Generalized Linear Randomized Response Modeling using GLMMRR

by Jean-Paul Fox, Konrad Klotzke and Duco Veen

Abstract Randomized response (RR) designs are used to collect response data about sensitive behaviors (e.g., criminal behavior, sexual desires). The modeling of RR data is more complex since it requires a description of the RR process. For the class of generalized linear mixed models (GLMMs), the RR process can be represented by an adjusted link function, which relates the expected RR to the linear predictor for most common RR designs. The package **GLMMRR** includes modified link functions for four different cumulative distributions (i.e., logistic, cumulative normal, Gumbel, Cauchy) for GLMs and GLMMs, where the package **lme4** facilitates ML and REML estimation. The mixed modeling framework in **GLMMRR** can be used to jointly analyze data collected under different designs (e.g., dual questioning, multilevel, mixed mode, repeated measurements designs, multiple-group designs). Model-fit tests, tools for residual analyses, and plot functions to give support to a profound RR data analysis are added to the well-known features of the GLM and GLMM software (package **lme4**). Data of Höglinger and Jann (2018) and Höglinger, Jann, and Diekmann (2014) are used to illustrate the methodology and software.

Introduction

Response distortion is known to be a potential threat to accurate (survey) results. The methodological literature shows many examples of misreporting of embarrassing or socially undesirable behaviors in surveys (Tourangeau and Yan, 2007; Tourangeau and Smith, 1996). The randomized response technique (RRT) has been developed to encourage respondents to answer truthfully to questions about sensitive behaviors. The RRT is designed to collect responses in an indirect way, where respondents are instructed to answer a sensitive question truthfully only with a certain probability. Therefore, an affirmative response to a sensitive question is masked by a random device at the individual level. The traditional way of direct questioning (DQ) about sensitive topics is known to lead to socially desirable responses or non-cooperation of the respondents. Honest responding is encouraged by RRT by increasing anonymity and confidentiality.

After the introduction of the randomized response (RR) design by Warner (1965), many randomized response (RR) designs have been introduced to improve RRT performance in collecting sensitive information. Warner's design is based on a deck of cards containing cards with the sensitive question and cards with its negation. Since then, different randomizing devices have been introduced to improve the privacy protection of the respondent. Outcomes of rolling dice or spinning a spinner have been used to determine at random one of the possible routings to response for a respondent. Another non-sensitive question has been introduced to make the design easier to implement, where the response to the non-sensitive question is used to steer the response process at random. For online surveys, an online randomizing device, such as a digital dice or spinner, is less reliable since the computer outcomes can be stored. Therefore, techniques have been developed where user interaction is requested to generate a random outcome, which cannot be digitally stored. Currently, there are many ways to randomize the response in an online survey, where the randomizing design properties – the level of privacy protection and the efficiency of the design – are still under the control of the investigator.

The flexibility in RR designs has also improved the utility of the RRT. They can be applied to survey items (Lensvelt-Mulders et al., 2005; van den Hout et al., 2010) but also to scale items for measuring sensitive constructs (Fox, 2005; Fox and Meijer, 2008; Fox et al., 2013). Different RRT implementations have been introduced and applied in many research fields. Together with this expansion of the RRT to different domains, statistical tools have been developed to improve the analysis of RR data. Scheers and Dayton (1988) developed a (logistic) linear regression method for RR data to explore relationships between variables in addition to estimating prevalence rates of the sensitive attribute. Since then, more advanced statistical regression models for RR data have been developed. For instance, randomized item-response theory (RIRT) models for measuring sensitive constructs (Fox, 2012) and mixture RR models to deal with non-cooperating participants (Fox et al., 2013). To further stimulate the use of RRT, software tools are needed to make the developed methods and designs for RR data widely available.

Recently, two packages for RR data have become available in R. The package **rr** of Blair et al. (2015) was developed for univariate power analysis to measure the sensitive item prevalence for four RR designs. It also comprehends a logistic regression function for RR data collected with a single RR

design. The package **RRreg** of Heck and Moshagen (2018) has logistic and linear regression routines, which can include random effects, and tools for power analysis for different RR designs (including those for continuous data). However, both packages are restricted to single-group RR designs and do not allow joint analysis of different randomization schemes across items and/or participants.

Multiple-group RR designs are particularly interesting in online surveys, where different RR designs and different random devices can be easily varied. For instance, the level of protection can be moderated by varying the design parameters, and RR designs can be varied across survey items to improve truth-telling. For validation studies, multiple-group RR designs are relevant to make comparisons between groups questioned with different RR designs (Höglinger and Jann, 2018; Höglinger et al., 2016). For a meta-analysis, a joint analysis of the available data can improve the measurement of a sensitive prevalence. Furthermore, differences in prevalence estimates can be examined across studies. More recently, there has been an increased interest in measuring the efficacy of the RRT to improve the honest disclosure of sensitive information (John et al., 2018). This is usually done through validation studies where a DQ group serves as the baseline, and it is investigated whether higher prevalence estimates are obtained with a single-group RRT. The huge amount of literature on the topic and mixed results about the benefits of RRT has led to skeptical positions toward RRT (Wolter and Preisendörfer, 2013). However, the RR design parameters (the random-device properties), as well as the type of RR design, can influence the response behavior, and care must be taken in designing the RRT to reduce the level of misreporting. It is not likely that a fixed privacy level encourages truth-telling in a uniform way across participants. The performance of RRT will vary across subgroups and participants, and different RR designs can vary in performance across subgroups depending on the level of sensitivity of the question.

The R package **GLMMRR** extends existing implementations by providing generalized regression tools for multiple-group RR designs. It builds in a natural way on common regression methods of the system package **stats** to make them suitable for RR data. The R package **GLMMRR** gives support to *generalized linear (mixed effects) regression modeling* of binary RR data and extends the popular generalized linear regression routines in R to handle RR outcomes by including RR-specific link functions for a wide variety of RR designs. The **GLMMRR** provides several important contributions for the joint regression analysis of binary RR data:

1. Different link functions (logit, probit, complementary log-log, cauchit) are supported to optimize the link between the linear predictor and the response and to avoid restrictions on the range of the (expected) response. The complementary log-log and cauchit link functions approach the asymptotes of zero and one asymmetrically.

2. Each link function is modified to an RR-link function to make it suitable for various RR designs (Warner, Unrelated question, Forced response, Kuk, Crosswise, Triangular), where two design parameters specify the properties of the RR design.

3. The package supports a joint regression analysis of RR data sampled with different RR designs and different design parameters. Each response observation is either indirectly observed with an RR design with unique design parameters or directly observed without an RR design.

4. For randomized item-response data, RR-design specific (weighted) prevalence rates and confidence intervals can be computed for each item. Trait levels can be estimated, where the items serve as indicators of a sensitive trait.

5. Extensive residual tools are available for the evaluation of the fit of the model. This includes Pearson, deviance and response residuals, goodness-of-fit statistics (Pearson's and deviance chi-square statistic, Hosmer-Lemeshow statistic), and the Akaike's Information Criterion (AIC) and the Bayesian Information Criterion (BIC).

6. The `RRglm` is used to fit a GLM, and an object of class `"RRglm"` is created, which extends the class `"glm"` with RR data. The `glm` methods are applicable to an object of class `"RRglm"`. The function `summary` or `print` can be used to obtain a summary of the results. Other useful features can be extracted using the well-known generic functions, such as `coefficients`, `effects`, `fitted.values`, and `anova`.

7. The `RRglmer` for fitting a GLMM returns an object of class `"RRglmerMod"` which extends the class `"glmerMod"` and `"merMod"` with RR data. The methods for an object of class `"glmerMod"` are applicable for an object of class `"RRglmerMod"` (the methods can be found in the **lme4** documentation).

This paper is organized as follows. In the next section, the RR-link functions are introduced, and the GLM and GLMM modeling framework for RR data is introduced. A general introduction is given where the responses are distributed according to a distribution of the exponential family, and the RR-link function is used to link the linear predictor to the expected (randomized) response. In Section 2.3, the ML and REML estimation method is discussed for the GLM(M) with RR-link functions. Then, the RIRT modeling approach for measuring sensitive constructs is discussed.

Section 2.5 discusses different methods to evaluate the fit of the model. Different residuals are defined (Pearson, deviance, response) for a GLM. For a GLMM, a conditional residual is defined as the difference between the observation and the conditional expected value (given the random effect estimate). Goodness-of-fit statistics are introduced by grouping the Pearson and deviance residuals. For a fixed number of groups, the Hosmer-Lemeshow (H-L), the Pearson, and the deviance statistic are approximately chi-square distributed and can be used to evaluate the global fit of a GLM. In Section 2.6, the two main functions of the package are discussed for fitting GLMs and GLMMs with RR data. In Section 2.7, an illustration of the package is shown by analyzing data from the validation study of Höglinger and Jann (2018), where differences in outcomes between RR designs are evaluated. Furthermore, the prevalence of student misconduct is analyzed with a randomized item-response theory model where students were assigned to different treatment groups, each assigned to a specific RR design. A joint modeling approach is carried out to examine differences between treatment groups. Then, in Section 2.8, the conclusions are given.

General randomized-response probability model

When collecting RR data, each observation is randomized before it is observed. The RR process makes it impossible to directly relate an observed RR to the sensitive question. It masks the answers of respondents. To ease the notation, a distinction is made between respondents with index i and items with index k. A further specification is possible where participants and/or items are grouped, which will be shown in the real data examples in Section 2.7. Thus, respondent's i prevalence to the sensitive question k, denoted as $\tilde{\pi}_{ik}$, cannot be directly measured. Instead, the RR probability, denoted as π_{ik}, is measured.

Fox et al. (2018) showed that for the common RR designs for binary data, the RR probability for the observed RR data can be related to the prevalence of the sensitive question through a linear equation. This linear relationship is given by,

$$\pi_{ik} = c_{ik} + d_{ik}\tilde{\pi}_{ik}. \tag{1}$$

The RR parameters c and d determine the type of RR design. A variety of RR designs can be represented by Equation (1): Warner's design (Warner, 1965), Unrelated Question (UQM) design, Forced response (FR) design (Boruch, 1971), Kuk's design (Kuk, 1990), Triangular design (Yu et al., 2008), and the Crosswise (CW) design (Yu et al., 2008). The RR parameters c and d are retrieved from the RR design parameters. For instance, for Warner's design, a positive response is given to the sensitive question with probability p_1 or a negative response to its negation with probability $1 - p_1$,

$$\pi_{ik} = p_1\tilde{\pi}_{ik} + (1 - p_1)(1 - \tilde{\pi}_{ik})$$
$$= \underbrace{(1 - p_1)}_{c_{ik}} + \underbrace{(2p_1 - 1)}_{d_{ik}}\tilde{\pi}_{ik}.$$

The parameters c_{ik} and d_{ik} describe the random response process, which are allowed to vary across questions and respondents such that the data-collection design is defined for every single response. Fox et al. (2018) give an overview in which the RR-design parameters are represented as RR parameters for various RRTs.

