Journal of Statistical Software

November 2017, Volume 81, Issue 10. *doi: 10.18637/jss.v081.i10*

Half-Normal Plots and Overdispersed Models in R: The hnp Package

Rafael A Moral
Universidade de São Paulo

John Hinde
NUI Galway

Clarice G B Demétrio
Universidade de São Paulo

Abstract

Count and proportion data may present overdispersion, i.e., greater variability than expected by the Poisson and binomial models, respectively. Different extended generalized linear models that allow for overdispersion may be used to analyze this type of data, such as models that use a generalized variance function, random-effects models, zero-inflated models and compound distribution models. Assessing goodness-of-fit and verifying assumptions of these models is not an easy task and the use of half-normal plots with a simulated envelope is a possible solution for this problem. These plots are a useful indicator of goodness-of-fit that may be used with any generalized linear model and extensions. For GLIM users, functions that generated these plots were widely used, however, in the open-source software R, these functions were not yet available on the Comprehensive R Archive Network (CRAN). We describe a new package in R, **hnp**, that may be used to generate the half-normal plot with a simulated envelope for residuals from different types of models. The function `hnp()` can be used together with a range of different model fitting packages in R that extend the basic generalized linear model fitting in `glm()` and is written so that it is relatively easy to extend it to new model classes and different diagnostics. We illustrate its use on a range of examples, including continuous and discrete responses, and show how it can be used to inform model selection and diagnose overdispersion.

Keywords: goodness-of-fit, generalized linear models, mixed models, R.

1. Introduction

An important step of statistical modeling of any sort is to perform diagnostic analyses to assess goodness-of-fit. Several problems arise when model assumptions are not met such as misleading estimates, standard errors and p values and/or wrong conclusions about the process being studied. Among the reasons for a poorly-fitted model, the most common ones are

- measurement error of observed and/or explanatory variables (including typos);
- incomplete or inadequate linear predictor to describe the systematic structure of the data;
- incorrect specification of the error distribution or link function;
- unmodelled overdispersion;
- combination of one or more of the above.

When fitting linear models under the normality assumption, goodness-of-fit can be checked using formal tests, such as the Shapiro-Wilk test for residual normality (Shapiro and Wilk 1965), or the Bartlett test for variance homogeneity (Bartlett 1937). However, these tests may fail under many circumstances, such as small sample sizes, and usually graphical techniques provide a better assessment for model goodness-of-fit. These techniques include plotting different types of residuals or influence measures (e.g., leverage and Cook's distance). Several types of residuals may be used (e.g., standardized residuals, studentized residuals, deletion residuals, Pearson residuals, deviance residuals, etc.). However, for other types of model, diagnostic checking may be problematic. Useful diagnostic-checking plots include

- residuals vs. explanatory variables – indicates whether higher-order terms should be included in the linear predictor or the need for transformation of the response and/or explanatory variables (for quantitative explanatory variables);
- residuals vs. explanatory variables not included in the model – a systematic relationship indicates that the explanatory variables should be included in the linear predictor;
- added-variable plot – detects the relationship of the response variable with an explanatory variable not included in the model allowing for the effects of other variables;
- residuals vs. fitted values – may reveal variance heterogeneity and/or outliers;
- (half-)normal plot of residuals – detects outliers and indicates whether the error distribution was specified appropriately.

Under a normally distributed error assumption, when a model fits the data well, it is expected that studentized residuals follow a t distribution on the residual degrees of freedom, which for large samples converges to the standard normal distribution. In this case, the half-normal plot of these residuals should show a straight 45° line. However, it is hard to interpret whether points are sufficiently aligned or if the inevitable irregularities are caused by something other than random fluctuations. Another difficulty posed by these plots is that the ordering of the observations may induce dependence. For generalized linear models, this plot may have several forms depending on the variance and link functions and response variable distribution. Atkinson (1985) proposes the addition of a simulated envelope so that interpretation is more straightforward. For a well-fitted model the envelope is such that model diagnostics are likely to fall within it. The purpose is not to provide a region for acceptance or rejection of observations but to serve as a guide of what to expect under a well-fitted model. These plots are also useful for detecting possible outliers, overdispersion, and if the link function and/or error distribution were properly specified (Demétrio, Hinde, and Moral 2014).

When fitting generalized linear models, or different types of extended models (e.g., zero-inflated models and mixed models), half-normal plots with simulated envelopes are useful to assess goodness-of-fit, especially when analyzing overdispersed data. Demétrio and Hinde (1997) wrote GLIM4 macros (Francis, Green, and Payne 1993) to produce these plots for different overdispersion models. We developed the R (R Core Team 2017) package **hnp** (Moral, Hinde, and Demétrio 2017) that provides functions for generating half-normal plots with a simulated envelope for a range of generalized linear models and extensions. The scope of the `hnp()` function can be easily extended to include different diagnostics and models by the user-specification of appropriate simulation, model fitting, and diagnostic extraction codes. Package **hnp** is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=hnp`.

2. Generalized linear models and overdispersion

The generalized linear models (GLMs) framework of statistical modeling, formulated by Nelder and Wedderburn (1972), provides a unified theory for the application of normal, binomial, Poisson, gamma and inverse Gaussian regression models and brings together methods for the analysis of count, proportion, continuous measurement and time to event data, that until then were studied separately. As well as building on the standard normal regression model, this theory also generalizes the ideas of analysis of variance through the analysis of deviance.

For random variables $Y_1, \ldots, Y_n$, with probability function (pf) (Y_i discrete) or probability density function (pdf) (Y_i continuous) $f(y_i; \theta_i)$, where θ_i is associated with explanatory variables $x_1, \ldots, x_p$, statistical modeling will typically be based on a random sample of n observations $(y_i; \mathbf{x}_i), \mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\top$. A GLM is defined in terms of three basic components. The first component, called the random component, is represented by the random variables $Y_1, \ldots, Y_n$ following a distribution which belongs to the exponential family of distributions and differs only in terms of a parameter θ_i. Writing the exponential family in canonical form, the pf or pdf is given by:

$$f(y_i; \theta_i, \phi) = \exp\{\phi^{-1}[y_i\theta_i - b(\theta_i)] + c(y_i; \phi)\}, \tag{1}$$

where $b(\cdot)$ and $c(\cdot)$ are known functions, $\phi > 0$ is a known dispersion parameter and θ_i is called the canonical parameter. The normal, Poisson, binomial, gamma, inverse Gaussian and negative binomial distributions (with suitable definitions of dispersion parameters) all belong to the exponential family and can be expressed in the canonical form (1). For this family of distributions we have that $\mathsf{E}(Y_i) = \mu_i = b'(\theta_i)$ and $\mathsf{VAR}(Y_i) = \phi b''(\theta_i) = \phi V_i$, where $V_i = V(\mu_i) = d\mu_i/d\theta_i$ is called variance function and depends only on the mean μ_i.

Writing $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$ as the design (model) matrix and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$ as the vector of unknown parameters, the second component is a linear predictor $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}$, called the systematic component. Finally, the third component, the link function $g(\cdot)$, which is a monotonic and differentiable function, links the random and systematic components by relating the mean to the linear predictor through $\eta_i = g(\mu_i)$.

