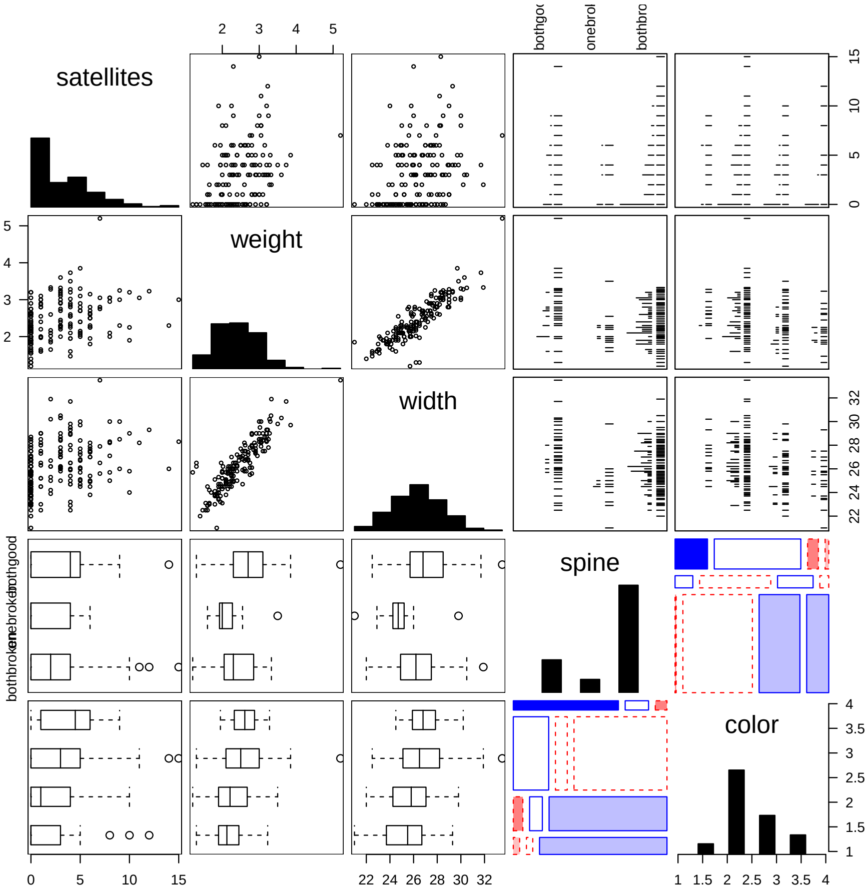

Figure 9.4: Generalized pairs plot for the CrabSatellites data. fig:crabs1-gpairs

Figure 9.5 shows the scatterplots of *satellites* against *width* and *weight* together with smoothed lowess curves.

```
plot(jitter(satellites) ~ width, data=CrabSatellites,
  ylab="Number of satellites (jittered)", xlab="Carapace width",
  cex.lab=1.25)
with(CrabSatellites, lines(lowess(width, satellites), col="red", lwd=2))
plot(jitter(satellites) ~ weight, data=CrabSatellites,
  ylab="Number of satellites (jittered)", xlab="Weight",
  cex.lab=1.25)
with(CrabSatellites, lines(lowess(weight, satellites), col="red", lwd=2))
```

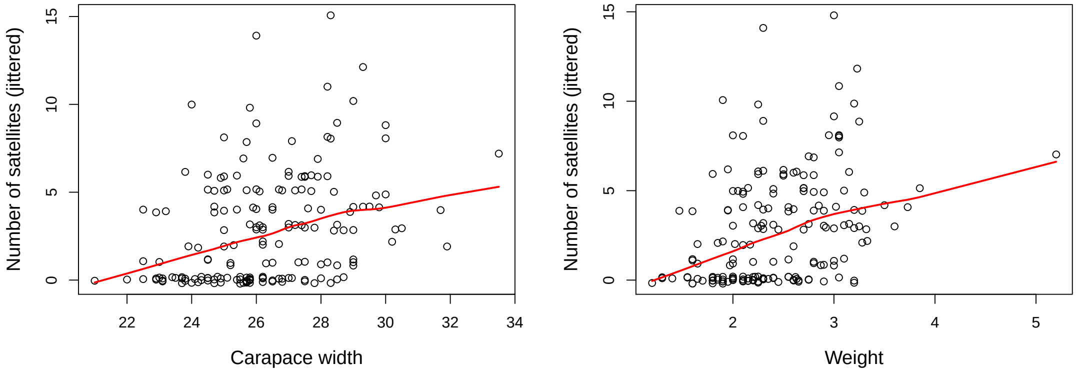

Figure 9.5: Scatterplots of number of satellites vs. width and weight, with lowess smooths. fig:crabs1-scats

Both variables show approximately linear relations to the mean number of satellites, so it would not be unreasonable to fit models using the identity link ($\mu \sim x$) rather than the log link ($\mu \sim \log(x)$) with the Poisson family GLM.

In these plots, we reduce the problem of overplotting of the discrete response by jittering, but an alternative technique is to transform a numeric count or continuous predictor to a factor (for visualization purposes only), thereby giving boxplots. A convenience function for this purpose, `cutfac()` is defined below. It acts like `cut()`, but gives nicer labels for the factor levels and by default chooses convenient breaks among the values based on deciles.

```
cutfac <- function(x, breaks = NULL, q=10) {
  if(is.null(breaks)) breaks <- unique(quantile(x, 0:q/q))
  x <- cut(x, breaks, include.lowest = TRUE, right = FALSE)
  levels(x) <- paste(breaks[-length(breaks)], ifelse(diff(breaks) > 1,
    c(paste("-", breaks[-c(1, length(breaks))] - 1, sep = ""), "+"), ""), sep = "")
  return(x)
}
```

Using this, the plots in Figure 9.5 can be re-drawn as boxplots, giving Figure 9.6.

```
plot(satellites ~ cutfac(width), data=CrabSatellites,
     ylab="Number of satellites", xlab="Carapace width (deciles)")
plot(satellites ~ cutfac(weight), data=CrabSatellites,
     ylab="Number of satellites", xlab="Weight (deciles)")
```

Wih this visual overview, we proceed to an initial Poisson GLM model, using all predictors. Note that *color* and *spine* are ordered factors, so `glm()` represents them as polynomial contrasts, as if they were coded numerically.