Although the RR designs work in different ways, a general RR model can be defined, which relates the RR data to the prevalence of the sensitive question. This general RR model is extended to a GLM and GLMM by linking the prevalence $\tilde{\pi}_{ik}$ to a linear predictor. The **GLMMRR** functions require the RR design parameters p_1 and p_2. The function `getRRparameters` provides the RR parameters c and d given the type of RR design and corresponding parameter values. This is illustrated in our applications in Section 2.7, where the RR designs UQM, FR, and CW are used, and the DQ design serves as a baseline.

Modified link functions for GLMs and GLMMs

The object is to define the natural link function, which relates the expected RR to a linear predictor for the prevalence $\tilde{\pi}_{ik}$. The RR parameters c_{ik} and d_{ik} are integrated into the link function to relate the expected RR observation to this linear predictor without explicitly parameterizing the prevalence. In this approach, the natural link functions for Bernoulli distributed observations are modified to account for the RR parameters. Simply by adjusting the natural link functions, the class of GLMs and GLMMs is extended to (binary) RR data.

Let Y_{ik} denote the (binary) RR observation of respondent i to item k. For binary data, the expected response is equal to the RR probability, π_{ik}. The expected response is related to a linear

predictor denoted as η_{ik}, which is a linear combination of predictor variables for respondent i related to the response to item k. For binary RR data, the expected RR observation is equal to the RR probability, π_{ik}, and the relationship with the η_{ik} can be represented by

$$
\begin{aligned}
E\left(Y_{ik} \mid \eta_{ik}\right) &= \pi_{ik} \\
&= c_{ik} + d_{ik}\,\tilde{\pi}_{ik} \\
&= c_{ik} + d_{ik}g^{-1}\left(\eta_{ik}\right) \\
&= c_{ik} + d_{ik}g^{-1}\left(\mathbf{x}_{ik}^{t}\boldsymbol{\beta} + \mathbf{z}_{ik}^{t}\boldsymbol{b}_{i}\right),
\end{aligned}
\tag{2}
$$

where the explanatory variables $\mathbf{x}_{ik}$ have fixed effects (i.e., common across respondents) and the $\mathbf{z}_{ik}^{t}$ have random effects (i.e., vary across respondents). The random effect $\boldsymbol{b}_{i}$ is assumed to have a multivariate normal distribution with mean zero and variance $\boldsymbol{\Sigma_b}$. For $c_{ik} = 0$ and $d_{ik} = 1$, the $g^{-1}()$ is the mean function for (non-randomized) Bernoulli distributed observations. It follows that the natural link function for Bernoulli distributed data, $g()$, needs to be modified to link the expected RR to the linear predictor for the prevalence $\tilde{\pi}_{ik}$,

$$
g\left(\frac{\pi_{ik} - c_{ik}}{d_{ik}}\right) = \eta_{ik}.
\tag{3}
$$

Consider the Bernoulli distributed RR observation with success probability π_{ik}, for which four different cumulative distribution functions can be applied (i.e., logistic, probit, Gumbel, Cauchy). Then, the distribution of the RR is given by

$$
Y_{ik} \sim \mathcal{B}\left(\pi_{ik}\right)
\tag{4}
$$

$$
\pi_{ik} = c_{ik} + d_{ik}g^{-1}\left(\eta_{ik}\right)
\tag{5}
$$

$$
= \begin{cases}
c_{ik} + d_{ik}\frac{\exp(\eta_{ik})}{1+\exp(\eta_{ik})} & \text{Logistic} \\
c_{ik} + d_{ik}\Phi\left(\eta_{ik}\right) & \text{Probit} \\
c_{ik} + d_{ik}\left(1 - \exp\left(1 - \exp\left(\eta_{ik}\right)\right)\right) & \text{Gumbel} \\
c_{ik} + d_{ik}\left(\arctan\left(\eta_{ik}\right)/\pi + \frac{1}{2}\right) & \text{Cauchy.}
\end{cases}
\tag{6}
$$

Next, four different link functions are defined by modifying the linear predictor's relation with the prevalence, such that it relates to the expected RR observation. The possible (modified) link functions are given by,

$$
\eta_{ik} = g\left((\pi_{ik} - c_{ik})/d_{ik}\right)
\tag{7}
$$

$$
= \begin{cases}
\ln\left(\frac{\pi_{ik}-c_{ik}}{c_{ik}+d_{ik}-\pi_{ik}}\right) & \text{Logit Link} \\
\Phi^{-1}\left(\frac{\pi_{ik}-c_{ik}}{d_{ik}}\right) & \text{Probit Link} \\
\ln\left(-\ln\left(\frac{c_{ik}+d_{ik}-\pi_{ik}}{d_{ik}}\right)\right) & \text{Complementary log-log Link} \\
\tan\left(\pi\left(\frac{\pi_{ik}-c_{ik}}{d_{ik}}\right)\right) & \text{Cauchit Link.}
\end{cases}
\tag{8}
$$

The considered four different link functions are modified versions of the common link functions as defined in, for example, McCullagh and Nelder (1989) and Tutz (2011). When $c_{ik} = 0$ and $d_{ik} = 1$, the common link functions for directly observed responses are given.

Exponential family distributions

The modified (natural) link functions can be employed for (Bernoulli) exponential family distributed RR data. Assume the observed RR data is distributed according to a distribution of the exponential family; that is,

$$
p\left(y_{ik} \mid \theta_{ik}, \phi_{ik}\right) = \exp\left(\frac{y_{ik}\theta_{ik} - A\left(\theta_{ik}\right)}{\phi} + C\left(y_{ik}, \phi\right)\right).
\tag{9}
$$

Then, the log-likelihood of the parameter θ_{ik} and ϕ is expressed as

$$
l\left(\theta_{ik}, \phi; \mathbf{y}\right) = \log p\left(\mathbf{y} \mid \theta_{ik}, \phi\right) = \sum_{i,k}\frac{y_{ik}\theta_{ik} - A\left(\theta_{ik}\right)}{\phi} + C\left(y_{ik}, \phi\right).
\tag{10}
$$

The θ_{ik} is the canonical parameter and depends via a linear predictor on explanatory variables. The dispersion parameter is usually unknown and used to model the variance of the response data. The functions $A(.)$ and $C(.)$ are known and determined by the specified distribution of the family. Then,

for Bernoulli distributed RR observations the natural form of the parameters is given by

$$\theta_{ik} = \log\left(\frac{c_{ik} + d_{ik}\tilde{\pi}_{ik}}{1 - (c_{ik} + d_{ik}\tilde{\pi}_{ik})}\right) \tag{11}$$

$$A(\theta_{ik}) = \log\left(1 + \exp\left(\theta_{ik}\right)\right) \tag{12}$$

$$\phi = 1. \tag{13}$$

Thus, the general properties of the exponential family distributions can be used to make inferences about the parameters. For instance, the maximum likelihood estimate for the population prevalence $\tilde{\pi}$ can be derived. The log-likelihood has a unique maximum at $\hat{\theta}$ which is the solution to

$$\sum_{i,k} y_{ik} = A'(\hat{\theta}) = \sum_{i,k}\left(1 + \exp(-\hat{\theta}_{ik})\right)^{-1}$$

$$\sum_{i,k} y_{ik} = \sum_{i,k}(c_{ik} + d_{ik}\hat{\pi})$$

$$\hat{\pi} = \left(\sum_{i,k} y_{ik} - \sum_{i,k} c_{ik}\right) / \sum_{i,k} d_{ik} = (\overline{y} - \overline{c})/\overline{d}. \tag{14}$$

This maximum likelihood (ML) estimate for the prevalence is referred to as a weighted (ML) estimate, where the weights are defined by the RR parameters.

Furthermore, the inverse of the function $A'(\theta)$ represents the natural link function. Thus, the modified natural link function for Bernoulli distributed RR data can be derived using this property of the exponential family distribution:

$$\pi_{ik} = A'\left(\theta_{ik}\right)$$

$$= \left[1 + \exp\left(-\log\left(\frac{c_{ik} + d_{ik}g^{-1}(\eta_{ik})}{1 - (c_{ik} + d_{ik}g^{-1}(\eta_{ik}))}\right)\right)\right]^{-1}$$

$$= c_{ik} + d_{ik}g^{-1}(\eta_{ik})$$

$$\eta_{ik} = g\left((\pi_{ik} - c_{ik})/d_{ik}\right),$$

where $g^{-1}(.)$ is the cumulative distribution function for (non-randomized) Bernoulli distributed data as defined in Equation (6). The corresponding link functions are defined in Equation (8).

ML and REML estimation

It can be shown that the maximum likelihood (ML) equations for the GLM for RR data resemble the general form of the GLM ML-equations (Fox et al., 2018, Appendix B). The only difference is that the conditional expected RR observation includes the RR parameters, and a modified link function is needed to link the conditional expected response to the linear term. The GLM parameters are usually estimated by ML methods using the iterative weighted least squares (IWLS) algorithm or Fisher scoring algorithm. The `glm` function implemented in R is a very flexible implementation of the general GLM framework (Chambers and Hastie, 1992). The package **GLMMRR** provides an expanded version of this function (`RRglm`), which includes modified link functions to fit GLMs on RR data. Given ML estimates, (maximum) likelihood theory can be used to obtain likelihood ratio tests, Wald and score tests.

The likelihood equations for the fixed effect parameters of the GLMM for RR data have the same structure as those for the GLMM (Fox et al., 2018, Appendix B). For the fixed effects, the GLM methodology can be used, since the fixed effects are not included in the random effect distribution. Depending on the dimension of the random effect parameter, numerical approximations are required to approximate the integrals to estimate the variance components and random effects. The numerical methods available in **GLMMRR** build on those available in the package **lme4**. Different numerical methods have been proposed; Laplace approximation and adaptive Gaussian quadrature are both implemented. Laplace approximation is usually fast and the default. The approximation improves when the cluster sizes increase. In Gaussian quadrature, a number of quadrature points need to be chosen, and the approximation is improved by increasing the number of quadrature points. Adaptive Gaussian quadrature usually fails when the dimension of the random effects is larger than two. It is also possible to compute restricted ML estimates (REML), which is also implemented. The `control` argument can be used in the function call to `RRglmer` to set the control parameters, which includes the optimizer to be used.