Nelder and Wedderburn (1972) proposed the scaled deviance as an overall discrepancy quantity to measure the fit of a GLM . This is given by

$$S_p = 2(\hat{l}_n - \hat{l}_p), \tag{2}$$

where $\hat{l}_p$ and $\hat{l}_n$ are the maximum of the log-likelihood functions for the current and saturated models, respectively, and the saturated model is one whose fit reproduces the observed data and typically has n parameters. These maximized likelihoods may be written as

$$\hat{l}_n = \phi^{-1}\sum_{i=1}^{n}[y_i\tilde{\theta}_i - b(\tilde{\theta}_i)] + \sum_{i=1}^{n} c(y_i, \phi)$$

and

$$\hat{l}_p = \phi^{-1}\sum_{i=1}^{n}[y_i\hat{\theta}_i - b(\hat{\theta}_i)] + \sum_{i=1}^{n} c(y_i, \phi),$$

where $\tilde{\theta}$ and $\hat{\theta}$ are the maximum likelihood estimates of the canonical parameter for the saturated and current models, respectively, hence we may also write the scaled deviance as

$$S_p = 2\phi^{-1}\sum_{i=1}^{n}[y_i(\tilde{\theta}_i - \hat{\theta}_i) + b(\hat{\theta}_i) - b(\tilde{\theta}_i)]. \tag{3}$$

Another important overall discrepancy measure of a fitted model is the generalized Pearson statistic X_p^2, given by

$$X_p^2 = \sum_{i=1}^{n}\frac{(y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)}. \tag{4}$$

For the true model and a known dispersion parameter ϕ both the scaled deviance and the scaled generalized Pearson statistic follow, asymptotically, a χ^2 distribution with $n-p$ degrees of freedom, although Jørgensen (2002) recommends the use of the scaled X_p^2 statistic as convergence to the reference distributions is more rapidly achieved as the sample size n increases. For further details on GLM theory, see McCullagh and Nelder (1989). When ϕ is unknown, there is no formal goodness-of-fit test available based on S_p or scaled X_p^2 and in this case X_p^2 is often used to estimate ϕ (Jørgensen 2002).

To analyze count and proportion data the Poisson and binomial models are, respectively, reasonable first choices. For these distributions the dispersion parameter is fixed and equal to 1. For a well-fitted model it is expected that the residual deviance and the generalized Pearson statistic should be approximately equal to the residual degrees of freedom, the expected value of the χ^2 reference distribution. When this does not happen, the model may not fit the data well (for various reasons, such as a misspecified linear predictor or link function, outliers, etc.), or simply, the variation in the data may be larger than is predicted by these models, a phenomenon often referred to as overdispersion, see Hinde and Demétrio (1998). There are different causes of overdispersion and failure to take it into account may result in misleading inferences. For a more thorough discussion, see Demétrio *et al.* (2014).

These basic models can be extended to incorporate overdispersion in several ways. A relatively simple extension involves the specification of a more general variance function. For example, to analyze count data, suppose that $Y_i \sim \mathrm{P}(\mu_i)$, then the Poisson model assumes that $\mathsf{E}(Y_i) = \mu_i$ and that $\mathsf{VAR}(Y_i) = \mu_i$. Then, a simple extension is to take $\mathsf{VAR}(Y_i) = \phi\mu_i$ and use a quasi-likelihood approach to estimate $\phi > 1$, as the variance is greater than expected under the Poisson model. This specification of the variance function is called constant overdispersion. This may be generalized further by taking

$$\mathsf{VAR}(Y_i) = \mu_i(1 + \phi\mu_i^{\delta}). \tag{5}$$

For $\phi = 0$, this is simply the variance function of the standard Poisson model; for $\delta = 0$ it gives the constant overdispersion case; while if $\delta = 1$ we get the same type of quadratic variance function as for the negative binomial type-II model.

A similar approach may also be adopted for proportion data. Suppose that $Y_i \sim \mathrm{B}(m_i, \pi_i)$, then the standard binomial model assumes that $\mathsf{E}(Y_i) = m_i\pi_i$ and $\mathsf{VAR}(Y_i) = m_i\pi_i(1 - \pi_i)$. If we now take an extended variance function

$$\mathsf{VAR}(Y_i) = m_i\pi_i(1 - \pi_i)\{1 + \phi(m_i - 1)^{\delta_1}[\pi_i(1 - \pi_i)]^{\delta_2}\}, \tag{6}$$

for $\phi = 0$, the expression is the same as for the standard binomial model; for $\delta_1 = \delta_2 = 0$, we obtain the constant overdispersion variance function; while if $\delta_1 = 1$ and $\delta_2 = 0$, this is the same form of variance function as the beta-binomial model.

Another way to model overdispersion consists in adding an observation-level random effect in the linear predictor, i.e.,

$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \sigma\mathbf{z},$$

where $\mathbf{z} = (z_1, z_1, \ldots, z_n)^\top$ is the vector of random effects, and typically we assume that $Z_i \sim \mathrm{N}(0, 1)$. For count data, this leads to the Poisson-normal model. For proportion data, when the logit link is used, this results in the binomial-logit-normal model. This approach is closely related to the ideas in two-stage models where the basic parameter of interest is assumed to be an individual level random variable. This leads to negative binomial models for count data (by assuming the Poisson mean to have a gamma distribution) and the beta-binomial model for counted proportions (by assuming the binomial probability to have a beta distribution). Further details on overdispersion models for count and proportion data may be found in Hinde and Demétrio (1998) and Demétrio *et al.* (2014).

3. Half-normal plots with simulated envelopes

This relatively easy technique consists in plotting the ordered absolute values of a model diagnostic versus the expected order statistics of a half-normal distribution, which can be approximated as

$$\Phi^{-1}\left(\frac{i + n - \frac{1}{8}}{2n + \frac{1}{2}}\right), \tag{7}$$

where i is the ith order statistic, $1 \leq i \leq n$ and n is the sample size, as in McCullagh and Nelder (1989, p. 407), following the results from Blom (1958) and Royston (1982). For a normal plot we use the ordered values versus the expected order statistics of a normal distribution, approximated as

$$\Phi^{-1}\left(\frac{i - \frac{3}{8}}{n + \frac{1}{4}}\right). \tag{8}$$

These order statistics are easily obtained using the R function `qnorm`, e.g., for $n = 7$ the expected order statistics of the half-normal distribution are

```
R> i <- 1:7
R> n <- 7
R> qnorm((i + n - 1/8) / (2 * n + 1/2))
```

```
[1] 0.1082554 0.2847156 0.4705935 0.6744898 0.9114298 1.2155984 1.7157550
```

Obtaining the simulated envelope for a half-normal plot is simple and consists of

1. fitting a model;
2. extracting model diagnostics and calculating sorted absolute values;
3. simulating 99 (or more) response variables using the same model matrix, error distribution and parameter estimates;
4. fitting the same model to each simulated response variable and extracting the same model diagnostics, and again sorting the absolute values;
5. computing the desired percentiles (e.g., 2.5 and 97.5) of the simulated diagnostic values at each value of the expected order statistic and using these to form the envelope.

3.1. Implemented model classes

Function `hnp()` handles many different model classes and more will be implemented as time goes by. So far, a range of generalized linear models and extensions are included:

- models for continuous data – Gaussian (`lm`, `aov` and `glm` functions), gamma (`glm` function), and inverse Gaussian (`glm` function) models;
- models for count data – Poisson and quasi-Poisson (`glm` function), negative binomial type-II (`glm.nb` function in package **MASS**, Venables and Ripley 2002; or `aodml` function in package **aods3**, Lesnoff and Lancelot 2013), and hurdle Poisson and negative binomial (`hurdle` function in package **pscl**, Zeileis, Kleiber, and Jackman 2008) models;
- models for proportion data – binomial and quasi-binomial (`glm` function), beta-binomial (`vglm` function in package **VGAM**, Yee 2010; `aodml` function in package **aods3**, `gamlss` function in package **gamlss**, Rigby and Stasinopoulos 2005; Stasinopoulos and Rigby 2007; `glmmadmb` function in package **glmmADMB**, Skaug, Fournier, Bolker, Magnusson, and Nielsen 2014), and multinomial (`multinom` function in package **nnet**, Venables and Ripley 2002) models;
- models for zero-inflated data – zero-inflated Poisson and negative binomial (`zeroinfl` function in package **pscl**), zero-inflated binomial (`vglm` function in package **VGAM**, `gamlss` function in package **gamlss**, `glmmadmb` function in package **glmmADMB**), and zero-inflated beta-binomial (`gamlss` function in package **gamlss**, `glmmadmb` function in package **glmmADMB**) models;
- mixed models – Gaussian (`lmer` function in package **lme4**, Bates, Mächler, Bolker, and Walker 2015; Doran, Bates, Bliese, and Dowling 2007), Poisson-normal and binomial-normal (`glmer` function in package **lme4**) models.