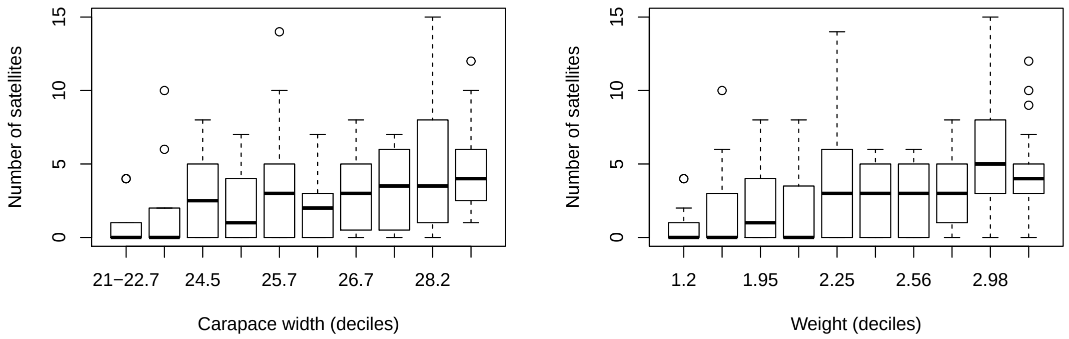

Figure 9.6: Boxplots of number of satellites vs. width and weight.fig:crabs1-boxplots

```
crabs.pois <- glm(satellites ~ ., data=CrabSatellites, family=poisson)
summary(crabs.pois)


...
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.7057     0.9344   -0.76   0.4501
## color.L       -0.4120     0.1567   -2.63   0.0085 **
## color.Q        0.1237     0.1231    1.00   0.3150
## color.C        0.0481     0.0914    0.53   0.5983
## spine.L        0.0618     0.0848    0.73   0.4660
## spine.Q        0.1585     0.1609    0.99   0.3244
## width          0.0165     0.0489    0.34   0.7358
## weight         0.4971     0.1663    2.99   0.0028 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 632.79  on 172  degrees of freedom
## Residual deviance: 549.56  on 165  degrees of freedom
## AIC: 920.9
...
```

The Wald tests for the coefficients show that only the linear effect of color and the effect of width are significant. Effect plots, in Figure 9.7, show the nature of these effects— lighter colored females attact more satellites, as do wider and heavier females.

```
plot(allEffects(crabs.pois), main="")
```

A simpler model can be constructed using `color` as a numeric variable, and either width or weight to represent female size. We choose weight here.[5]

[5] Agresti (2013, §4.3) and others who have analyzed this example uses carapace width as the main quantitative predictor, possibly because width might be more biologically salient to the single males than weight. This is a case where two highly correlated predictors are each strongly related to the outcome, yet partial tests (controlling for all others) may prefer one over the other.

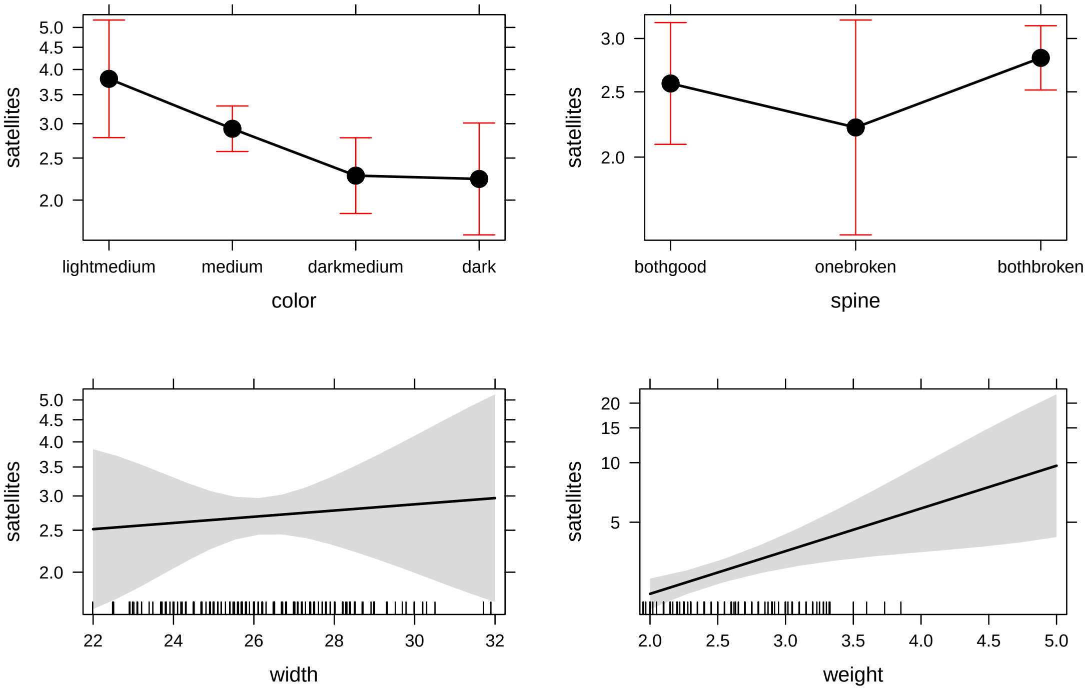

Figure 9.7: Effect plots for the predictors in the Poisson regression model for the CrabSatellites data. fig:crabs1-eff1

```
CrabSatellites1 <- transform(CrabSatellites,
  color = as.numeric(color))

crabs.pois1 <- glm(satellites ~ weight + color, data=CrabSatellites1,
                   family=poisson)
summary(crabs.pois1)


...
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.0888     0.2544    0.35    0.727
## weight        0.5458     0.0675    8.09    6e-16 ***
## color        -0.1728     0.0615   -2.81    0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 632.79  on 172  degrees of freedom
## Residual deviance: 552.77  on 170  degrees of freedom
## AIC: 914.1
...
```

From the statistical and graphical analysis so far, the answser to the question posed in this example is clear: unattached male horseshoe crabs prefer light-colored big, fat mamas!