Randomized item-response theory modeling

An important class of item response theory (IRT) models belong to the GLMM class (Rijmen et al., 2003). These IRT models have a linear component for the transformed expected values of a binary response variable, which contains a random component(s) representing the latent variable(s). In the most common form, the linear term has a random component representing a random person effect and fixed components representing item effects. In Section 2.2.1, it is shown that the GLMM is extended to RR data using modified link functions. Therefore, IRT models belonging to the GLMM class can also be generalized to randomized item-response theory (RIRT) models.

The randomized item-response observations are clustered by persons and items, with c_{ik} and d_{ik} representing the RR parameters. The latent variable for person i is denoted by ϑ_i and the effect of item k by β_k. Then, the RIRT model – the Rasch model for RR data – can be represented as

$$
\begin{aligned}
\pi_{ik} &= c_{ik} + d_{ik}g^{-1}\left(\vartheta_i + \beta_k\right) \qquad (15) \\
\vartheta_i &\sim N(0, \sigma^2),
\end{aligned}
$$

where g^{-1} is usually the logistic or cumulative normal distribution function. The item effects have a positive sign and should be interpreted as a easiness parameters.

The RIRT model can be fitted in R using the function `RRglmer` from our package **GLMMRR**. This is similar to fitting an IRT model using the function `glmer` from **lme4** (De Boeck et al., 2011), except that a modified link function is required for the RIRT. The data needs to be in a long format. Then, each data case has a response (`response`), a person identifier (`person`), an item identifier (`item`), the type of RR design (`RRmodel`), and RR design parameters (`RRp1,RRp2`). The RR model and parameters are allowed to vary across data cases.

To fit the RIRT model in Equation (15), a linear component is defined from the factor variables: `-1+item+(1|person)`. The `-1` restricts the general mean to zero, the `item` represents the item parameters, and the `(1|person)` represents the latent variable (random effect). This linear component is set equal to the outcome `response` in the model formula. To fit a logistic RIRT, the call to `RRglmer` includes a modified logistic link function (`RRlink.logit`):

```
RRglmer(response ~ -1 + item + (1|person), link = "RRlink.logit",
    RRmodel = RRmodel, p1=RRp1, p2=RRp2, data=data)
```

where the `RRmodel`, `p1`, and `p2` arguments define the RR design for every single response. The probit RIRT model can be fitted using the link function `RRlink.probit`. The RIRT models can be extended by including (1) item-covariate models (e.g., linear-logistic test model), (2) person-covariate models (e.g., multilevel IRT), and (3) person-by-item covariate models. Covariates with fixed or random effects can be included in the linear term.

Model fit and diagnostics

The likelihood ratio (LR) test can be used to compare nested models. The nested model is a restriction of a more general model by restricting one or more parameters most often to zero. The log-likelihood ratio (multiplied by minus two) is asymptotically chi-square distributed with the degrees of freedom equal to the difference in the number of free parameters. The LR test can be performed with the `anova` function to compare two nested GLM(M)s for RR data. Note that the LR test cannot be applied to test a hypothesis on the boundary of the parameter space. Since then the LR statistic is no longer chi-square distributed. ML estimation is preferred when comparing models that only differ in their fixed part.

To compare (non)-nested models, the usual information criteria can be used. The AIC and the BIC are both computed for GLM and GLMM, with the function `RRglm` and `RRglmer`, respectively, and reported in their output. The `anova` function also reports the AIC and BIC.

To evaluate the significance of fixed effects, a z-statistic is reported, which is the ratio of the parameter estimate and the estimated standard error. The z-statistic is asymptotically equivalent to the LR test. P-values are reported in the output of `RRglm` and `RRglmer` with the assumption that the z-statistic is asymptotically normally distributed.

Residuals

The error term in GLM(M)s represents a Bernoulli random error term, and the errors are assumed to be independently distributed. There are different types of residuals and different types of residual sum

of squares to examine the fit of the model. The estimated residuals in the GLM and GLMM are based on the fitted RR probabilities, $\hat{\pi}_{ik}$. The fitted prevalence can be computed as $\hat{\tilde{\pi}}_{ik} = (\hat{\pi}_{ik} - c_{ik})/d_{ik}$, according to Equation (1).

GLM The residuals are computed for each observation y_{ik}, but for notational convenience, the index k is dropped, and the index i refers to a single observation y_i. The Pearson and deviance residual is often computed, and both are used in a goodness-of-fit statistic. The Pearson residual is defined as the standardized difference of the response and its expected value,

$$r_p(y_i, \hat{\pi}_i) = \frac{y_i - \hat{\pi}_i}{\sqrt{\hat{\pi}_i(1 - \hat{\pi}_i)}}.$$

The covariate patterns can be used to group the residuals in a natural way. Each unique covariate pattern defines a cluster j, where the number of response observations in cluster j is given by n_j, the observed proportion $\overline{y}$, and a unique estimated probability denoted as $\hat{\pi}_j$. Then, the Pearson residual for cluster j is defined as

$$r_p(\overline{y}_j, \hat{\pi}_j) = \frac{\overline{y}_j - \hat{\pi}_j}{\sqrt{\hat{\pi}_j(1 - \hat{\pi}_j)/n_j}}.$$

The clustering of observations can also be based on the predicted outcome (Hosmer and Lemeshow, 1980). The observations are sorted according to their fitted probabilities, and this sorted vector is divided into J clusters of equal size. The corresponding clustered Pearson residuals are referred to as H-L residuals. The Pearson residuals and clustered Pearson and H-L residuals can be extracted from a fitted object of class `RRglm` with the following commands, respectively:

```
residuals(object, type="pearson")
residuals(object, type="pearson.grouped")
residuals(object, type="hosmer-lemeshow", ngroups=10)
```

For the H-L residuals, the `ngroups` argument defines the desired number of clusters, and the default is 10. The sum of squared grouped residuals defines a goodness-of-fit statistic, which is given by

$$X^2 = \sum_{j=1}^{J} r^2(\overline{y}_j, \hat{\pi}_j) = \sum_{j=1}^{J} \frac{n_j \left(\overline{y}_j - \hat{\pi}_j\right)^2}{\hat{\pi}_j(1 - \hat{\pi}_j)}.$$

Depending on the type of clustering, the statistic is the Pearson goodness-of-fit statistic, X_p^2, or the H-L goodness-of-fit statistic, X_{HL}^2. The Pearson statistic is asymptotically chi-square distributed for a fixed number of groups with degrees of freedom $J - (q + 1)$, and q the number of independent covariates. When the predictor variables are continuous, the X_p^2 test cannot be used. For semi-continuous observations, it is possible to find a clustering with a sufficient number of observations in each cluster. The X_{HL}^2 statistic is assumed to be chi-square distributed with $J - 2$ degrees of freedom (Hosmer and Lemeshow, 1980). The X_p^2 and X_{HL}^2 statistics can be computed from a fitted object of class `RRglm` using the function `RRglmGOF`:

```
RRglmGOF(object, doPearson = TRUE, doHlemeshow = TRUE,
    hlemeshowGroups = 10, rm.na = TRUE)
```

The default number of clusters for the X_{HL}^2 statistic is ten, and data cases with missing observations are excluded. The deviance residual is defined as

$$r_D(\bar{y}_i, \hat{\pi}_i) = sign(\bar{y}_i - \hat{\pi}_i) \sqrt{2n_i \left(\bar{y}_i log \left(\frac{\bar{y}_i}{\hat{\pi}_i} \right) + (1 - \bar{y}_i) log \left(\frac{1 - \bar{y}_i}{1 - \hat{\pi}_i} \right) \right)}, \quad (16)$$

where $sign(\bar{y}_i - \hat{\pi}_i) = 1$ when $\bar{y}_i \geq \hat{\pi}_i$ and -1 when $\bar{y}_i < \hat{\pi}_i$. The grouped deviance residual is defined for cluster i with $n_i > 1$ and denoted as $r_d(\bar{y}_i, \hat{\pi}_i)$. For clustered observations, the sum of squared deviance residuals is considered to be a goodness-of-fit statistic, denoted as X_d^2, which is asymptotically chi-square distributed with $J - (q + 1)$ degrees of freedom. The deviance residuals, the grouped deviance residuals, and the deviance goodness-of-fit statistic can be extracted from a fitted `RRglm` object with the commands, respectively:

```
residuals(out, type = "deviance")
residuals(out, type = "deviance.grouped")
RRglmGOF(object, doDeviance = TRUE, rm.na = TRUE)
```

In the output of `RRglmGOF`, the goodness-of-fit test(s) are reported, the p-value(s), the degrees of freedom, and the number of groups. For the H-L test, the results are also given for each cluster. The

deviance statistic is equivalent to the LR statistic for testing the fitted model against the saturated model, which has a deviance of zero. The Pearson goodness-of-fit statistic is a score test statistic, also testing the fitted model against the saturated model. Thus, the Pearson and the deviance statistic are the score test and LR test for GLMs, respectively.

GLMM Different residuals can be considered for GLMMs. The response residual conditional on the random effect is defined as the difference between the observation and the conditional expected value:

$$
\begin{aligned}
r_c(y_{ik}, \hat{\pi}_{ik}) &= y_{ik} - E(y_{ik} \mid \mathbf{x}_{ik}, \mathbf{z}_{ik}, \mathbf{b}_i) \\
&= y_{ik} - \pi \left(\mathbf{x}_{ik}^t \hat{\boldsymbol{\beta}} + \mathbf{z}_{ik}^t \hat{\mathbf{b}}_i \right).
\end{aligned}
\tag{17}
$$

It can be extracted from the `residual` function by providing the argument `type="response"`. The unconditional response can be computed by integrating out the random effects to obtain the difference between the observation and the marginal mean. This residual is computed with the argument `type="unconditional.response"`.

The conditional and unconditional response residuals can be standardized by dividing them by their standard deviation, which leads to Pearson residuals:

```
residuals(object, type = "pearson")
residuals(object, type = "unconditional.pearson")
```

Finally, as a result of sustained compatibility with **lme4**, the usage of residuals aimed at `"merMod"` objects, such as working residuals and conditional deviance residuals, is not limited. Pearson residuals can also be scaled by any given user-specified weights.

Software

The package **GLMMRR** contains two main functions, `RRglm` and `RRglmer`, for fitting a GLM and a GLMM given RR data, respectively. Both functions include the four link functions (logit, probit, cloglog, cauchit) for the different RR designs. The function `RRglm` makes a call to the function `glm` with the appropriate link function to fit a GLM for (binary) RR data. In the same way, the function `RRglmer` makes a call to the function `glmer` to fit a GLMM for (binary) RR data. The fit of both models, GLM and GLMM, is arranged by the computational routines of `glm` and `glmer`. Their general control parameters, the model and data-checking options, the type of optimizer, number of iterations can be specified in the `RRglm` and `RRglmer` functions. Thus, the numerical optimization algorithm and its specification can be defined in a similar way as in the call to functions `glm` and `glmer`.