3.2. Simulation procedures

For most of the implemented model classes, the simulation procedures are already implemented in the R base packages or in the packages being used to fit the models, such as the `rbinom`, `rpois`, `rmultinom`, `rgamma` and `rnorm` functions for binomial, Poisson, multinomial, gamma and normal models. Package **mgcv**'s (Wood 2006) function `rig` was used for inverse Gaussian models; package **gamlss**'s (Rigby and Stasinopoulos 2005; Stasinopoulos and Rigby 2007) functions `rBB`, `rZIBI` and `rZIBB` were used for beta-binomial, zero-inflated binomial and zero-inflated beta-binomial models fitted using `gamlss`; package **VGAM**'s (Yee 2010) functions `rbetabinom` and `rzibinom` were used for beta-binomial and zero-inflated binomial models fitted using `vglm`; package **lme4**'s (Bates *et al.* 2015; Doran *et al.* 2007) function `simulate` was used for generalized linear mixed models fitted using `lmer` and `glmer`; package **MASS**'s (Venables and Ripley 2002) function `rnegbin` for negative binomial models.

For the quasi-binomial model the random samples were simulated using `rbinom` and then multiplied by ϕ (`summary(model)$dispersion`) and the residuals were then scaled by $1/\sqrt{\phi}$. For the quasi-Poisson model, the function `rnbinom` was adapted with the `size` argument set to $\mu/(\phi - 1)$. New functions were written for zero-inflated binomial and zero-inflated beta-binomial models fitted with **glmmADMB** (Skaug *et al.* 2014) and for zero-inflated and hurdle Poisson and negative binomial models fitted with **pscl** (Zeileis *et al.* 2008). The simulation procedures are all accessible in the code of package **hnp**.

3.3. New class implementation

To produce the half-normal plot with a simulated envelope, three procedures are required: (i) one to extract a diagnostic measure from the fitted model, (ii) one to simulate response variables using information from the model (error distribution, model matrix and fitted parameters), and finally (iii) one to refit the same model to the simulated data. The `hnp` function firstly recognizes the model class of the fitted object. If this class is not yet implemented, it returns an error. However, users may opt to supply their own diagnostic extraction, simulation and model fitting functions so that the half-normal plot is produced, see Section 4 for a practical guide.

4. Examples

Package **hnp** provides a range of examples drawn from Demétrio *et al.* (2014) and two of them will be discussed below for overdispersed proportion and count data. An example on orange tree embryogenesis (Tomaz, Mendes, Filho, Demétrio, Jansakul, and Rodriguez 1997) will be used to discuss new class implementation and an example on leukemia recurrence times (Miller 1997) will be used to discuss the survival analysis models implementation.

4.1. Overdispersed proportion data

A major pest of stored maize in Brazil is *Sitophilus zeamais*. In an experiment to assess the insecticide action of organic extracts of *Annona mucosa* (Annonaceae), Petri dishes containing 10g of corn were treated with extracts prepared with different parts of the plant (seeds, leaves and branches) at a concentration of 1500mg/kg or just water (control), using a completely randomized design with 10 replicates. Then 20 *Sitophilus zeamais* adults were placed in each Petri dish and, after 60 days, the numbers of damaged and undamaged corn grains were counted, see Ribeiro *et al.* (2013).

We begin by fitting a standard binomial model, i.e., $Y_{ij} \sim \mathrm{B}(m_{ij}, \pi_{ij})$, to the data using the logit link with the following linear predictor:

$$\log\left(\frac{\pi_{ij}}{1-\pi_{ij}}\right) = \beta_0 + e_i, \tag{9}$$

where β_0 is the intercept and e_i is the effect of the ith extract, $i = 1, \ldots, 4$. We can fit this model in R and produce a simple analysis of deviance table using `glm()` and `anova()`:

```
R> library("hnp")
R> data("corn", package = "hnp")
R> fit1_b <- glm(cbind(y, m - y) ~ extract, family = binomial, data = corn)
R> anova(fit1_b, test = "Chisq")

Analysis of Deviance Table

Model: binomial, link: logit

Response: cbind(y, m - y)

Terms added sequentially (first to last)


        Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                       39     801.96
extract  3   636.04        36     165.92 < 2.2e-16 ***
--
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The residual deviance is much larger than the number of residual degrees of freedom, indicating that the model does not fit the data well. We can easily check this by producing the half-normal plot with simulation envelope, see Figure 1(a) and it is clear that this is not a good model fit.

```
R> set.seed(1234)
R> hnp(fit1_b, xlab = "Half-normal scores", ylab = "Deviance residuals",
+    main = "(a) Binomial model", pch = 4)
```

By default the deviance residuals have been used. Users may change colors, sizes or point character symbols freely using the same arguments passed to `plot()` and `par()`. We now turn into fitting overdispersion models, and begin by considering a quasi-likelihood approach:

```
R> fit2_b <- glm(cbind(y, m - y) ~ extract, family = quasibinomial,
+    data = corn)
R> summary(fit2_b)$dispersion

[1] 4.409755

R> hnp(fit2_b, xlab = "Half-normal scores", ylab = "Deviance residuals",
+    main = "(b) Quasi-binomial model", pch = 4)
```

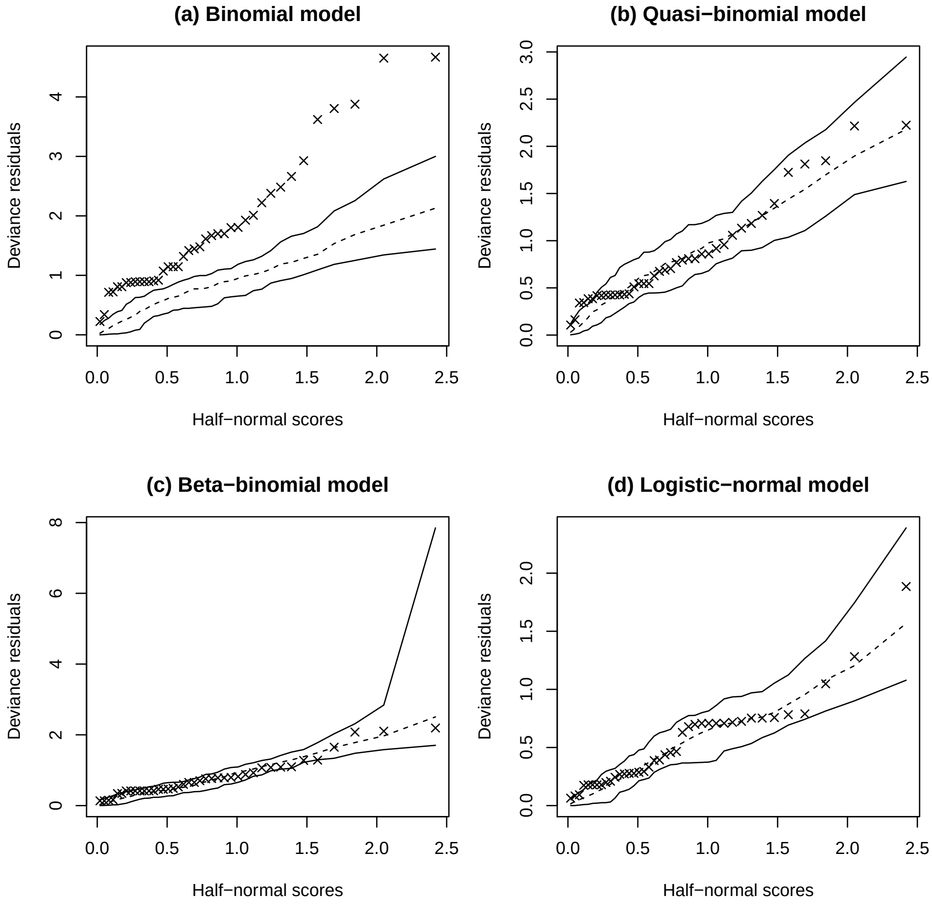

Figure 1: Half-normal plots with simulated envelopes for the deviance residuals from (a) the binomial model, (b) the quasi-binomial model, (c) the beta-binomial model, and (d) the binomial-logit-normal model, fitted to the corn data.

We observe that the estimated dispersion parameter is $\hat{\phi} = 4.41$ and the half-normal plot indicates that this model fit was satisfactory with most of the deviance residuals lying within the simulated envelope, see Figure 1(b). We may also easily fit other overdispersion models, such as the beta-binomial model using the same linear predictor (9) or the binomial-logit-normal model in which a random effect is included in the linear predictor:

$$\log\left(\frac{\pi_{ij}}{1-\pi_{ij}}\right) = \beta_0 + e_i + \sigma Z_{ij}, \tag{10}$$

where $Z_{ij} \sim \mathrm{N}(0, \sigma^2)$. To fit these models, we make use of the packages **aods3** (Lesnoff and Lancelot 2013) and **lme4** (Bates *et al.* 2015; Doran *et al.* 2007):