Yet, neither of these models fit well, as can be seen from their residual deviances and likelihood-ratio tests.

```
vcdExtra::Summarise(crabs.pois, crabs.pois1)

## Likelihood summary table:
##              AIC BIC LR Chisq Df Pr(>Chisq)
## crabs.pois   921 946      550  8     <2e-16 ***
## crabs.pois1  914 924      553  3     <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Perhaps there is something else to be learned here.

△

9.3 Models for overdispersed count data

{sec:glm-overdisp}

In practice, the Poisson model is often very useful for describing the relationship between the mean μ_i and the linear predictors, but typically underestimates the variance in the data. The consequence is that the Poisson standard errors are too small, rendering the Wald tests of coefficients, $z_j = \widehat{\beta}_j / se(\widehat{\beta}_j)$ (and other hypothesis test statistics) too large, and thus overly liberal.

In applications of the GLM, overdispersion is usually assessed by the likelihood-ratio test of the deviance (or the Pearson statistic) given in Section 9.1.3, but there is a subtle problem here. Lack of fit in a GLM for count data can result either from a mis-specified model for the systematic component (omitted or unmeasured predictors, non-linear relations, etc.) or from failure of the Poisson mean = variance assumption. Thus, use of these methods requires some high degree of confidence that the systematic part of the model has been correctly specified, so that any lack of fit can be attributed to overdispersion.

One way of dealing with this is to base inference on so-called *sandwich* covariance estimators that are robust against some types of model mis-specification. In R, this is provided by the `sandwich()` function in the sandwich package, and can be used with `coeftest(model, vcov=sandwich)` to give overdispersion-corrected hypothesis tests. Alternatively, the Poisson model variance assumption can be relaxed in the quasi-Poisson model and the negative-binomial model as discussed below.

9.3.1 The quasi-Poisson model

{sec:glm-quasi}

One obvious solution to the problem of overdispersion for count data is the relaxed assumption that the conditional variance is merely *proportional* to the mean,

$$\mathcal{V}(y_i | \eta_i) = \phi \mu_i$$

Overdispersion is the common case of $\phi > 1$, implying that the conditional variance increases faster than the mean, but the opposite case of underdispersion, $\phi < 1$ is also possible, though relatively rare in practice. This strategy entails estimating the dispersion parameter ϕ from the data, and gives the ***quasi-Poisson model*** for count data.

One possible estimate is the residual deviance divided by degrees of freedom. However, it is more common to use the Pearson statistics, that gives a method-of-moments estimate with improved statistical properties.

$$\widehat{\phi} = \frac{X_P^2}{n-p} = \sum_{i=1}^{n} \frac{(y_i - \widehat{\mu}_i)^2}{\widehat{\mu}_i} / (n-p)$$

It turns out that this model gives the same coefficient estimates as the standard Poisson GLM, but inference is adjusted for over/under dispersion. In particular, following Eqn. (9.1) the standard errors of the model coefficients are multiplied by $\widehat{\phi}^{1/2}$ and so are inflated when overdispersion is present. In R, the quasi-Poisson model with this estimated dispersion parameter is fitted with the `glm()` function, by setting `family=quasipoisson`.

{ex:phdpubs2}

EXAMPLE 9.3: Publications of PhD candidates

For the `PhdPubs` data, the deviance and Pearson estimates of dispersion ϕ can be calculated using the results of the Poisson model saved in the `phd.pois` object. The Pearson estimate, 1.83, indicates that standard errors of coefficients in this model should be multiplied by $\sqrt{1.83} = 1.35$, a 35% increase, to correct for overdispersion.

```
with(phd.pois, deviance/df.residual)

## [1] 1.797

sum(residuals(phd.pois, type = "pearson")^2)/phd.pois$df.residual

## [1] 1.83
```

The quasi-Poisson model is then fitted using `glm()` as:

```
phd.qpois <- glm(articles ~ ., data=PhdPubs, family=quasipoisson)
```

For use in other computation, the dispersion parameter estimate $\widehat{\phi}$ can be obtained as the `dispersion` value of the `summary()` method for a quasi-Poisson model.

```
(phi <- summary(phd.qpois)$dispersion)