The function `RRglm` and `RRglmer` creates an object of class `"RRglm"` and `"RRglmer"`, respectively. The package's summary, print and plot function, can be used to get estimation results from an object of each class. Data needs to be defined in long format, where an RR specification is needed for every single case, including the type of RR model and the design parameters. The package allows for different RR models and different design parameters across data cases. Together with a binary RR (outcome) variable and possible predictors, a GLM can be fitted. When also including a (factor) cluster variable, implying a correlation among clustered observations, a GLMM can be fitted. The complete functionality of the package can be accessed by making further input specifications.

Input

- `RRglm`
  The general call to the function is: `RRglm(formula,link,item,RRmodel,p1,p2,data,na.action = "na.omit",...)`

  - formula: a two-sided linear formula object describing the model to be fitted, with the response on the left of a $\sim$ operator and the terms, separated by $+$ operators, on the right.
  - link: a GLM RRlink function for binary outcomes. Must be a function name; `RRlink.logit`, `RRlink.probit`, `RRlink.cloglog`, and `RRlink.cauchit`.
  - item: optional item identifier to obtain prevalence estimates per level of item.
  - RRmodel: the RR model per data case. Available options: `DQ`, `Warner`, `Forced`, `UQM`, `Crosswise`, `Triangular`, and `Kuk`.

- p1: the RR parameter p_1, defined per data case ($0 \leq p_1 \leq 1$).
- p2: the RR parameter p_2, defined per data case ($0 \leq p_2 \leq 1$).
- data: a data frame containing the variables named in formula as well as the RR model and parameters. If the required information cannot be found in the data frame, or if no data frame is given, then the variables are taken from the environment from which RRglm is called.
- na.action: a function that indicates what should happen when the data contain NAs. The default action (na.omit, as given by getOption("na.action"))) strips any observations with any missing values in any variables.

- `RRglmer`
  The general call to the function is `RRglmer(formula,item,link,RRmodel,p1,p2,data,control = glmerControl(),na.action = "na.omit",...)`

  - formula: a two-sided linear formula object describing both the fixed and random effects part of the model, with the response on the left of a $\sim$ operator and the terms, separated by + operators, on the right. Random-effect terms are distinguished by vertical bars ("|") separating expressions for design matrices from grouping factors.
  - item : optional item identifier to obtain prevalence estimates per level of item.
  - link: a GLM RRlink function for binary outcomes. Must be a function name; `RRlink.logit`, `RRlink.probit`, `RRlink.cloglog`, and `RRlink.cauchit`.
  - RRmodel: the RR model per data case. Available options: `DQ`, `Warner`, `Forced`, `UQM`, `Crosswise`, `Triangular`, and `Kuk`.
  - p1: the RR parameter p_1, defined per data case ($0 \leq p_1 \leq 1$).
  - p2: the RR parameter p_2, defined per data case ($0 \leq p_2 \leq 1$).
  - data: a data frame containing the variables named in formula as well as the RR model and parameters. If the required information cannot be found in the data frame, or if no data frame is given, then the variables are taken from the environment from which RRglmer is called.
  - na.action: a function that indicates what should happen when the data contain NAs. The default action (na.omit, as given by getOption("na.action"))) strips any observations with any missing values in any variables.
  - control: a list (of correct class, resulting from `lmerControl()` or `glmerControl()`, respectively) containing control parameters, including the nonlinear optimizer to be used and parameters to be passed through to the nonlinear optimizer; see the `lmerControl documentation` for details.

Output

- `RRglm` An object of class `RRglm`, which extends the class `glm` with RR data. The object of class `RRglm` contains the regular GLM output and the following components:

  - Item: the item levels for each data case – prevalence rates are computed per level of item.
  - RRc: the RR-parameter c for each data case (Equation (1)).
  - RRd: the RR-parameter d for each data case (Equation (1)).
  - RRmodel: the RR model for each data case.
  - RRp1: the RR design parameter p_1.
  - RRp2: the RR design parameter p_2.

- `RRglmer` An object of class `"RRglmerMod"`, which extends the class `"glmerMod"` with RR data. Many methods are available for the general class `"merMod"` to which `"glmerMod"` and also the class `"RRglmerMod"` belongs. The methods for the `"merMod"`-class can be found in the documentation of **lme4** and they are applicable to an object from class *"RRglmerMod"*.

Applications

An RR validation study

Höglinger and Jann (2018) conducted an online experiment to validate different RRTs on the platform Amazon Mechanical Turk (https://boris.unibe.ch/81516). Participants were asked to play one of the two dice games, after which they were asked if they played honestly using randomly one of four RRTs. In the roll-a-six game, the participant rolled a digital die by clicking a button and was asked if the first roll resulted in a six. In the prediction game, the participant was asked to think of a number, roll the digital die, and was asked if the outcome corresponded to the memorized prediction. The participants were asked if they won in the dice game, and since both games relied on self-reports about the dice-roll outcomes, cheating was easily possible. However, for the roll-a-six game, the virtual dice outcomes were registered, and it was evaluated if participants illegitimately claimed a $2 bonus payment. This enabled classifying the participants as `cheater` or `honest` (non-cheater) players. In the prediction game, individual cheating was not detectable. However, it was expected that around one-sixth of all predictions were correct since the dice outcomes were random. A systematic deviation from this percentage was attributed to cheating.

To disguise the true purpose of the study, the survey was posted as a survey on "Mood and Personality" and included a range of questions, for instance, questions on the big five personality trades. Participants were randomly appointed to one of four questioning methods (DQ, CW, FR, UQ) and were asked four sensitive questions. Besides the question about honest playing in the dice game, questions were asked about shoplifting, tax evasion, and voting. Although the answers to these last three questions could not be validated, the prevalence estimates of those who were identified as cheaters and non-cheaters in the dice games can be compared across RRTs.

Höglinger and Jann (2018) compared prevalence estimates of cheating in both games to those computed from the answers to the question of whether they played honestly using the four RR techniques. They concluded that two RR techniques (FR, UQ) performed similarly to DQ and did not reduce the level of misreporting. The level of underreporting was reduced by CW. However, CW also increased the level of overreporting, and the corresponding prevalence estimates were substantially higher than the true prevalence estimates of cheating.

The **GLMMRR** is used to do a joint regression analysis of the RR data of the four questioning techniques. In a joint analysis, the levels of under- and overreporting across RRTs can be directly compared through an interaction analysis of identified cheaters and the RRTs. The joint analysis is needed to quantify the different levels of underreporting across RRTs, and to identify who was misreporting. It is also examined if any background variables explain differences in misreporting across RRTs and sensitive questions.

The experiment had a two-by-three-by-five factorial design. Factor one represented the type of dice game (prediction game, roll-a-six game). Factor two was the type of RRT to ask the sensitive questions (DQ, CW, FR, UQ). The third factor was included to examine whether the implementations of the random devices produced the expected outcome distributions. This third factor is integrated into the current study by defining RRTs with different RR design parameters. Höglinger and Jann (2018) discuss the RRTs (factor 2), the random devices (factor 3), and the corresponding RR design parameters. Their supplement *Documentation and codebook of the survey* was used to prepare the raw data for this study for analysis in R, which includes the specification of the design parameters for the RRTs. Our supplementary R script 'ASQ-MTurk data.R' comprises the code for preparing the data MTURK. The package **GLMMRR** contains the prepared data object MTURK, which is constructed from the raw data using the code of the supplementary R script.

The true prevalence estimates of cheating are estimated for the roll-a-six game (`dicegame=2`) for the different RRTs. The estimates are based on the discrepancy between the actual dice outcome and the participant's response to whether the first roll was a six, which was asked through direct questioning. The true prevalence estimates are reported for each of the groups assigned to one of the RRTs.

```
R> package(GLMMRR)
R> data("MTURK", package="GLMMRR")
R> by(MTURK$cheaterdc[MTURK$dicegame==2],
+   MTURK$RRmodel[MTURK$dicegame==2],mean,na.rm=TRUE)

MTURK$RRmodel[MTURK$dicegame == 2]: DQ
[1] 0.04450262
--------------------------------------------------------------------------------
MTURK$RRmodel[MTURK$dicegame == 2]: Crosswise
[1] 0.06032787
```

```
--------------------------------------------------------------------------------
MTURK$RRmodel[MTURK$dicegame == 2]: UQM
[1] 0.05004812
--------------------------------------------------------------------------------
MTURK$RRmodel[MTURK$dicegame == 2]: Forced
[1] 0.05188067
```

Around 5% of the participants cheated in the roll-a-six game, and the estimates are comparable across RRT groups. It is examined if the prevalence estimates can also be recovered from the participant's responses to the question if they honestly reported whether a six was rolled (item `honest dice game reporting`, where the response is coded as honest=0 and dishonest=1). This question was asked to all participants, but different RRTs were used to obtain the response. The object is to validate the RRTs by comparing the prevalence estimate of cheating with the true prevalence estimate. The prevalence estimates are computed using the `RRglm` function where the `item` is the sensitive question if they are reported honestly. The design parameters are stored in the data variables `RRp1` and `RRp2`, which are also reported in the output. It can be seen that, for instance, the CW design has two sets of parameters, where some participants were questioned with the CW method and design parameters .16 and 0, and others with .20 and 0.

```
R> MTURK_cheating1 <- MTURK[which(MTURK$dicegame==2 &
+    MTURK$Question=="cheating dice game"),]
R> rolla6 <- RRglm(RR_response ~ 1, item = Question,
+    link = "RRlink.logit", RRmodel = RRmodel,
+    p1=RRp1,p2=RRp2,data = MTURK_cheating1)
R> summary(rolla6)

### GLMMRR - Binary Randomized Response Data ###
Generalized linear fixed-effects model

Family:                        binomial
Link function:                 RRlogit


---------------------------------------------------------
Item:                        cheating dice game
Model(s):                DQ (1.00 | 0.00)
                         Crosswise (0.16 | 0.00) (0.20 | 0.00)
                         UQM (0.78 | 0.49) (0.78 | 0.52)
                         Forced (0.75 | 0.67)

## Estimated Population Prevalence (weighted per RR model)
   RRmodel estimate.weighted se.weighted    n
 Crosswise        0.143393    0.0204156 1142
        DQ        0.039370    0.0099632  381
    Forced       -0.019361    0.0172690  769
       UQM        0.053496    0.0165870  778
```

For the question *honest dice game reporting*, the estimated population proportion of dishonest reporters (ML estimate with standard errors) is reported for each RR design. Each RRT estimate is the (weighted) average across different design parameters. It follows that the prevalence estimate of the CW method overestimates the true value of 6% of detected cheaters who were CW questioned. Furthermore, the DQ and UQ perform approximately the same. The estimate is even negative for the FR technique. This can occur when some participants did not follow the RR instructions and/or when the random device distribution deviates from the expected distribution. This could also be an underlying problem of the CW method. A total of 33.6% admitted that they do not know exactly the birthday of their parents. This could bias their response to the unrelated question, which stated if their father/mother's birthday was in January or February (with an expected probability of 15.9%), or between the 1st and the 6th of the month (with an expected probability of 19.7%). The overestimation of the CW method could be caused by honest reporters who incorrectly answered `No` to the unrelated question about the birthday of one of their parents.