```
R> library("aods3")
R> fit3_b <- aodml(cbind(y, m - y) ~ extract, family = "bb", data = corn)
R> hnp(fit3_b, xlab = "Half-normal scores", ylab = "Deviance residuals",
+    main = "(c) Beta-binomial model", pch = 4)
R> library("lme4")
```

```
R> x <- factor(seq_len(nrow(corn)))
R> fit4_b <- glmer(cbind(y, m - y) ~ extract + (1 | x),
+    family = binomial, data = corn)
R> hnp(fit4_b, xlab = "Half-normal scores", ylab = "Deviance residuals",
+    main = "(d) Binomial-logit-normal model", pch = 4)
```

Both resulting half-normal plots with simulation envelopes show that these are also satisfactory models for this data set, see Figures 1(c) and 1(d).

The three overdispersion models seem to fit the data equally well, which is not a surprise because the binomial sample sizes do not vary much (from 32 to 39) and when they are equal the beta-binomial and binomial-logit-normal variance functions both reduce to the constant overdispersion form, which may be a reasonable choice for this case, see Ribeiro *et al.* (2013).

4.2. Overdispersed count data

For the same experiment described in the previous Section we now turn to focus on the number of emerged insects (progeny) after 60 days, see Ribeiro *et al.* (2013). We begin by fitting a standard Poisson model, i.e., $Y_{ij} \sim \mathrm{P}(\mu_{ij})$, using a log link with the following linear predictor:

$$\log(\mu_{ij}) = \beta_0 + e_i, \tag{11}$$

where β_0 is the intercept and e_i is the effect of the ith extract, $i = 1, \ldots, 4$. We can fit this model in R and produce a simple analysis of deviance as before, using `glm()` and `anova()`:

```
R> data("progeny", package = "hnp")
R> fit1_p <- glm(y ~ extract, family = poisson, data = progeny)
R> anova(fit1_p, test = "Chisq")
```

```
Analysis of Deviance Table

Model: poisson, link: log

Response: y

Terms added sequentially (first to last)


        Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                       39     534.44
extract  3   444.68        36      89.77 < 2.2e-16 ***
--
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The residual deviance is again much larger than the number of residual degrees of freedom, indicating that the model does not fit the data well. This is confirmed by the half-normal plot with simulated envelope, see Figure 2(a):

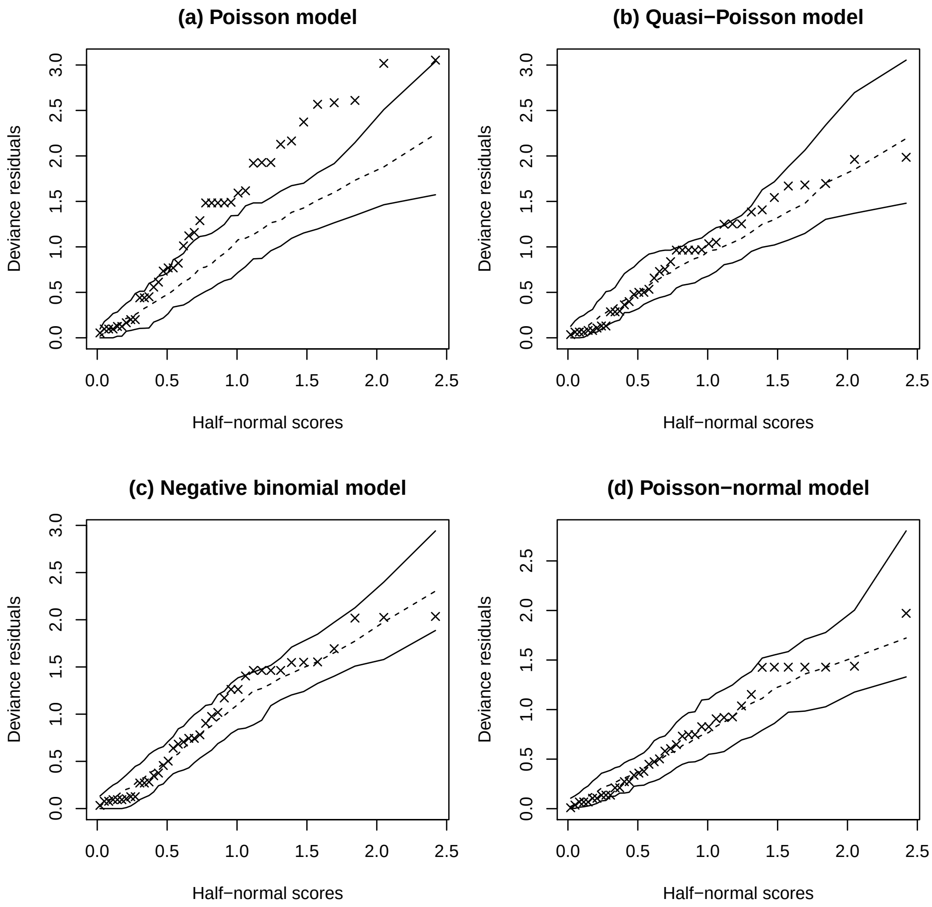

Figure 2: Half-normal plots with simulated envelopes for the deviance residuals from (a) the Poisson model, (b) the quasi-Poisson model, (c) the negative binomial type-II model, and (d) the Poisson-normal model, fitted to the progeny data.

```
R> hnp(fit1_p, xlab = "Half-normal scores", ylab = "Deviance residuals",
+    main = "(a) Poisson model", pch = 4)
```

We now fit overdispersion models, and begin with a constant overdispersion quasi-Poisson model:

```
R> fit2_p <- glm(y ~ extract, family = quasipoisson, data = progeny)
R> summary(fit2_p)$dispersion

[1] 2.365385

R> hnp(fit2_p, xlab = "Half-normal scores", ylab = "Deviance residuals",
+    main = "(b) Quasi-Poisson model", pch = 4)
```

We observe that the estimated dispersion parameter is $\hat{\phi} = 2.37$ and the half-normal plot indicates that this model fit was satisfactory with most of the deviance residuals lying within the simulated envelope, see Figure 2(b). We now make use of package **MASS** to fit the negative binomial type-II model using the same linear predictor (11) and **lme4** to fit the Poisson-normal model with a random effect in the linear predictor:

$$\log(\mu_{ij}) = \beta_0 + e_i + \sigma Z_{ij}, \tag{12}$$

where $Z_{ij} \sim \mathrm{N}(0, \sigma^2)$.

```
R> library("MASS")
R> fit3_p <- glm.nb(y ~ extract, data = progeny)
R> hnp(fit3_p, xlab = "Half-normal scores", ylab = "Deviance residuals",
+    main = "(c) Negative binomial model", pch = 4)
R> library("lme4")
R> x <- factor(seq_len(nrow(progeny)))
R> fit4_p <- glmer(y ~ extract + (1 | x), family = poisson, data = progeny)
R> hnp(fit4_p, xlab = "Half-normal scores", ylab = "Deviance residuals",
+    main = "(d) Poisson-normal model", pch = 4)
```

Both half-normal plots with simulation envelopes show that these are also satisfactory models for this data set, see Figures 2(c) and 2(d), although there is perhaps some suggestion that the Poisson-normal model seems to capture the bulk of the variability better than the others.