## [1] 1.83
```

Note that this value can be compared to the variance/mean ratio of 2.91 calculated for the marginal distribution in Example 9.1; there is considerable improvement taking the predictors into account.

△

9.3.2 The negative-binomial model

{sec:glm-negbin}

The negative-biomial model for count data was introduced in Section 3.2.3 as a different generalization of the Poisson model that allows for overdispersion. In the context of the GLM, this can be developed as the extended form where the distribution of $y_i \,|\, \boldsymbol{x}_i$ where the mean μ_i for fixed $\boldsymbol{x}_i$ can vary across observations i according to a gamma distribution with mean μ_i and a constant shape parameter, θ, reflecting the additional variation due to heterogeneity.

For a fixed value of θ, the negative-binomial is another special case of the GLM. The expected value of the response is again $\mathcal{E}(y_i) = \mu_i$, but the variance function is $\mathcal{V}(y_i) = \mu_i + \mu_i^2/\theta$, so the variance of y increases more rapidly than that of the Poisson distribution. Some authors (e.g., Agresti (2013), Hilbe (2014)) prefer to parameterize the variance function in terms of $\alpha = 1/\theta$, giving

$$\mathcal{V}(y_i) = \mu_i + \mu_i^2/\theta = \mu_i + \alpha\mu_i^2 \ ,$$

so that α is a kind of dispersion parameter. Note that as $\alpha \to 0$, $\mathcal{V}(y_i) \to \mu_i$ and the negative-binomial converges to the Poisson.

The MASS package provides the family function `negative.binomial(theta)` that can be used directly with `glm()` provided that the argument `theta` is specified. One example would be the related geometric distribution (Section 3.2.4), that is the special case of $\theta = 1$. This can be fitted in R by setting `family=negative.binomial(theta=1)` in the call to `glm()`.

Most often, θ is unknown and must be estimated from the data. In this case, the negative-binomial model is not a special case of the GLM, but it is possible to obtain maximum likelihood estimates of both $\boldsymbol{\beta}$ and θ, by iteratively estimating $\boldsymbol{\beta}$ for fixed θ and vice-versa. This method is implemented in the `glm.nb()` in the package MASS.

{ex:crabs-nbin}

EXAMPLE 9.4: Mating of horseshoe crabs

For example, for the `CrabSatellites` data, we can fit the general negative-binomial model with θ free.

```
library(MASS)
crabs.nbin <- glm.nb(satellites ~ weight + color, data=CrabSatellites1)
crabs.nbin$theta

## [1] 0.9556
```

The estimated value $\widehat{\theta}$ returned by `glm.nb()` is not very far from 1. Hence, we might also consider fixing $\theta = 1$, as illustrated below.

```
crabs.nbin1 <- glm(satellites ~ weight + color, data=CrabSatellites1,
                   family=negative.binomial(1))
```

△

9.3.3 Visualizing the mean–variance relation

The quasi-Poisson and negative-binomial models have different variance functions, and one way to visualize which provides a better fit to the data is to group the data according to the fitted value of the linear predictor, calculate the mean and variance for each group, and then plot the variances against the means. A smoothed curve will then approximate the *empirical* mean–variance relationship. To this, we can add curves showing the mean–variance function implied by various models.[6]

{ex:phdpubs3}

EXAMPLE 9.5: Publications of PhD candidates

For the `PhdPubs` data, the fitted values are obtained with `fitted()` for the Poisson and negative binomial models. Either set can be used to categorize the observations into groups for the purpose of calculating means and variances of the response.

```
fit.pois <- fitted(phd.pois, type="response")
fit.nbin <- fitted(phd.nbin, type="response")
```

[6] This idea and the example that follows was suggested by Germán Rodrigues in a Stata example given at `http://data.princeton.edu/wws509/stata/overdispersion.html`.

Here we use a simpler version of the cutfac() function to group a numeric variable into quantile-based groups. cutq() also uses deciles by default, and just uses simple integer values for the factor labels.

```
cutq <- function(x, q = 10) {
    quantile <- cut(x, breaks = quantile(x, probs = 0:q/q),
        include.lowest = TRUE, labels = 1:q)
    quantile
}
```

Using this, we create a variable group giving 20 quantile groups of the fitted values, and then use aggregate() to find the mean and variance of the number of articles in each group.

```
group <- cutq(fit.nbin, q=20)
qdat <- aggregate(PhdPubs$articles,
          list(group),
          FUN = function(x) c(mean=mean(x), var=var(x)))
qdat <- data.frame(qdat$x)
qdat <- qdat[order(qdat$mean),]
```

We can then calculate the theoretical variances implied by the quasi-Poisson and negative-binomial models:

```
phi <- summary(phd.qpois)$dispersion
qdat$qvar <- phi * qdat$mean
qdat$nbvar <- qdat$mean + (qdat$mean^2) / phd.nbin$theta
head(qdat)

##      mean   var  qvar  nbvar
## 1 0.6122 0.784 1.121 0.7776
## 2 1.1489 1.782 2.103 1.7312
## 8 1.2444 2.462 2.278 1.9276
## 4 1.2609 1.708 2.308 1.9622
## 6 1.2727 1.831 2.330 1.9873
## 7 1.2979 4.344 2.376 2.0409
```

The plot, shown in Figure 9.8, then simply plots the points and uses lines() to plot the model-implied variances.