Misreporting is investigated further by computing the reduction in misreporting for each RRT for detected cheaters (dishonest reporters) and non-cheaters (honest reporters). Therefore, a dummy coded variable is defined for the cheaters (`cheaterdc=1`) and for each of the RRTs (DQ, CW, UQ, FR). Logistic regression analysis is performed using the `RRglm` function for the RR data of item `honest dice game reporting` (variable `RR_response`) in the roll-a-six game conditional on the RRT and cheater identifiers.

```
R> rolla6A <- RRglm(RR_response ~ 1 + cheaterdc + CW + UQ + FR +
+   cheaterdc*CW+cheaterdc*UQ, item = Question,
+   link = "RRlink.logit", RRmodel = RRmodel, p1=RRp1, p2=RRp2,
+   data = MTURK_cheating1, na.action = "na.omit")
R> summary(rolla6A)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -4.8807     0.6043  -8.077 6.63e-16 ***
cheaterdc     5.8302     0.8087   7.209 5.63e-13 ***
CW            2.8283     0.6375   4.437 9.14e-06 ***
UQ            1.3453     0.8399   1.602   0.1092
FR           -1.3582     0.6941  -1.957   0.0504 .
cheaterdc:CW -3.6246     0.9143  -3.964 7.36e-05 ***
cheaterdc:UQ -2.0967     1.0798  -1.942   0.0522 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3988.2  on 3069  degrees of freedom
Residual deviance: 2631.5  on 3063  degrees of freedom
  (5 observations deleted due to missingness)
AIC: 2645.5

Number of Fisher Scoring iterations: 7
```

The reference level is the DQ technique (factor `RRmodel`) for those who were identified as honest reporters (factor `cheaterdc`) responding to being dishonest in the roll-a-six game. For DQ, the odds for dishonest reporting for non-cheaters is around zero ($\exp(-4.88) = 0.008$), and for cheaters around 2.58. Of the DQ participants, identified as cheaters, the probability of admitting to being dishonest about whether a six was rolled is around 72%. Participants might have guessed that this type of misreporting could be easily detected, which led to a non-substantial level of misreporting under DQ. The odds for dishonest reporting for non-cheaters in the UQ and FR condition are also close to zero. The UQ and FR effects are also not significantly different from zero. For the non-cheaters, the response technique, DQ, UQ, or FR, did not influence their response. However, in the CW condition, the odds of dishonest reporting for non-cheaters is significant and around .13, with a probability of 11.3% of dishonest reporting while being identified to be honest reporters. This led to the overestimation of the true prevalence rate by CW.

The opposite occurred for the cheaters in the CW and UQ condition. It was expected that the cheaters would be honest about their dishonesty in the roll-a-six game without the risk of disclosure in the CW and UQ condition. Instead, for both techniques, the admitted level of misreporting by cheaters decreased. For cheaters in the CW and UQ condition, the probability of being dishonest is around 53.8% and 54.9%, respectively, which is much less than the 72% under DQ. Höglinger and Jann (2018) argued that some cheaters might have misused the RRT to answer untruthfully without risk of detection who would have felt compelled to answer truthfully in DQ. It turned out that it was not possible to estimate an interaction effect for cheaters in the FR condition.

The three other sensitive questions (voting, shoplifting, tax evasion) do not have the problem that participants might be suspicious of being disclosed for their dishonesty as in the roll-a-six game. However, it is expected that the prevalence rates will be lower for the DQ group. Their responses are not masked, and those participants are tended to underreport the questioned behavior in comparison to those not under risk of disclosure through the RR questioning techniques. Differences in prevalence rates across sensitive items (factor `Question`) and question techniques (factor `RRmodel`) are explored through a logistic regression using the `RRglm` function. (factor `Question`)

```
R> MTURK_cheating2 <- MTURK[which(MTURK$dicegame==2 &
+   MTURK$Question!="cheating dice game"),]
R> rolla6B <- RRglm(RR_response ~ 1 + RRmodel + Question +
+   cheaterdc*RRmodel, item = Question, link = "RRlink.logit",
+   RRmodel = RRmodel,p1=RRp1,p2=RRp2,data = MTURK_cheating2)

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)             -0.87793    0.08118 -10.815  < 2e-16 ***
RRmodelCrosswise         0.26976    0.09415   2.865  0.00417 **
```

```
RRmodelUQM                      0.39283    0.09279   4.233  2.3e-05 ***
RRmodelForced                   0.11554    0.09783   1.181  0.23762
Questionshoplifting             0.61299    0.07040   8.708  < 2e-16 ***
Questiontax evasion            -1.01421    0.09100 -11.145  < 2e-16 ***
cheaterdc                       0.23951    0.31801   0.753  0.45135
RRmodelCrosswise:cheaterdc     -0.04895    0.40461  -0.121  0.90371
RRmodelUQM:cheaterdc           -0.30170    0.42096  -0.717  0.47355
RRmodelForced:cheaterdc         0.32900    0.42022   0.783  0.43367
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 12705  on 9207  degrees of freedom
Residual deviance: 11797  on 9198  degrees of freedom
  (21 observations deleted due to missingness)
AIC: 11817

Number of Fisher Scoring iterations: 4
```

The intercept corresponds to non-cheaters in the DQ condition responding to the item non-voting. For DQ, the odds for non-voting of the non-cheaters is around .41 ($\exp(-0.88) = 0.41$), and for cheaters, around 0.53. However, the cheaters did not respond significantly differently from the non-cheaters as they did in reporting about being dishonest in the dice game. The prevalence rates of those questioned with privacy protection (CW, UQ, FR) are higher – the CW and UQ rates are significantly higher – than those questioned directly. The corresponding odds ratios are .54, 62, and .47 for CW, UQ, and FR, respectively. The prevalence rates for shoplifting are significantly higher and for tax evasion significantly lower than for non-voting. It is apparent that cheaters do not report significantly differently under a privacy-protected response technique in comparison to the non-cheaters since the interaction effects are approximately zero and non-significant. This occurred when cheaters were asked about their dishonesty in the roll-a-six game. This was probably provoked by the knowledge that their dishonesty could be detected under DQ. For the questions about non-voting, shoplifting, and tax evasion, it was known that this was not possible, so cheaters were not/less inclined to misreport in the non-DQ condition.

The weighted prevalence rate (averaged over results from the same questioning technique with different design parameters) for each item and questioning technique are reported in the output. It follows that the prevalence rates are lower for DQ than for the other RRTs, but the differences are small. Note that the reported significance in rates between CW and DQ and between UQ and DQ were averaged across the three items.

```
### GLMMRR - Binary Randomized Response Data ###
Generalized linear fixed-effects model

Family:                       binomial
Link function:                RRlogit

----------------------------------------------------------
Item:                      non voting
Model(s):             DQ (1.00 | 0.00)
                      Crosswise (0.16 | 0.00) (0.20 | 0.00)
                      UQM (0.78 | 0.49) (0.78 | 0.52)
                      Forced (0.75 | 0.67)

## Estimated Population Prevalence (weighted per RR model)
   RRmodel estimate.weighted se.weighted    n
 Crosswise          0.38215     0.022718 1142
        DQ          0.30607     0.023673  379
    Forced          0.33333     0.023720  768
       UQM          0.33889     0.022338  776


----------------------------------------------------------
Item:                      shoplifting
Model(s):             DQ (1.00 | 0.00)
                      Crosswise (0.16 | 0.00) (0.20 | 0.00)
                      UQM (0.78 | 0.49) (0.78 | 0.52)
```

```
                    Forced (0.75 | 0.67)

## Estimated Population Prevalence (weighted per RR model)
   RRmodel estimate.weighted se.weighted     n
 Crosswise          0.46339     0.022914 1145
        DQ          0.44357     0.025452  381
    Forced          0.47479     0.024016  769
       UQM          0.56174     0.022936  778


-----------------------------------------------------------
Item:                        tax evasion
Model(s):                 DQ (1.00 | 0.00)
                          Crosswise (0.16 | 0.00) (0.20 | 0.00)
                          UQM (0.78 | 0.49) (0.78 | 0.52)
                          Forced (0.75 | 0.67)

## Estimated Population Prevalence (weighted per RR model)
   RRmodel estimate.weighted se.weighted     n
 Crosswise          0.18850     0.021098 1143
        DQ          0.11549     0.016374  381
    Forced          0.13057     0.021182  771
       UQM          0.19407     0.020301  775
```

The inclusion of the interaction effect between cheating status and RRT can be tested through a model comparison. The model is fitted without the interaction term (object `rolla6C`), and the `anova` function is used to produce a deviance table for the fitted objects. It follows that the interaction term does not lead to a significant model improvement.

```
R> anova(rolla6C, rolla6B,test="Chisq")
Analysis of Deviance Table

Model 1: RR_response ~ 1 + RRmodel + Question
Model 2: RR_response ~ 1 + RRmodel + Question + cheaterdc * RRmodel
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      9202      11802
2      9198      11797  4   5.3509   0.2532
```

Residual analysis

The fit of the GLM with factor variables `RRT` and `Question` (object `rolla6C`) is evaluated with a residual analysis. The function `RRglmGOF` is used to compute overall goodness-of-fit tests. The Pearson and deviance chi-square statistics are computed given a grouping of the observations based on the predictor variables. The H-L statistic is computed for ten groups of (approximately) equal size. There is evidence for a lack-of-fit when the statistic values are large.

```
R> RRglmGOF(RRglmOutput = rolla6C, doPearson = TRUE, doDeviance = TRUE,
+    doHlemeshow = TRUE)

GLMMRR - Binary Randomized Response Data

Goodness-of-Fit Testing
Response variable:                RR_response
Predictor(s):                       RRmodel Question
Entries dataset:                9208
-----------------------------------------------------------
Summary:


                 Statistic P.value df Groups
Pearson          12.95      0.0438  6  12
Deviance         13.01      0.0429  6  12
Hosmer-Lemeshow  8.82       0.3578  8  10


-----------------------------------------------------------
```