4.3. Implementing new model classes

Negative binomial type-I model using package **gamlss**

We now turn to a data set on a tissue-culture experiment using the orange variety *Caipira*. To study the effect of six sugars (maltose, glucose, galactose, lactose, sucrose and glycerol) on the stimulation of somatic embryos from callus cultures, the number of embryos after approximately four weeks was observed. The experiment was set up in a completely randomized block design with five blocks and the six sugars at dose levels of 18, 37, 75, 110 and 150 μM for the first five and 6, 12, 24, 36 and 50 μM for glycerol, see Tomaz *et al.* (1997). The main interest was in the dose-response relationship and the data shows high variability. In their analysis, Tomaz *et al.* (1997) used a quasi-Poisson model. An alternative is the negative binomial type-I model (Jansakul and Hinde 2004), which has the same variance function as the quasi-Poisson, that is, it is also a constant overdispersion model (see (5), with $\delta = 0$). However, as the negative binomial type-I is a fully specified probability model (albeit not in the exponential family) it is possible to obtain standard maximum likelihood parameter estimates.

For sugars lactose and galactose there seems to be a quadratic relationship when we look at the scatter plot, see Figure (4), which justifies the use of the following linear predictor:

$$\log(\mu_{ijk}) = \beta_0 + b_j + \beta_{1_k} d_i + \beta_{2_k} d_i^2, \tag{13}$$

where β_0 is the intercept, b_j is the effect of the jth block, $j = 1, \ldots, 5$, d_i is the ith dose, $i = 1, \ldots, 5$, and β_{1_k} and β_{2_k} are the linear and quadratic dose effects for the kth sugar, $k = 1, \ldots, 6$. Package **gamlss** allows for simple implementation of the negative binomial type-I model using `family = NBII()`[1]:

[1] Note that in package **gamlss** (as of version 4.3-8) families `NBI()` and `NBII()` correspond to the negative binomial type-II and type-I models, respectively, i.e., the types are switched.

```
R> data("orange", package = "hnp")
R> library("gamlss")
R> fit_nbI <- gamlss(embryos ~ block + poly(dose, 2) * sugar,
+    family = NBII(), data = orange)

GAMLSS-RS iteration 1: Global Deviance = 1381.312
GAMLSS-RS iteration 2: Global Deviance = 1289.437
GAMLSS-RS iteration 3: Global Deviance = 1271.898
GAMLSS-RS iteration 4: Global Deviance = 1267.792
GAMLSS-RS iteration 5: Global Deviance = 1267.293
GAMLSS-RS iteration 6: Global Deviance = 1267.23
GAMLSS-RS iteration 7: Global Deviance = 1267.222
GAMLSS-RS iteration 8: Global Deviance = 1267.222
```

The `hnp` function does not handle '`gamlss`' objects with negative binomial type-I family. Attempting to use it yields an error. In order to use it we must pass three helper functions to `hnp`. The first function, passed to argument `diagfun`, must extract the desired model diagnostics. In this case, we will use the z-scores (see Rigby and Stasinopoulos 2005, for further details), which are the default standardized residuals computed from the function `resid` on a '`gamlss`' object:

```
R> d.fun <- function(obj) resid(obj)
```

The second function, passed to argument `simfun`, is used to simulate random samples using the same model matrix and distribution used as when fitting the model to the original data:

```
R> s.fun <- function(n, obj) rNBII(n, obj$mu.fv, obj$sigma.fv)
```

A final function, passed to argument `fitfun`, is used to refit the simulated data using the same model as fitted to the original data:

```
R> f.fun <- function(y.)
+    gamlss(y. ~ block + poly(dose, 2) * sugar, family = NBII(),
+      data = orange)
```

By setting `newclass = TRUE` we can now pass the three helper functions and the data set to the `hnp` function to produce the half-normal plot with simulated envelope:

```
R> hnp(fit_nbI, newclass = TRUE, diagfun = d.fun, simfun = s.fun,
+    fitfun = f.fun, xlab = "Half-normal scores", ylab = "z-scores",
+    main = "", pch = 4, cex = 1, cex.lab = .8, cex.axis = .8)
```

As expected, this model also fits the data well, see Figure 3.

The fitted mean curves are easily obtained using the `predict` function and are displayed with the raw data in Figure 4:

```
R> fit_pred <- gamlss(embryos ~ poly(dose, 2) * sugar, family = NBII(),
+    data = orange)
```

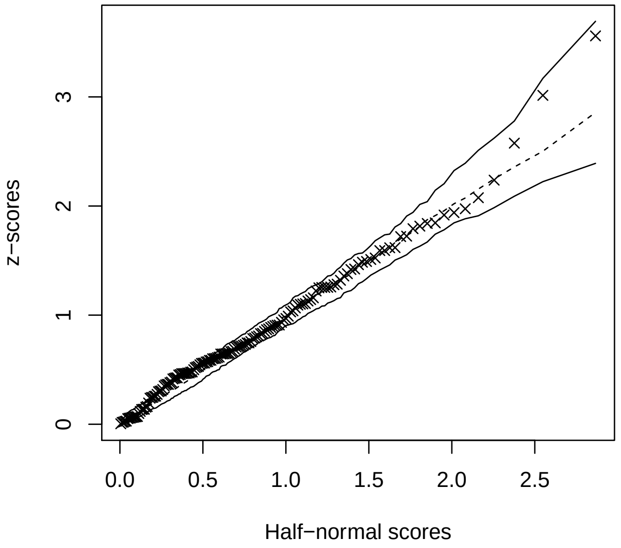

Figure 3: Half-normal plot with simulated envelope for the z-scores from the negative binomial type-I model fitted to the orange data using `gamlss`.

```
R> orange.pred <- rbind(expand.grid(sugar = levels(orange$sugar)[-6],
+    dose = 18:150), expand.grid(sugar = "Glycerol", dose = 6:50))
R> orange.pred$pred <- predict(fit_pred, newdata = orange.pred,
+    type = "response")
R> library("latticeExtra")
R> trellis.par.set(strip.background = list(col = "lightgrey"))
R> xyplot(embryos ~ dose | sugar, scales = list(relation = "free"),
+      layout = c(3, 2), data = orange, col = 1, xlab = "Dose levels",
+      ylab = "Number of embryos") +
+    as.layer(xyplot(pred ~ dose | sugar, type = "l", col = 1,
+      data = orange.pred))
```

Exponential and Weibull models using package ***survival***

In a clinical trial to assess the efficacy of maintenance chemotherapy for acute myelogenous leukemia (AML), after reaching a state of remission through treatment with chemotherapy, 23 patients were randomized into two groups. The first group continued receiving maintenance chemotherapy and the second did not. The observed outcome was the time (in weeks) until relapse and it was right-censored (Miller 1997). The AML data set is available as object `leukemia` in package **survival** (Therneau 2017):

```
R> library("survival")
R> data("leukemia", package = "survival")
```