```
with(qdat, {
  plot(var ~ mean, xlab="Mean number of articles", ylab="Variance",
       pch=16, cex=1.2, cex.lab=1.2)
  abline(h=mean(PhdPubs$articles), col=gray(.40), lty="dotted")
  lines(mean, qvar, col="red", lwd=2)
  lines(mean, nbvar, col="blue", lwd=2)
  lines(lowess(mean, var), lwd=2, lty="dashed")
  text(3, mean(PhdPubs$articles), "Poisson", col=gray(.40))
  text(3, 5, "quasi-Poisson", col="red")
  text(3, 6.7, "negbin", col="blue")
  text(3, 8.5, "lowess")
})
```

We can see from this plot that the variances implied by the quasi-Poisson and negative-binomial models are in reasonable accord with the data and with each other up to a mean of about 2.5. They diverge substantially at the upper end, for the 20–30% of the most productive candidates, where the quadratic variance function of the negative-binomial provides a better fit.

△

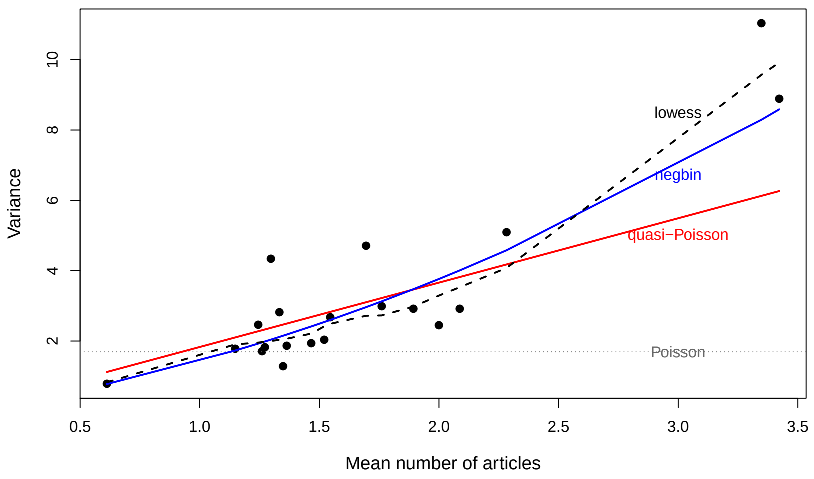

Figure 9.8: Mean–variance functions for the PhdPubs data. Points show the observed means and variances for 20 quantile groups based on the fitted values in the negative-binomial model. The labeled lines and curves show the variance functions implied by various models.

fig:phd-mean-var-plot

9.3.4 Visualizing goodness-of-fit

glm-visfit}

Even with correction for overdispersion, goodness-of-fit tests provide only an overall summary of model fit. Some specialized tests for particular forms of overdispersion are also available (e.g., see Cameron and Trivedi (1998, Chapter 5)), but these only identify general problems and cannot provide detailed indications of the possible source of these problems.

In Chapter 3, we illustrated the use of rootograms for visualizing goodness-of-fit to a wide variety discrete distributions using the `plot()` method for class "goodfit" objects with the vcd package. However, those methods were developed for one-way discrete distributions without explanatory variables.

Kleiber and Zeileis (2014) have generalized this idea to the wider class of GLM-related count regression models considered here. The countreg package provides a new implementation of `rootogram()` with methods for all of these models (and others not metioned). We illustrate these plots for the models considered to this point, and then extend this use for models allowing for excess zero counts in Section 9.4.

{ex:phdpubs4}

EXAMPLE 9.6: Publications of PhD candidates

For the `PhdPubs` data, Figure 9.9 shows hanging rootograms for the Poisson and negative-binomial models produced using `countreg::rootogram`[7] on the fitted model objects. We are looking both for general patterns of under/over fit, as well as counts that stand out as poorly fitted against the background.

```
library(countreg)
countreg::rootogram(phd.pois, max=12, main="PhDPubs: Poisson")
countreg::rootogram(phd.nbin, max=12, main="PhDPubs: Negative-Binomial")
```

[7] At the time of this writing, `rootogram` in countreg conflicts with the version in vcd, so we qualify the use here with the package name.

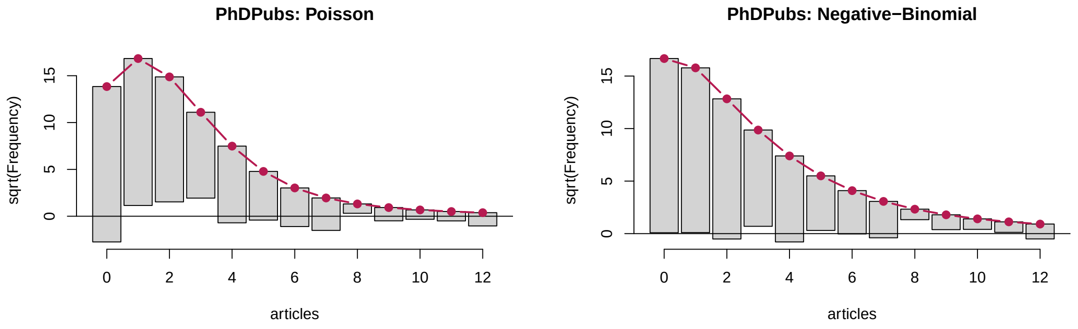

Figure 9.9: Hanging rootograms for the PhdPubs data.fig:phdpubs4-rootogram

The Poisson model shows a systematic, wave-like pattern with excess zeros, too few observed frequencies for counts of 1–3, but generally greater frequencies for counts of 4 or more. The negative-binomial model clearly fits much better, though there is a peculiar tendency among the smaller frequencies for 8 or more articles. △

{ex:crabs2}

EXAMPLE 9.7: Mating of horseshoe crabs

Figure 9.10 shows similar plots for the same two models fit to the number of crab satellites. The fit of the Poisson model clearly reveals the excess of zero male satellites. For the negative-binomial, the rootogram no longer exhibits same wave-like pattern, however, the underfitting of the count for 0 and overfitting for counts 1–2 is characteristic of data with excess zeros.