It follows that the Pearson and deviance goodness-of-fit statistics show a lack of fit, whereas the H-L statistic does not show a lack of fit. To further investigate the fit, the H-L statistic is computed for

the same model but with different link functions. A different link function can improve the fit of the model and reduce the effects of a misspecified linear predictor. The estimated statistic values are given for the probit link function (see supplementary R-code for the complete analysis), which shows only a slight improvement in fit. The decrease in AIC is small and around .62.

```
R> rolla6E <- RRglm(RR_response ~ 1 + RRmodel + Question,
+   item = Question, link = "RRlink.probit",RRmodel = RRmodel,
+   p1=RRp1,p2=RRp2,data = MTURK_cheating2)
R> RRglmGOF(RRglmOutput = rolla6E, doPearson = TRUE, doDeviance = TRUE,
+   doHlemeshow = TRUE) ## improved fit over logit link


-----------------------------------------------------------
Summary:


               Statistic P.value df Groups
Pearson          12.35    0.0545  6  12
Deviance         12.40    0.0537  6  12
Hosmer-Lemeshow   8.69    0.3689  8  10


-----------------------------------------------------------
```

Finally, the fitted probabilities are plotted against the estimated Pearson residuals for each RRT (R-code below) – it is also possible to plot the object of class `"RRglm"`, `plot(rolla6C,which = 3,type = "pearson")`. In Figure 1, the Pearson residuals (filled circles) under the CW method are large for the zero response observations. Typical for CW, high fitted probabilities (prevalence rates) correspond to zero and to one response. There are three items, which leads to three different fitted probabilities for DQ. CW was used with two different sets of design parameters, which led to more than three fitted probabilities. The differences between prevalence rates and Pearson residuals for different design parameters are much larger for CW than for UQ and FR. For UQ and FR, the estimated Pearson residuals and fitted probabilities hardly differ across design parameters. It follows that the CW method is particularly sensitive to deviations from the design parameters. For an observed prevalence rate above .50, increasing the design parameter for CW will decrease the prevalence estimate. A possibility is that the design parameter for CW was incorrect and too low, which led to the overestimation of the prevalence rate for CW in the roll-a-six game. The residuals are also relatively large for low fitted probabilities, which is typical for DQ when the true prevalence rate is small. For reasons of brevity, we have omitted a further improvement of the model and to explain individual differences in prevalence by including individual predictor variables (e.g., gender, education, Big Five personality traits).

```
R> set <- names(rolla6C$linear.predictors)
R> dataset <- MTURK_cheating2[set,]
R> plot(rolla6C$fitted.values,residuals(rolla6C, type = "pearson"),
+   cex=.8,bty="l",xlim=c(.1,.8),ylim=c(-3,3),xlab="Fitted",
+   ylab="Residual (Pearson)")
R> points(rolla6C$fitted.values[which(dataset$DQ==1)],
+   residuals(rolla6C, type = "pearson")[which(dataset$DQ==1)],
+   xlim=c(0,1),ylim=c(-1,1),pch=15,col="black")
R> points(rolla6C$fitted.values[which(dataset$CW==1)],
+   residuals(rolla6C, type = "pearson")[which(dataset$CW==1)],
+   pch=16,col="green")
R> points(rolla6C$fitted.values[which(dataset$UQ==1)],
+   residuals(rolla6C, type = "pearson")[which(dataset$UQ==1)],
+   pch=17,col="red")
R> points(rolla6C$fitted.values[which(dataset$FR==1)],
+   residuals(rolla6C, type = "pearson")[which(dataset$FR==1)],
+   pch=18,col="blue")
R> legend(.6,3,c("DQ","CW","FR","UQ"),col=c("black","green",
+   "blue","red"),pch = c(15,16,18,17), bg = "gray95",cex=.7)
```

Extended item response modeling: Prevalence of student misconduct

Höglinger et al. (2014) performed an online survey to estimate the prevalence of various forms of student misconduct (e.g., plagiarizing, cheating in exams). RRT was used since students might be reluctant to reveal this kind of information. Four different types of RRTs were used (DQ, FR, CW,
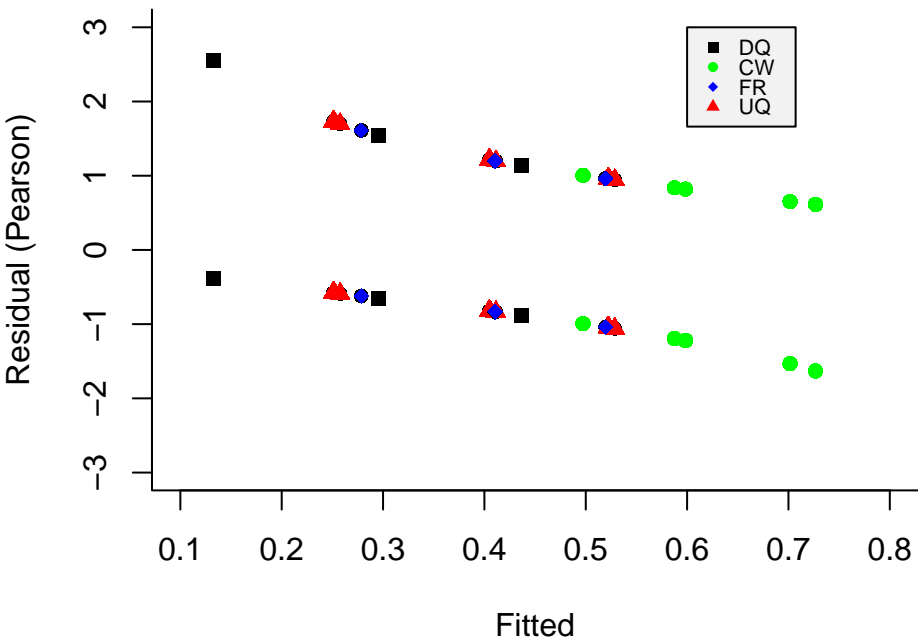
Figure 1: The roll-a-six game: for each RRT fitted success probabilities against Pearson residuals.

UQ). The RRTs were tailored to be implemented online which led to two implementations of FR (pick-a-numer, random wheel) and of CW (unrelated question, random wheel). In total, there were six different RRTs to which participants were randomly assigned: DQ, two implementations of FR, UQ, and two implementations of CW. For all RRT implementations, the level of protection was also varied, which led to different design probabilities. In total, six different sets of parameters for UQ, two for FR, and eight for CW. The package **GLMMRR** was used to do a joint regression analysis of the randomized item-response data, (1) to measure differences between RRTs, and (2) to examine differences in prevalence rates across conditions and students.

No.	Item
1	copied from other students during an exam (copied)
2	used illicit crib notes in an exam (crib notes)
3	used prescription drugs to enhance your performance (drugs)
4	handed in someone else's work without citing (plagiarism)
5	had someone else write a large part of a submitted paper (someone else's work)

Table 1: Five items for measuring prevalence of student misconduct.

Five sensitive items about student misconduct were surveyed using the six different RRTs with different design parameters. In Table 1, the five items are given about respondents' own misconduct during exams and submitting a paper. The items were assumed to measure student misconduct. An RIRT model is used to measure each student's level of misconduct given RR data while taking into account that students were assigned to different RRT conditions. A multiple-group (normal ogive) RIRT model is fitted, where the groups represent the RRT conditions. The function `RRglmer` of **GLMMRR** is used with factor variable `Question` and factor variable `expcond` representing the items and experimental conditions (RRTs), respectively, for the randomized responses (`RR_response`). The input `item` equals factor variable `Question` to obtain the (weighted) prevalence estimates for each item.

The latent variable is represented by the student identifier `id`, where each student responded to the five items. This latent variable represents a student's propensity to misconduct on an exam measured by the five items in Table 1. This (continuous) personality trait is measured with the

RIRT model with data collected under different RRT designs.

```
R> out.re <- RRglmer(RR_response ~ 1 + Question + expcond + (1|id),
+    item=Question, link = "RRlink.probit", RRmodel = RRmodel,
+    p1=p1,p2=p2,data = ETHBE, control=glmerControl(
+    optimizer="bobyqa", optCtrl = list(maxfun = 200000)))
R> summary(out.re)


     AIC      BIC   logLik deviance df.resid
 21015.5  21103.2 -10496.7  20993.5    21394


Random effects:
 Groups Name        Variance Std.Dev.
 id     (Intercept) 0.4342   0.659
Number of obs: 21405, groups:  id, 4281


Fixed effects:
                                      Estimate Std. Error z value Pr(>|z|)
(Intercept)                           -1.20625    0.05633 -21.416  < 2e-16 ***
Questioncrib notes                    -0.34520    0.04582  -7.534 4.93e-14 ***
Questiondrugs                         -0.92016    0.05612 -16.395  < 2e-16 ***
Questionhanded in plagiarism          -0.83103    0.05369 -15.478  < 2e-16 ***
Questionhanded in someone else's work -1.06092    0.05982 -17.736  < 2e-16 ***
expcondFR pick-a-number                0.94880    0.06524  14.543  < 2e-16 ***
expcondCM pick-a-number                0.53036    0.08064   6.577 4.80e-11 ***
expcondFR random wheel                 0.99941    0.06520  15.329  < 2e-16 ***
expcondUQ Benford                      0.39748    0.07245   5.487 4.10e-08 ***
expcondCM unrelated question           0.67867    0.08017   8.465  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The levels of factor variable Question represent the item difficulties, where the intercept represents the first item (copied work). The levels of factor variable expcond represent the mean level of each RRT condition, and the intercept is the level of the DQ group. The latent variable variance in student misconduct is around 0.43, which shows that conditional on the item and experimental condition differences, around 30% ($= .43/(1 + .43)$) of the variance can be attributed to individual differences. The latent variable estimates represent the levels of student misconduct.

It is expected that students underreport their behavior. With an RRT individual responses are masked, and students are expected to answer more truthfully. Although students were randomly assigned to RRT conditions, the prevalence rates appear to be different across RRTs. The package **multcomp** is used to test individual null hypotheses representing differences between RRTs. Linear combinations of the experimental condition parameters are defined and tested simultaneously. Four hypotheses are evaluated to test the difference in prevalence rates (1) between the two FR implementations (pick a number, random wheel), (2) between the two CW implementations (pick-a-number, unrelated question), (3) between FR and CW, (4) between FR and UQ. A matrix K is defined which represents the contrasts of interest and the function glht is used to perform the general linear hypothesis testing.