Two widely used models in survival analysis are the exponential and the Weibull models. They are both easily fitted to data using the function `survreg` from the package **survival**. We proceed by fitting the exponential and Weibull models to the AML data including the treatment factor in the linear predictor:

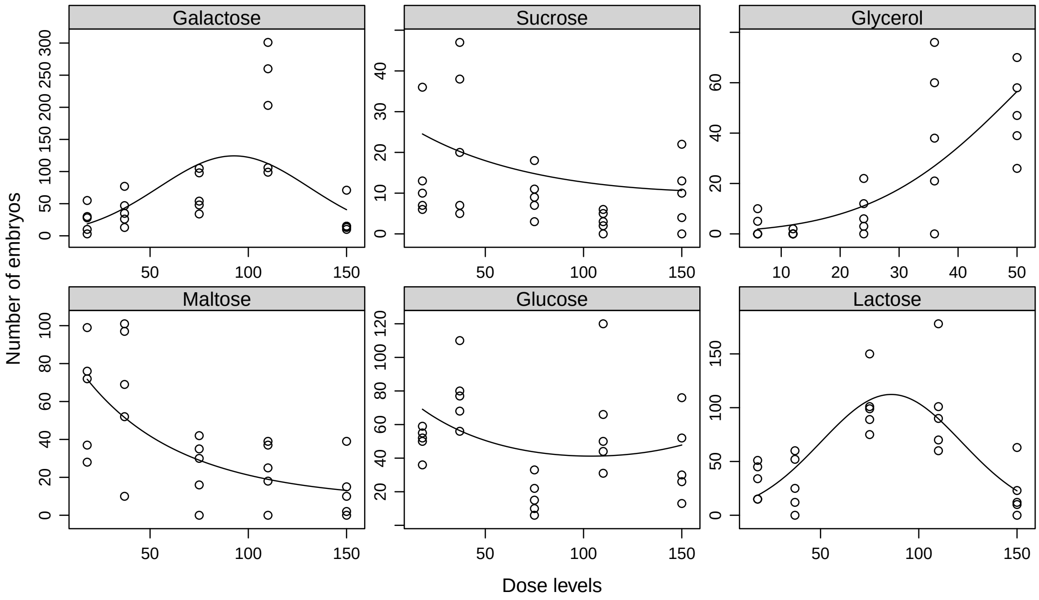

Figure 4: Number of embryos produced by oranges of the *Caipira* variety treated with six sugars at five different concentrations and fitted dose-response curves from the negative binomial type-I model.

```
R> fit_exp <- survreg(Surv(time, status) ~ x - 1, dist = "exponential",
+    data = leukemia)
R> fit_weib <- survreg(Surv(time, status) ~ x - 1, dist = "weibull",
+    data = leukemia)
```

We now must provide the three helper functions for `hnp` in the same way as in the previous section, albeit with slight differences. The `resid` function for '`survreg`' objects can be used to compute the deviance residuals for survival analysis models. The three helper functions may be written as:

```
R> d.fun <- function(obj) resid(obj, type = "deviance")
R> s.fun.exp <- function(n, obj) rexp(n, rate = 1 / predict(obj))
R> s.fun.weib <- function(n, obj)
+    rweibull(n, shape = 1 / obj$scale, scale = predict(obj))
R> f.fun.exp <- function(y.) {
+    leukemia$censoring <- c(sample(leukemia$status[1:11]),
+      sample(leukemia$status[12:23]))
+    return(survreg(Surv(y., censoring) ~ x - 1, dist = "exponential",
+      data = leukemia))
+ }
R> f.fun.weib <- function(y.) {
+    leukemia$censoring <- c(sample(leukemia$status[1:11]),
+      sample(leukemia$status[12:23]))
+    return(survreg(Surv(y., censoring) ~ x - 1, dist = "weibull",
+      data = leukemia))
+ }
```

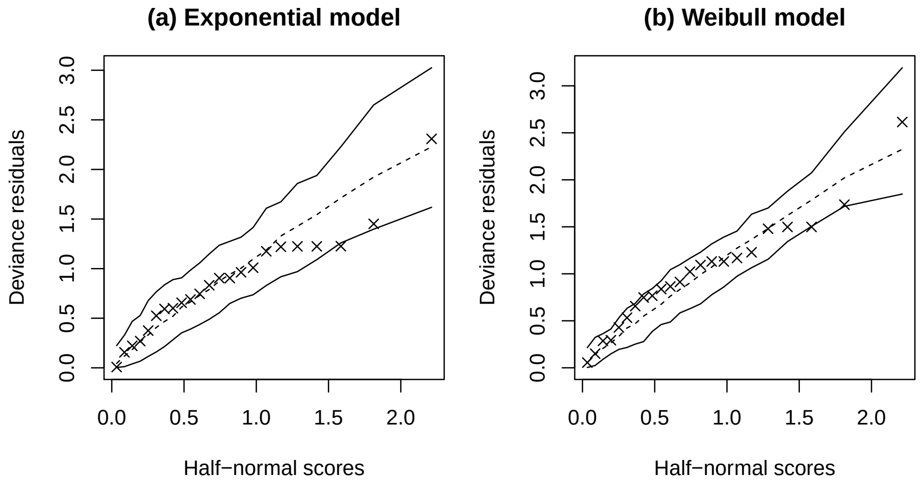

Figure 5: Half-normal plot with simulated envelope for the deviance residuals from (a) the exponential model and (b) the Weibull model fitted to the AML data.

```
R> hnp(fit_exp, newclass = TRUE, diagfun = d.fun, simfun = s.fun.exp,
+    fitfun = f.fun.exp, pch = 4, cex = 1, main = "(a) Exponential model",
+    xlab = "Half-normal scores", ylab = "Deviance residuals")
R> hnp(fit_weib, newclass = TRUE, diagfun = d.fun, simfun = s.fun.weib,
+    fitfun = f.fun.weib, pch = 4, cex = 1, main = "(b) Weibull model",
+    xlab = "Half-normal scores", ylab = "Deviance residuals")
```

We observe that both models seem to fit the data reasonably well, see Figure 5. This is as we might expect as the Weibull scale parameter is not significantly different from 1, which corresponds to the exponential distribution.

Weibull model using function `optim` *to maximize the likelihood*

Now suppose that we want to fit a model that has not yet been implemented in any R function. It is also possible to produce the half-normal plots with simulation envelopes if a fitting function is written for this new type of model. For sake of simplicity, here we show how this can be done by programming an estimation procedure for the Weibull model and using the AML data set. For parametric survival analysis models, the log-likelihood function can be written as

$$l(\boldsymbol{\theta};\mathbf{y}) = \sum_{i=1}^{n} \delta_i \log\{f(y_i;\boldsymbol{\theta})\} + (1-\delta_i)\log\{S(y_i;\boldsymbol{\theta})\},$$

where $\boldsymbol{\theta}$ is the vector of parameters, y_i is the time for the ith individual, $i = 1, \ldots, n$, $f(\cdot)$ is the probability density function, $S(\cdot) = 1 - F(\cdot)$ is the survival function and δ_i is the censoring indicator (for observation i, $\delta_i = 1$ if not censored and $\delta_i = 0$ if censored). The log-likelihood function under non-informative censoring for the Weibull model may be written in R as below, with `param` the vector of parameters, `y` the data vector, `cens` the censoring indicator vector and `X` a generic model matrix:

```
R> loglik.weibull <- function(params, y, cens, X) {
+    shape <- exp(params[1])
+    mu <- exp(X %*% params[-1])
+    llik <- cens * dweibull(y, shape, mu, log = TRUE) +
+      (1 - cens) * pweibull(y, shape, mu, lower.tail = FALSE, log = TRUE)
+    return(- sum(llik))
+  }
```