```
countreg::rootogram(crabs.pois, max=15, main="CrabSatellites: Poisson")
countreg::rootogram(crabs.nbin, max=15, main="CrabSatellites: Negative-Binomial")
```

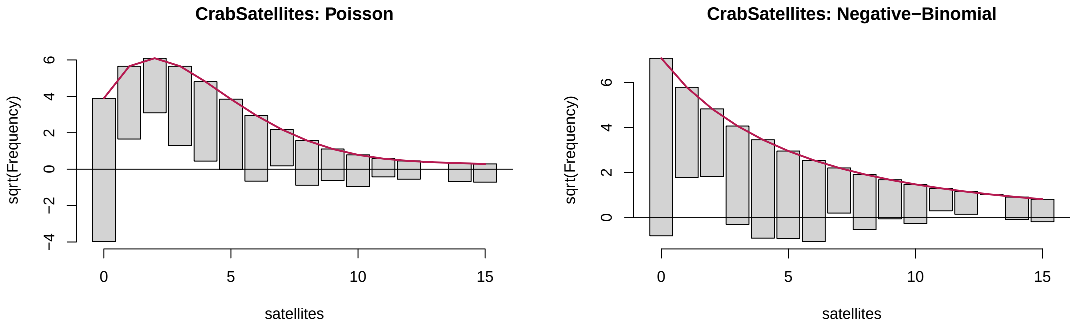

Figure 9.10: Hanging rootograms for the CrabSatellites data.fig:crabs2-rootogram

△

9.4 Models for excess zero counts

{sec:glm-zeros}

In addition to overdispersion, many sets of empirical data exhibit a greater prevalence of zero counts than can be accommodated by the Poisson or negative-binomial models. We saw this in the PhdPubs data set, where there were many candidates who had not published at all, and in the CrabSatellites data where a large number of females attracted no unattached males. Other examples abound in many different fields: studies of the use of health care services often find that many people never visit a hospital in some time frame; similarly, the distribution of insurance claims often shows large numbers who make no claims (Yip and Yau, 2005) because of under-reporting of small claims, policy deductible provisions and desire to avoid premium increases.

Beyond simply identifying this as a problem of lack-of-fit, understanding the reasons for excess zero counts can make a contribution to a more complete explanation of the phenomenon of interest, and this requires both new statistical models and visualization techniques illustrated in this section.

In the first example, Long (1997) argued that the PhD candidates might fall into two distinct groups: "publishers" (perhaps striving for an academic career) and "non-publishers" (seeking other career paths). Of the 275 observations having articles==0, some might not have published due to chance or unmeasured factors. One reasonable form of explanation is that the observed zero counts reflect a mixture of the two latent classes— those who simply have not yet published and those who will likely never publish. A statistical formulation of this idea leads to the class of *zero-inflated* models described below.

A different form of explanation is that there may be some special circumstance or "hurdle" required to achieve a positive count, like publishing the master's thesis (such as being driven internally by a personality trait or externally by pressure from a mentor). This idea leads to the class of *hurdle* models that entertain and fit (simultaneously) two separate models: one for the occurrrence of the zero counts, and one for the positive counts.

9.4.1 Zero-inflated models

{sec:glm-zip}

Zero-inflated models, introduced by Lambert (1992) as the ***zero-inflated Poisson*** (ZIP) model, provide an attractive solution to the problem of dealing with an overabundance of zero counts. It postulates that the observed counts arise from a mixture of two latent classes of observations: some structural zeros for whom y_i will always be 0, and the rest, sometimes giving random zeros. The ZIP model is comprised of two components:

- A model for the binary event of membership in the unobserved (latent) class of those for whom the count is necessarily zero (e.g., "non-publishers"). This is typically taken as a logistic regression for the probability π_i that observation i is in this class, with predictors $\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_q$, giving

$$\text{logit}(\pi_i) = \boldsymbol{z}_i^{\mathsf{T}} \boldsymbol{\gamma} = \gamma_0 + \gamma_1 z_{i1} + \gamma_2 z_{i2} + \cdots + \gamma_q z_{iq} \ . \tag{9.4}$$

{eq:zip-logit}

- A Poisson model for the other class (e.g., "publishers"), for whom the observed count may 0 or positive. This model typically uses the usual log link to predict the mean, using predictors $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_p$, so

$$\log_e \ \mu(\boldsymbol{x}_i) = \boldsymbol{x}_i^{\mathsf{T}} \boldsymbol{\beta} = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_q x_{ip} \ . \tag{9.5}$$

{eq:zip-pois}

In application, it is permissible and not uncommon to use the same set of predictors $x = z$ in both submodels, but the notation indicates that this is not required. Some simple special cases arise when the model for the always zero latent class is an intercept-only model, $\text{logit}(\pi_i) = \gamma_0$, implying the same probability for all individuals, and (less commonly) when the Poisson mean model is intercept-only with no predictors but there might be excess zero counts.

With this setup, one can show that the probability of observing counts of $y_i = 0$ and $y_i > 0$ are

$$\begin{aligned} \Pr(y_i = 0 \,|\, \boldsymbol{x}, \boldsymbol{z}) &= \pi_i + (1 - \pi_i) e^{-\mu_i} \\ \Pr(y_i \,|\, \boldsymbol{x}, \boldsymbol{z}) &= (1 - \pi_i) \times \left[\frac{{\mu_i}^{y_i} e^{-\mu_i}}{y_i!} \right] , \qquad y_i \geq 0 \end{aligned} \tag{9.6}$$

{eq:zip-pro

where the term in brackets in the second equation is the Poisson probability $\Pr(y = y_i)$ with rate parameter $\mathrm{Pois}(\mu_i)$. In these equations, $\pi_i = \mathrm{logit}^{-1}(\boldsymbol{z}_i^{\mathsf{T}} \boldsymbol{\gamma})$ depends on the $\boldsymbol{z}$ through Eqn. (9.4), and $\mu_i = \exp(\boldsymbol{x}^{\mathsf{T}} \boldsymbol{\beta})$ depends on the $\boldsymbol{x}$ through Eqn. (9.5).

The conditional expectation and variance of y_i then have the forms

$$\begin{aligned} \mathcal{E}(y_i) &= (1 - \pi_i)\, \mu_i \\ \mathcal{V}(y_i) &= (1 - \pi_i)\, \mu_i (1 + \mu_i \pi_i) \ . \end{aligned}$$

Thus, when $\pi_i > 0$, the mean of y is always less than μ_i, and the variance of y is greater than its mean by a dispersion factor of $(1 + \mu_i \pi_i)$.

There is nothing special about the use of the Poisson distribution here. The model for the count variable could also be taken as the negative-binomial, giving a *zero-inflated negative-binomial* (ZINB) model using $\mathrm{NBin}(\mu, \theta)$ or a *zero-inflated geometric* model using $\mathrm{NBin}(\mu, \theta = 1)$.

There are several packages in R capable of fitting zero-inflated models. The most mature and complete of these is `zeroinfl()` in the countreg package (a successor to the pscl package) The function `zeroinfl()` is modeled after `glm()`, but provides an extended syntax for the model formula.

If the `formula` argument is supplied in the form `y ~ x1 + x2 + ...`, it not only describes the count regression of y on $x_1, x_2, \ldots$, but also implies that the *same* set of regressors, $z_j = x_j$, is used for the zero count binary submodel. The extended syntax uses the notation `y ~ x1 + x2 + ... | z1 + z2 + ...` to specify the x variables separately, conditional on (`|`) the always-zero count model `y ~ z1 + z2 + ...`. The model for the not-always-zero class can be specified using the `dist` argument, with possible values `"poisson"`, `"negbin"` and `"geometric"`.

{sec:glm-hurdle}

9.4.2 Hurdle models

A different class of models capable of accounting for excess zero counts is the ***hurdle model*** proposed initially by Cragg (1971) and developed further by Mullahy (1986). This model also uses a separate logistic regression submodel to distinguish counts of $y = 0$ from larger counts, $y > 0$. The submodel for the positive counts is expressed as a (left) *truncated* Poisson or negative-binomial model, excluding the zero counts. As an example, consider a study of behavioral health in which one outcome is the number of cigarettes smoked in one month. All the zero counts will come from non-smokers and smokers will nearly always smoke a positive number.

This differs from the set of ZIP models in that classes of $y = 0$ and $y > 0$ are now considered fully-observed, rather than latent. Conceptually, there is one process and submodel acounting for the zero counts and a separate process accounting for the positive counts, once the "hurdle" of

$y = 0$ has been passed. In other words, for ZIP models, the first process generates only extra zeros beyond those of the regular Poisson distribution. For hurdle models, the first process generates all of the zeros. The probability equations corresponding to Eqn. (9.6) are:

$$\begin{aligned} \Pr(y_i = 0 \,|\, \boldsymbol{x}, \boldsymbol{z}) &= \pi_i \\ \Pr(y_i \,|\, \boldsymbol{x}, \boldsymbol{z}) &= \frac{(1-\pi_i)}{1-e^{-\mu_i}} \times \left[\frac{\mu_i{}^{y_i} e^{-\mu_i}}{y_i!} \right] \quad , \qquad y_i \geq 0 \end{aligned} \tag{9.7}$$

The hurdle model can be fitted in R using the `hurdle()` function from the countreg package. The syntax for the model formula is the same extended form provided by `zeroinfl()`, where `y ~ x1 + x2` uses the same regressors for the zero and positive count submodels, while `y ~ x1 + x2 | z1 + z2` uses `y ~ z1 + z2` for the zero hurdle model. Similarly, the count distribution can be given as `"poisson"`, `"negbin"` or `"geometric"` with the `dist` argument. For `hurdle()`, the distribution for zero model can be specified with a `zero.dist` argument. The default is `"binomial"` (with a logit `link`), but other right-censored distributions can also be specified.

9.4.3 Visualizing zero counts

{sec:glm-viszero}

Both the zero-inflated and hurdle models treat the zero counts $y = 0$ specially with separate submodels, so the binary event of $y = 0$ vs. $y > 0$ can be visualized using any of the techniques illustrated in Chapter 7. See Section **??**, Section **??** and Section **??** for some examples that plot both the binary observations and a model summary or smoothed curve to show the relationships with one or more regressors. To apply these ideas in the current context, simply define or plot a logical variable corresponding to the expression `y==0`, giving values of `TRUE` or `FALSE`.

A different, and simpler idea is illustrated here using what is called a ***spine plot*** Hummel (1996) when a predictor x is a discrete factor or ***spinogram*** when x is continuous. Both are forms of mosaic plots with special formatting of spacing and shading, and in this context they plot $\Pr(y = 0|x)$ against $\Pr(x)$; when x is numerical, it is first made discrete, as in a histogram. Then, in the spine plot or spinogram, the widths of the bars correspond to the relative frequencies of x and heights of the bars correspond to the conditional relative frequencies of $y = 0$ in every x group. In R, spine plots are implemented in the function `spineplot()`, however, this is what you get by default if you use `plot(y==0 ~ x)` to plot the binary factor against any regressor x.

A related graphical method is the ***conditional density plot*** (Hofmann and Theus, 2005). The conditional probabilities $\Pr(y = 0|x)$ are derived using a smoothing approach (via `density()`) over x rather than by making x discrete. These plots are provided by `cdplot()` in the graphics package and a similar `cd_plot()` in vcd. The smoothing method for the density estimate is controlled by a `bw` (bandwith) method and other arguments.

{ex:crabs-zero}

Example 9.8: Mating of horseshoe crabs

For the `CrabSatellites` data, we can examine the relationship of the zero counts (females who attract no unattached male satellites) to the predictors using spinograms or conditional density plots. Here, we consider *`weight`* and *`color`* (treated numerically) as predictors. **TODO**: Fixup use of color in Example 9.2 so it doesn't cause a problem here