```
R> K <- matrix(c(c(0, 0, 0, 0, 0, 1, 0, -1, 0 , 0),
+            c(0, 0, 0, 0, 0, 0, 1, 0, 0 ,-1),
+            c(0, 0, 0, 0, 0, 1, -1, 1, 0 ,-1),
+            c(0, 0, 0, 0, 0, 0, 0, 1, -1 ,0)), nrow=4,ncol=10,byrow=T)
R> t <- glht(out.re, linfct = K)
R> summary(t)


        Simultaneous Tests for General Linear Hypotheses


Linear Hypotheses:
       Estimate Std. Error z value Pr(>|z|)
1 == 0 -0.05061    0.06176  -0.819    0.870
2 == 0 -0.14831    0.09125  -1.625    0.343
3 == 0  0.73917    0.11108   6.654   <0.001 ***
4 == 0  0.60193    0.06963   8.645   <0.001 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Adjusted p values reported -- single-step method)
```

It follows that the different implementations of FR do not lead to different prevalence rates. Both FR implementations provide a similar level of privacy protection. Furthermore, both CW implementations have similar prevalence rates. For FR and CW, differences between random devices do not lead to different prevalence estimates. The difference is significant between FR and CW, where FR produces higher rates than CW. FR also produced higher rates than UQ. It is remarkable that in this study, students reported much higher rates with FR than CW, whereas in the RR validation study, it was the other way around. It is not clear why with FR, higher rates were obtained in the student misconduct study than with CW in comparison to the roll-a-six validation study. However, in the roll-a-six game, the prevalence rates with DQ also differed not much with those from FR.

The prevalence rates for the five items are estimated for each RRT, averaging across results from different design parameters. In the output, an overview is given of the RRTs which were used to administer each item. For each RRT, the design parameters are given which were used. For instance, for item `copied`: DQ has one set of design parameters, UQ six sets of design parameters, FR two sets of design parameters, and CW eight sets of design parameters. Note that different design parameters were used for different implementations of a random online device (Höglinger et al., 2014).

```
### GLMMRR - Binary Randomized Response Data ###
Generalized linear mixed-effects model

Family:                     binomial
Link function:              RRprobit


-----------------------------------------------------------
Item:                         copied
Model(s):             DQ (1.00 | 0.00)
                      UQM (0.70 | 0.49) (0.70 | 0.50) (0.70 | 0.52)
                          (0.78 | 0.49) (0.78 | 0.50) (0.78 | 0.52)
                      Forced (0.67 | 0.17) (0.75 | 0.17)
                      Crosswise (0.17 | 0.00) (0.20 | 0.00) (0.23 | 0.00)
(0.25 | 0.00) (0.26 | 0.00) (0.30 | 0.00) (0.75 | 0.00) (0.83 | 0.00)

## Estimated Population Prevalence (weighted per RR model)
   RRmodel estimate.weighted se.weighted     n
 Crosswise          0.28679    0.047932 1427
        DQ          0.21806    0.015389  720
    Forced          0.41265    0.017797 1417
       UQM          0.17351    0.022187  717


-----------------------------------------------------------
Item:                         crib notes
Model(s):             DQ (1.00 | 0.00)
                      UQM (0.70 | 0.49) (0.70 | 0.50) (0.70 | 0.52)
                          (0.78 | 0.49) (0.78 | 0.50) (0.78 | 0.52)
                      Forced (0.67 | 0.17) (0.75 | 0.17)
                      Crosswise (0.17 | 0.00) (0.20 | 0.00) (0.23 | 0.00)
(0.25 | 0.00) (0.26 | 0.00) (0.30 | 0.00) (0.75 | 0.00) (0.83 | 0.00)

## Estimated Population Prevalence (weighted per RR model)
   RRmodel estimate.weighted se.weighted     n
 Crosswise          0.17570    0.044884 1427
        DQ          0.10278    0.011317  720
    Forced          0.32396    0.016822 1417
       UQM          0.14438    0.021528  717


-----------------------------------------------------------
Item:                         drugs
Model(s):             DQ (1.00 | 0.00)
                      UQM (0.70 | 0.49) (0.70 | 0.50) (0.70 | 0.52)
                          (0.78 | 0.49) (0.78 | 0.50) (0.78 | 0.52)
                      Forced (0.67 | 0.17) (0.75 | 0.17)
                      Crosswise (0.17 | 0.00) (0.20 | 0.00) (0.23 | 0.00)
```

```
(0.25 | 0.00) (0.26 | 0.00) (0.30 | 0.00) (0.75 | 0.00) (0.83 | 0.00)

## Estimated Population Prevalence (weighted per RR model)
   RRmodel estimate.weighted se.weighted    n
 Crosswise          0.095850   0.0475537 1427
       DQ           0.029167   0.0062712  720
    Forced          0.162119   0.0138931 1417
       UQM          0.045145   0.0187692  717


-----------------------------------------------------------
Item:                         handed in plagiarism
Model(s):             DQ (1.00 | 0.00)
                      UQM (0.70 | 0.49) (0.70 | 0.50) (0.70 | 0.52)
                          (0.78 | 0.49) (0.78 | 0.50) (0.78 | 0.52)
                      Forced (0.67 | 0.17) (0.75 | 0.17)
                      Crosswise (0.17 | 0.00) (0.20 | 0.00) (0.23 | 0.00)
(0.25 | 0.00) (0.26 | 0.00) (0.30 | 0.00) (0.75 | 0.00) (0.83 | 0.00)

## Estimated Population Prevalence (weighted per RR model)
   RRmodel estimate.weighted se.weighted    n
 Crosswise          0.081899   0.0444126 1427
       DQ           0.029167   0.0062712  720
    Forced          0.192582   0.0145974 1417
       UQM          0.070978   0.0195603  717


-----------------------------------------------------------
Item:                         handed in someone else's work
Model(s):             DQ (1.00 | 0.00)
                      UQM (0.70 | 0.49) (0.70 | 0.50) (0.70 | 0.52)
                          (0.78 | 0.49) (0.78 | 0.50) (0.78 | 0.52)
                      Forced (0.67 | 0.17) (0.75 | 0.17)
                      Crosswise (0.17 | 0.00) (0.20 | 0.00) (0.23 | 0.00)
 (0.25 | 0.00) (0.26 | 0.00) (0.30 | 0.00) (0.75 | 0.00) (0.83 | 0.00)

## Estimated Population Prevalence (weighted per RR model)
   RRmodel estimate.weighted se.weighted    n
 Crosswise          0.033014   0.0468658 1427
       DQ           0.015278   0.0045711  720
    Forced          0.146240   0.0134956 1417
       UQM          0.019240   0.0178254  717
```

The prevalence estimates are almost always the lowest for DQ. Most likely, students underreported their behavior when directly asked. In Figure 2, the weighted prevalence estimates are plotted for each item and RRT. This is the first of four plots when plotting an object of class `RRglmerMod` of the package **GLMMRR**. It can be seen that copying work is the most popular way, and then using a crib note. Although the CW condition has the highest number of students, the confidence interval of the CW estimates is much wider than for the other conditions. The CW method is less efficient in estimating student prevalence than the other RRTs.

Model fit

The relevance of the random person component is examined by comparing the multiple-group IRT model with a (multiple-group) GLM, which has the same fixed effect part.

```
R> out.fe <- RRglm(RR_response ~ 1 + Question + expcond, item=Question,
+     link = "RRlink.probit",RRmodel = RRmodel,p1=p1,p2=p2,data = ETHBE)
R> anova(out.re,out.fe)
Data: df
Models:
out.fe: RR_response ~ 1 + Question + expcond
out.re: RR_response ~ 1 + Question + expcond + (1 | id)
      npar  AIC   BIC logLik deviance  Chisq Df Pr(>Chisq)
out.fe   10 21147 21227 -10564    21127
out.re   11 21016 21103 -10497    20994 133.82  1  < 2.2e-16 ***
```
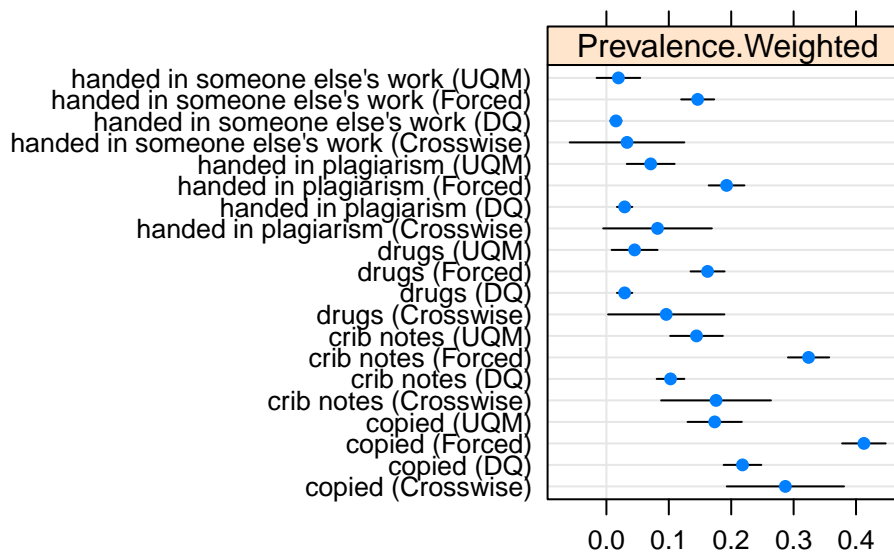
Prevalence (95%–CI)

Figure 2: Prevalence estimates of types of student misconduct for different RRTs.

It follows that the RIRT model fits the data better than the GLM, with a lower AIC and BIC and a significant decrease in deviance. For the RIRT model, conditional Pearson residuals are computed given the random effect. The Pearson residuals are plotted against the fitted probabilities. To obtain the fitted probabilities, the fitted values on the linear predictor scale are computed using the `predict` function. Then, according to Equation 1, the fitted probabilities are computed using parameters c and d, which can be computed from function `getRRparameters` given the RRT and RR design parameters.