Parameter estimates are easily obtained through `optim` and are comparable to the estimates obtained through `survreg`:

```
R> weibull.fit <- optim(log(c(1, 60, 24)), loglik.weibull,
+    y = leukemia$time, cens = leukemia$status,
+    X = model.matrix(~ x - 1, leukemia))
R> round(exp(weibull.fit$par), 2)

[1]  1.26 60.90 24.05

R> round(as.numeric(with(fit_weib, c(1 / scale, exp(coefficients)))), 2)

[1]  1.26 60.89 24.04
```

To use the `hnp` function to obtain the half-normal plot with simulated envelope, we must first write a function for fitting the model. To make the later writing of `hnp`'s helper functions easier, we can include the calculation of a vector of residuals in the fitting function. In this case we will use the modified Cox-Snell residuals, defined as

$$\hat{r}_{cs_i} = -\log \hat{S}(y_i) + 1 - \delta_i,$$

where $\hat{S}(\cdot)$ is the estimated survival function and y_i is the observation or censoring time. The following code may be used to implement these functions and fit the model:

```
R> cox.snell <- function(params, y, X, cens) {
+    shape <- params[1]
+    mu <- X %*% params[-1]
+    cs <- - pweibull(y, shape, mu, lower.tail = FALSE, log = TRUE)
+    return(cs + 1 - cens)
+  }
R> fit.model <- function(fmla, cens, init, data) {
+    resp <- with(data, eval(fmla[[2]]))
+    X <- model.matrix(fmla, data)
+    fit <- optim(init, loglik.weibull, y = resp, cens = cens, X = X)
+    result <- list(pars = exp(fit$par), resid = cox.snell(
+      params = exp(fit$par), y = resp, X = X, cens = cens))
+    return(result)
+  }
R> fit_weib.lik <- fit.model(fmla = time ~ x - 1, cens = leukemia$status,
+    init = log(c(1, 60, 24)), data = leukemia)
R> fit_weib.lik
```

```
$pars
[1]  1.264268 60.899983 24.046215

$resid
 [1] 0.08916287 0.14193473 1.14193473 0.21417391 0.29198102 1.37442192
 [7] 0.42584008 0.47859201 1.68213340 0.74012510 4.41810086 0.13730057
[13] 0.13730057 0.24873372 0.24873372 0.41529928 1.59747151 0.94531324
[19] 1.15774844 1.32270784 1.49209119 2.08510851 2.20846458
```

Now we may write the helper functions for `hnp` as

```
R> d.fun <- function(obj) obj$resid
R> s.fun <- function(n, obj) {
+    params <- obj$pars
+    rweibull(n, shape = params[1], scale = rep(params[2:3], each = 15))
+  }
R> f.fun <- function(y.) {
+    censoring <- c(sample(leukemia$status[1:11]),
+      sample(leukemia$status[12:23]))
+    leukemia$new.response <- y.
+    return(fit.model(new.response ~ x - 1, cens = censoring,
+      init = log(fit_weib.lik$pars), data = leukemia))
+  }
```

And produce the half-normal plot:

```
R> hnp(fit_weib.lik, newclass = TRUE, diagfun = d.fun, simfun = s.fun,
+    fitfun = f.fun, main = "", pch = 4, cex = 1, cex.lab = .8,
+    cex.axis = .8, xlab = "Half-normal scores",
+    ylab = "Modified Cox-Snell residuals")
```

As expected, the model fits the data, see Figure 6. As we are using a different type of residuals, the shape of the envelope is different than for the Weibull model fitted using `survreg`. Indeed, for a well-fitting model the Cox-Snell residuals should follow an exponential distribution with unit mean, rather than the normal distribution for the deviance residuals.

5. Discussion

Half-normal plots with simulation envelopes are a useful tool to assess goodness-of-fit for a range of different models, such as classical linear models, generalized linear models, models for overdispersed and zero-inflated data, survival analysis models, as well as mixed models. The key point lies in the simulation procedures which sometimes may be problematic due to model complexity. The **hnp** package allows for implementation of any model for which it is possible to write diagnostic extraction, simulation and fitting codes. This may also be useful for didactic purposes and in simulation studies.

There are other packages that provide a (half-)normal plot with simulated envelope for a few model classes. For example, package **ssym**'s function `envelope` produces a normal plot

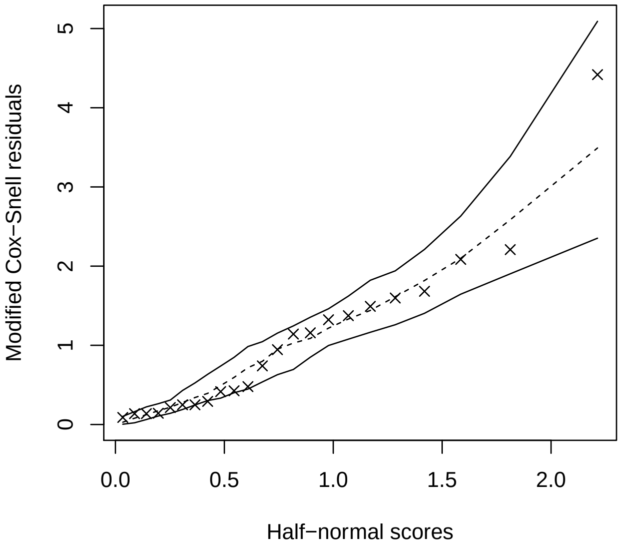

Figure 6: Half-normal plot with simulated envelope for the modified Cox-Snell residuals from the Weibull model fitted to the AML data.

with simulated envelope for semi-parametric log-symmetric regression models (Vanegas and Paula 2015). Package **pgam** also includes an `envelope` function to produce normal plots with simulated envelopes for Poisson-gamma additive models fitted using the roughness penalty approach (Junger and Leon 2012). Package **betareg** includes a `plot` method (argument `which = 5`) that produces a half-normal plot with simulated envelope for residuals of a beta regression model (Cribari-Neto and Zeileis 2010). Also, package **car** provides a function, `qqPlot`, whose method for '`lm`' objects generates a normal plot with simulated envelope for studentized residuals of a linear regression (Fox and Weisberg 2011).

In the `hnp` function, by default the half-normal distribution is used to plot the diagnostic quantities against its expected order statistics (the normal distribution may be used by setting `halfnormal = FALSE`). However there is no reason to do so other than several types of residuals are shown to follow the normal distribution and so we would expect them to form a straight line in the plot. Again the aim in producing a simulation envelope is to reduce subjective bias as to the departure of points from a straight line (when the diagnostic quantity distribution is expected to be normal) and to provide the expected shape of the plot when the distribution is not expected to be normal. Depending upon model complexity the simulation and especially the fitting procedures may be time consuming. Therefore, considerable time may be spent to produce a simulation envelope with the default 99 simulations when one fitting procedure already takes a long time. So, sometimes it may be wiser to use just 19 simulations (`sim = 19`) and form the envelope from the minimum and maximum values obtained in the simulations (`conf = 1`), as proposed by Atkinson (1985), so that there is a chance of approximately 1 in 20 that the observed value of the diagnostic quantity is the most extreme and lies outside of the envelope.

Throughout this paper we have used different types of residuals in the examples, but for generalized linear models mainly deviance residuals, which are shown to have important asymptotic properties and be suitable for diagnostic analyses (McCullagh and Nelder 1989).