```r
op <- par(cex.lab=1.2, mfrow = c(1, 2))
plot(factor(satellites == 0) ~ weight, data = CrabSatellites,
     breaks = quantile(weight, probs=seq(0,1,.2)), ylevels=2:1,
     ylab="No satellites")
plot(factor(satellites == 0) ~ color, data = CrabSatellites,
     breaks = quantile(color, probs=seq(0,1,.33)),  ylevels=2:1,
     ylab="No satellites")
par(op)
```

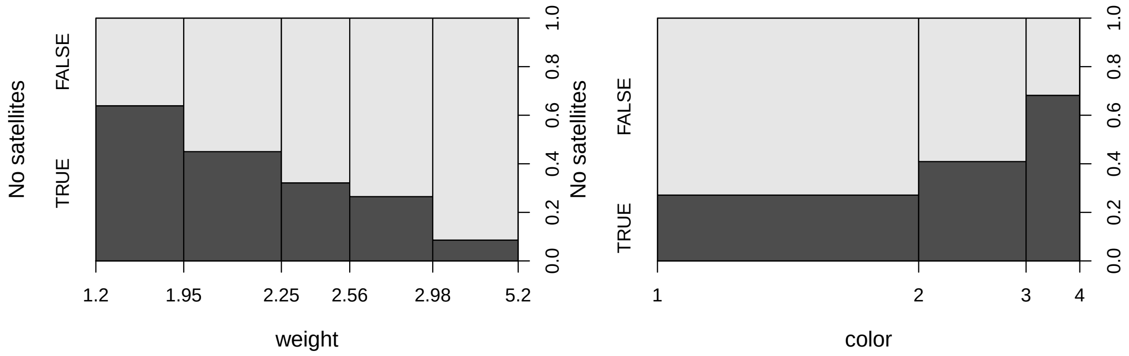

Figure 9.11: Spinograms for the CrabSatellites data. The variables weight (left) and color(right) have been made discrete using quantiles of their distributions. fig:crabs-zero-spinogram