For mixed models, the question of which type of residuals should be used for goodness-of-fit assessment is still an active area of research (Nobre and da Motta Singer 2007). However, any type of model diagnostic may be used because they are simply being compared with what we might expect if the model were true, as given by the repeated sequences of "data simulation – model re-fitting – diagnostic extraction". It is important to point out that every time this plot is produced the envelope bands change slightly, hence it sometimes may be useful to produce several half-normal plots and observe how many points lie outside of the bands as well as their position. Of course, not only are we not ruling out the use of other goodness-of-fit assessment techniques, but also we encourage that different tools are used to ensure that the model fits the data well so that no misleading inference is made. The **hnp** package merely provides a, hopefully, useful and flexible tool to help in this assessment.

Acknowledgments

We thank the editors and reviewers for comments that led to the improved structure of the **hnp** package. This research has been funded by "Fundação de Amparo à Pesquisa do Estado de São Paulo" (FAPESP proc. no. 2011/15253-6 and 2014/12903-8) and "Conselho Nacional de Desenvolvimento Científico e Tecnológico" (CNPq proc. no. 304948/2013-6).

References

Atkinson AC (1985). *Plots, Transformations, and Regression: An Introduction to Graphical Methods of Diagnostic Regression Analysis*. Clarendon Press, Oxford.

Bartlett MS (1937). "Properties of Sufficiency and Statistical Tests." *Proceedings of the Royal Society of London A*, **160**(901), 268–282. doi:10.1098/rspa.1937.0109.

Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using **lme4**." *Journal of Statistical Software*, **67**(1), 1–48. doi:10.18637/jss.v067.i01.

Blom G (1958). *Statistical Estimates and Transformed Beta-Variables*. John Wiley & Sons, New York.

Cribari-Neto F, Zeileis A (2010). "Beta Regression in R." *Journal of Statistical Software*, **34**(2), 1–24. doi:10.18637/jss.v034.i02.

Demétrio CGB, Hinde J (1997). "Half-Normal Plots and Overdispersion." *GLIM Newsletter*, **27**, 19–26.

Demétrio CGB, Hinde J, Moral RA (2014). "Models for Overdispersed Data in Entomology." In CP Ferreira, WAC Godoy (eds.), *Ecological Modelling Applied to Entomology*, pp. 219–259. Springer-Verlag.

Doran H, Bates D, Bliese P, Dowling M (2007). "Estimating the Multilevel Rasch Model: With the **lme4** Package." *Journal of Statistical Software*, **20**(2), 1–18. doi:10.18637/jss.v020.i02.

Fox J, Weisberg S (2011). *An R Companion to Applied Regression*. 2nd edition. Sage, Thousand Oaks. URL http://socserv.socsci.mcmaster.ca/jfox/Books/Companion.

Francis B, Green M, Payne C (eds.) (1993). *The GLIM System: Release 4 Manual*. Clarendon Press, Oxford.

Hinde J, Demétrio CGB (1998). "Overdispersion: Models and Estimation." *Computational Statistics & Data Analysis*, **27**(2), 151–170. doi:10.1016/s0167-9473(98)00007-3.

Jansakul N, Hinde J (2004). "Linear Mean-Variance Negative Binomial Models for Analysis of Orange Tissue-Culture Data." *Songklanakarin Journal of Science and Technology*, **26**(5), 683–696. URL http://rdo.psu.ac.th/sjstweb/journal/26-5/09orange-tissue-culture.pdf.

Jørgensen B (2002). "Generalized Linear Models." In AH El-Shaarawi, WW Piegorsch (eds.), *Encyclopedia of Environmetrics*, volume 2, pp. 873–880. John Wiley & Sons, Chichester.

Junger W, Leon AP (2012). ***pgam**: Poisson-Gamma Additive Models*. R package version 0.4.12, URL https://CRAN.R-project.org/package=pgam.

Lesnoff M, Lancelot R (2013). ***aods3**: Analysis of Overdispersed Data Using S3 Methods*. R package version 0.4-1, URL https://CRAN.R-project.org/package=aods3.

McCullagh P, Nelder JA (1989). *Generalized Linear Models*. Chapman and Hall, London.

Miller RG (1997). *Survival Analysis*. John Wiley & Sons, New York.

Moral RA, Hinde J, Demétrio CGB (2017). ***hnp**: Half-Normal Plots with Simulation Envelopes*. R package version 1.2-4, URL https://CRAN.R-project.org/package=hnp.

Nelder JA, Wedderburn RWM (1972). "Generalized Linear Models." *Journal of the Royal Statistical Society A*, **135**(3), 370–384. doi:10.2307/2344614.

Nobre JS, da Motta Singer J (2007). "Residual Analysis for Linear Mixed Models." *Biometrical Journal*, **49**(6), 863–875. doi:10.1002/bimj.200610341.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Ribeiro LP, Vendramin JD, Bicalho KU, Andrade MS, Fernandes JB, Moral RA, Demétrio CGB (2013). "*Annona Mucosa* Jacq. (Annonaceae): A Promising Source of Bioactive Compounds Against *Sitophilus Zeamais* Mots. (Coleoptera: Curculionidae)." *Journal of Stored Products Research*, **55**, 6–14. doi:10.1016/j.jspr.2013.06.001.

Rigby RA, Stasinopoulos DM (2005). "Generalized Additive Models for Location, Scale and Shape." *Journal of the Royal Statistical Society C*, **54**(3), 507–554. doi:10.1111/j.1467-9876.2005.00510.x.

Royston JP (1982). "Algorithm AS 177: Expected Normal Order Statistics (Exact and Approximate)." *Journal of the Royal Statistical Society C*, **31**(2), 161–165. doi:10.2307/2347982.

Shapiro SS, Wilk MB (1965). "An Analysis of Variance Test for Normality (Complete Samples)." *Biometrika*, **52**(3–4), 591–611. doi:10.1093/biomet/52.3-4.591.

Skaug H, Fournier D, Bolker B, Magnusson A, Nielsen A (2014). *Generalized Linear Mixed Models Using **AD Model Builder***. R package version 0.8.1, URL http://glmmadmb.R-Forge.R-project.org/.

Stasinopoulos DM, Rigby RA (2007). "Generalized Additive Models for Location Scale and Shape (GAMLSS) in R." *Journal of Statistical Software*, **23**(7), 1–46. doi:10.18637/jss.v023.i07.

Therneau TM (2017). ***survival**: Survival Analysis*. R package version 2.41-3, URL https://CRAN.R-project.org/package=survival.

Tomaz ML, Mendes BMJ, Filho FAM, Demétrio CGB, Jansakul N, Rodriguez APM (1997). "Somatic Embryogenesis in *Citrus* SPP.: Carbohydrate Stimulation and Histodifferentiation." *In Vitro Cellular & Developmental Biology – Plant*, **37**, 446–452. doi:10.1007/s11627-001-0078-y.

Vanegas LH, Paula GA (2015). ***ssym**: Fitting Semi-Parametric Log-Symmetric Regression Models*. R package version 1.5.4, URL https://CRAN.R-project.org/package=ssym.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York.

Wood SN (2006). *Generalized Additive Models: An Introduction with R*. Chapman and Hall/CRC.

Yee TW (2010). "The **VGAM** Package for Categorical Data Analysis." *Journal of Statistical Software*, **32**(10), 1–34. doi:10.18637/jss.v032.i10.

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.

Affiliation:

Rafael A. Moral, Clarice G. B. Demétrio
Departamento de Ciências Exatas
ESALQ – Universidade de São Paulo
Av. Pádua Dias, 11, Piracicaba – SP, Brazil
E-mail: rafael.moral@usp.br, clarice.demetrio@usp.br

John Hinde
School of Mathematics, Statistics and Applied Mathematics
National University of Ireland, Galway
University Road, Galway, Ireland
E-mail: john.hinde@nuigalway.ie

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
November 2017, Volume 81, Issue 10 *Submitted:* 2015-09-24
doi:10.18637/jss.v081.i10 *Accepted:* 2016-10-18