```r
op <- par(cex.lab=1.2, mfrow = c(1, 2))
cdplot(factor(satellites == 0) ~ weight, data = CrabSatellites,
       ylevels=2:1, ylab="No satellites")
cdplot(factor(satellites == 0) ~ color, data = CrabSatellites,
       ylevels=2:1, , ylab="No satellites")
par(op)
```

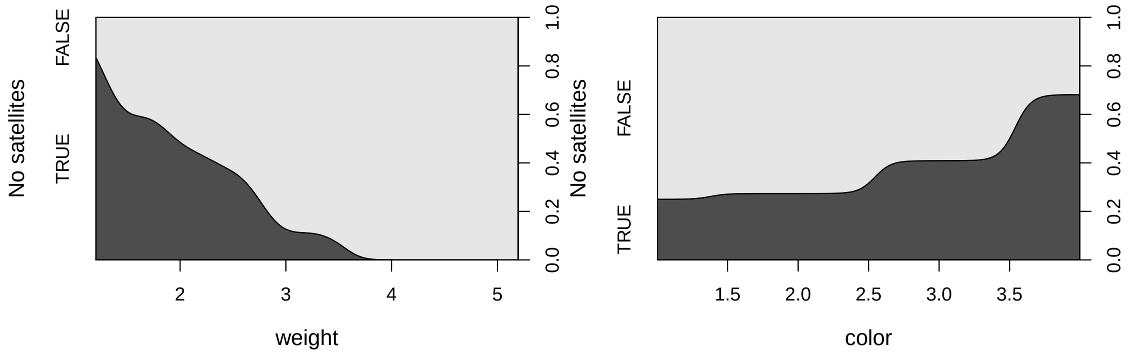

Figure 9.12: Conditional density plots for the CrabSatellites data. The region shaded below shows the conditional probability density estimate for a count of zero. fig:crabs-zero-cdplot

△

9.5 Diagnostic plots for model checking

c:glm-diag}

9.6 Chapter summary

9.7 Further reading

9.8 Lab exercises

```
# detach(package:ggtern)  ## detach any masking packages
.locals$ch09 <- setdiff(ls(), .globals)
.locals$ch09

##  [1] "art.fac"          "art.tab"          "blogits"
##  [4] "crabs.nbin"       "crabs.nbin1"      "crabs.pois"
##  [7] "crabs.pois1"      "CrabSatellites"   "CrabSatellites1"
## [10] "cutfac"           "cutq"             "fit.nbin"
## [13] "fit.pois"         "group"            "interp"
## [16] "knitrSet"         "logi.hist.plot"   "logit2p"
## [19] "LRtest"           "op"               "phd.nbin"
## [22] "phd.pois"         "phd.qpois"        "PhdPubs"
## [25] "phi"              "print_coef"       "pun_cotab"
## [28] "qdat"             "spar"

remove(list=.locals$ch09[sapply(.locals$ch09,function(n){!is.function(get(n))})])
```

References

Agresti, A. (2013). *Categorical Data Analysis*. Wiley Series in Probability and Statistics. New York: Wiley-Interscience [John Wiley & Sons], 3rd edn.

Brockmann, H. J. (1996). Satellite male groups in horseshoe crabs, *Limulus polyphemus*. *Ethology*, 102(1), 1–21.

Cameron, A. C. and Trivedi, P. K. (1998). *Regression analysis of count data*. Econometric society monographs. Cambridge (U.K.), New York: Cambridge University Press.

Cragg, J. G. (1971). Some statistical models for limited dependent variables with application to the demand for durable goods. *Econometrica*, 39, 829–844.

Fox, J. (2008). *Applied Regression Analysis and Generalized Linear Models*. Thousand Oaks, CA: Sage, 2nd edn.

Hilbe, J. M. (2014). *Modeling Count Data*. New York, NY: Cambridge University Press.

Hofmann, H. and Theus, M. (2005). Interactive graphics for visualizing conditional distributions.

Hummel, J. (1996). Linked bar charts: Analyzing categorical data graphically. *Computational Statistics*, 11, 23–33.

Kleiber, C. and Zeileis, A. (2014). Visualizing count data regressions using rootograms. Working papers, Faculty of Economics and Statistics, University of Innsbruck.

Lambert, D. (1992). Zero-inflated poisson regression, with an application to defects in manufacturing. *Technometrics*, 34, 1–14.

Long, J. S. (1997). *Regression Models for Categorical and Limited Dependent Variables*. Thousand Oaks, CA: Sage Publications.

McCulloch, C. E. and Neuhaus, J. M. (2005). Generalized linear mixed models. In *Encyclopedia of Biostatistics*. John Wiley & Sons, Ltd.

Mullahy, J. (1986). Specification and testing of some modified count data models. *Journal of Econometrics*, 33, 341–365.

Nelder, J. A. and Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society, Series A*, 135, 370–384.

Yip, K. C. and Yau, K. K. (2005). On modeling claim frequency data in general insurance with extra zeros. *Insurance: Mathematics and Economics*, 36(2), 153–163.