```
R> eta <- predict(out.re, type = "link")
R> dum <- getRRparameters(ETHBE$RRmodel, ETHBE$p1, ETHBE$p2)
R> pp <- dum$c + dum$d*pnorm(eta)
R> resid <- residuals(out.re, type = c("pearson"))
R> plot(pp[ETHBE$expcond=="direct questioning"],
+    resid[ETHBE$expcond=="direct questioning"],bty="l",
+    xlim=c(0,1),ylim=c(-2,6),cex=.8,xlab="Fitted probability",ylab="Pearson residual")
R> points(pp[ETHBE$expcond=="CM pick-a-number"],
+    resid[ETHBE$expcond=="CM pick-a-number"],cex=.8,pch=19,col="grey80")
R> points(pp[ETHBE$expcond=="FR pick-a-number"],
+    resid[ETHBE$expcond=="FR pick-a-number"],cex=.8,pch=15,col="red")
R> points(pp[ETHBE$expcond=="FR random wheel"],
+    resid[ETHBE$expcond=="FR random wheel"],cex=.8,pch=16,col="blue")
R> points(pp[ETHBE$expcond=="CM unrelated question"],
     resid[ETHBE$expcond=="CM unrelated question"],cex=.8,pch=17,col="green")
R> points(pp[ETHBE$expcond=="UQ Benford"],
+     resid[ETHBE$expcond=="UQ Benford"],cex=.8,pch=18,col="purple")
R> abline(h=1,lty=2,col="grey")
R> abline(h=-1,lty=2,col="grey")
R> legend(.6,6,c("DQ","FR pick-a-number","FR random wheel",
+    "CM unrelated", "UQ","CM pick-a-number"),
+    col=c("black","red","blue","green","purple","grey80"),
+    pch = c(1,15,16,17,18,19),cex=.7,bty="n")
```

In Figure 3, the Pearson residuals are plotted for each RRT. It can be seen that for DQ, the residuals are large (small) for positive (zero) responses since the prevalence rates are low. The residuals for FR and UQ are relatively small, partly because the corresponding fitted probabilities are in the middle of the scale. For the CW methods, it can be seen that the residuals are large (small) for zero (positive) responses since they correspond to high prevalence rates. Further analysis

to improve the model by incorporating predictor variables is omitted for reasons of brevity.
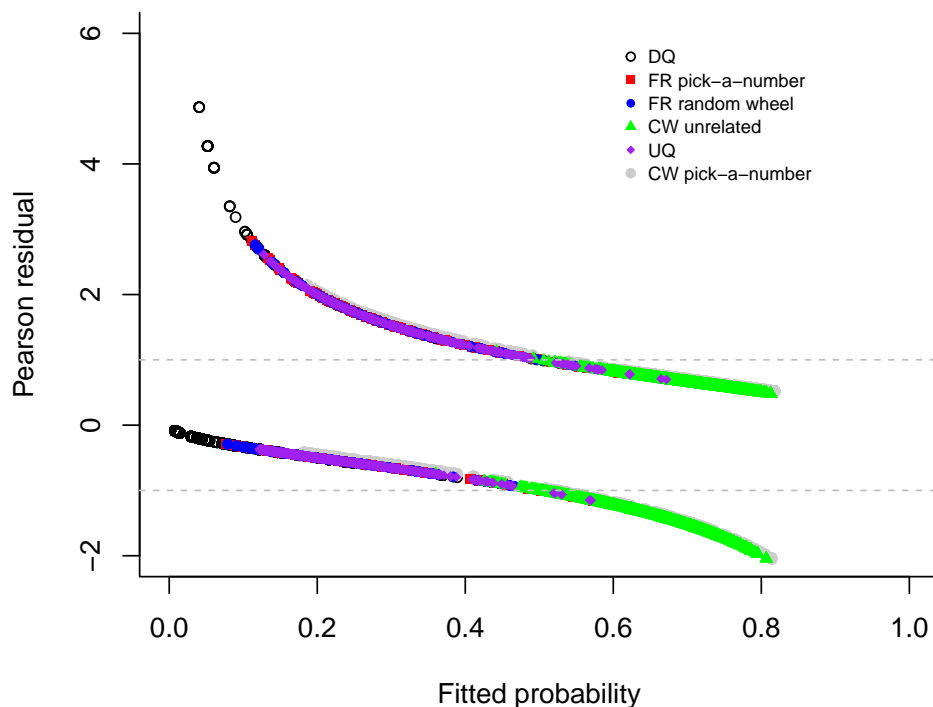

Figure 3: Fitted probabilities of types of student misconduct for different RRTs versus the Pearson residuals.

Discussion

The **GLMMRR** provides tools for fitting GLM(M)s on RR data, where the RR design and the design parameters are allowed to vary across observations. The multiple-group RR designs are particularly interesting to validate RR methods, to examine the sensitivity of the attributes, and to examine the influence of different levels of privacy protection. The multiple group modeling approach also supports simultaneously testing RR design effects across items and participants. The tools provide support to substantive applications using RR techniques, which can include different RR designs and advanced GLM and GLMM methods to analyze RR data. It is our objective to stimulate applied and methodological RR research by offering the open-source software **GLMMRR**. The tools extend the popular modeling tools of **lme4**, and class functions are simply extended to deal with RR data while maintaining the general features included in the GLM and GLMM software (e.g., **lme4**; Bates et al., 2015).

When some respondents do not follow the RR design instructions, the GLM(M) does not fit the data (e.g., Böckenholt and van der Heijden, 2007; Fox et al., 2013; De Jong et al., 2010). In that case, a composite link function can be used to include a linear predictor for those not following the instructions and one for those following the instructions. Currently, the ML estimations methods for **GLMMRR** cannot handle composite link functions. More research is needed to develop and implement estimation methods that can handle in a flexible way GLMMs with composite link functions (Thompson and Baker, 1981).

The modified link functions can be used for other types of GLMMs. In longitudinal research, in practice, time is often observed in discrete units. The discrete-time hazard defines the probability of the occurrence of an event at time t. The GLM(M) can be used to describe the link between the hazard rate and a linear predictor. The GLM(M) with the complementary log-log link function applies when the data is generated by a continuous-time proportional hazards model (Allison, 1982). Thus, for instance, when using an RR design to collect information about sensitive events (e.g.,

events related to war, trauma, sexual assault), the GLMM with a modified link function is an appropriate model to analyze the RR data. For count RR data, the complementary log-log link function can be used to model the probability of an RR of a non-zero observation (Fox et al., 2018). Ordinal data can be modeled with a (cumulative) link function for the proportional odds, which is defined by cumulative probabilities. For each response category, a GLMM defines a cumulative probability that response falls below or in this category. The modified link functions can be used to model cumulative probabilities with GLMMs for ordinal RR data.

The GLMMs can be computationally intensive and usually require relatively large sample sizes. RR designs also require larger samples sizes to achieve the same level of accuracy as DQ. Furthermore, the prevalence of sensitive behaviors is often relatively low, and to obtain reliable estimates, more data is required. The **GLMMRR** package supports large sample sizes, which can be around 10^6 observations.

Bibliography

P. D. Allison. Discrete-time methods for the analysis of event histories. *Sociological methodology*, 13: 61–98, 1982. URL https://doi.org/10.2307/270718. [p581]

D. Bates, M. Mächler, B. Bolker, and S. Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015. URL https://doi.org/10.18637/jss.v067.i01. [p581]

G. Blair, Y.-Y. Zhou, and K. Imai. rr: Statistical methods for the randomized response technique, 2015. URL https://cran.r-project.org/package=rr. R package version 1.4. [p561]

U. Böckenholt and P. G. van der Heijden. Item randomized-response models for measuring noncompliance: Risk-return perceptions, social influences, and self-protective responses. *Psychometrika*, 72(2):245–262, 2007. URL https://doi.org/10.1007/s11336-005-1495-y. [p581]

R. F. Boruch. Maintaining confidentiality on data in educational research: A systemic analysis. *American Psychologist*, 26(5):413–430, 1971. URL https://doi.org/10.1037/h0031502. [p563]

J. M. Chambers and T. J. Hastie. *Statistical models in S*. Pacific Grove, CA: Wadsworth & Brooks, 1992. [p565]

P. De Boeck, M. Bakker, R. Zwitser, M. Nivard, A. Hofman, F. Tuerlinckx, I. Partchev, et al. The estimation of item response models with the lmer function from the lme4 package in R. *Journal of Statistical Software*, 39(12):1–28, 2011. URL https://doi.org/10.18637/jss.v039.i12. [p566]

M. G. De Jong, R. Pieters, and J.-P. Fox. Reducing social desirability bias through item randomized response: An application to measure underreported desires. *Journal of Marketing Research*, 47(1): 14–27, 2010. URL https://doi.org/10.1509/jmkr.47.1.14. [p581]

J.-P. Fox. Multilevel IRT using dichotomous and polytomous response data. *British Journal of Mathematical and Statistical Psychology*, 58(1):145–172, 2005. URL https://doi.org/10.1348/000711005X38951. [p561]

J.-P. Fox. Bayesian randomized item response theory models for sensitive measurement. In *Handbook of modern item response theory. Vol. 1. Models*. Chapman and Hall/CRC Press, 2012. [p561]

J.-P. Fox and R. R. Meijer. Using item response theory to obtain individual information from randomized response data: An application using cheating data. *Applied Psychological Measurement*, 32(8):595–610, 2008. URL https://doi.org/10.1177/0146621607312277. [p561]

J.-P. Fox, M. Avetisyan, and J. van der Palen. Mixture randomized item-response modeling: A smoking behavior validation study. *Statistics in medicine*, 32(27):4821–4837, 2013. URL https://doi.org/10.1002/sim.5859. [p561, 581]

J.-P. Fox, D. Veen, and K. Klotzke. Generalized linear mixed models for randomized responses. *Methodology*, 2018. URL https://doi.org/10.1027/1614-2241/a000153. [p563, 565, 582]

D. W. Heck and M. Moshagen. RRreg: An R package for correlation and regression analyses of randomized response data. *Journal of Statistical Software*, 85(2):1–29, 2018. ISSN 1548-7660. URL https://doi.org/10.18637/jss.v085.i02. [p562]

M. Höglinger and B. Jann. More is not always better: An experimental individual-level validation of the randomized response technique and the crosswise model. *PloS one*, 13(8):e0201770, 2018. URL https://doi.org/10.1371/journal.pone.0201770. [p562, 563, 570, 572]

M. Höglinger, B. Jann, and A. Diekmann. Online survey on exams and written papers; documentation. University of Bern Social Sciences Working Papers 8, University of Bern, Department of Social Sciences, May 2014. URL https://ideas.repec.org/p/bss/wpaper/8.html. [p575, 578]

M. Höglinger, B. Jann, and A. Diekmann. Sensitive questions in online surveys: An experimental evaluation of different implementations of the randomized response technique and the crosswise model. In *Survey Research Methods*, volume 10, pages 171–187, 2016. URL https://doi.org/10.18148/srm/2016.v10i3.6703. [p562]

D. W. Hosmer and S. Lemeshow. Goodness of fit tests for the multiple logistic regression model. *Communications in statistics-Theory and Methods*, 9(10):1043–1069, 1980. URL https://doi.org/10.1080/03610928008827941. [p567]

L. K. John, G. Loewenstein, A. Acquisti, and J. Vosgerau. When and why randomized response techniques (fail to) elicit the truth. *Organizational Behavior and Human Decision Processes*, 148: 101–123, 2018. URL https://doi.org/10.1016/j.obhdp.2018.07.004. [p562]

A. Y. Kuk. Asking sensitive questions indirectly. *Biometrika*, 77(2):436–438, 1990. URL https://doi.org/10.1093/biomet/77.2.436. [p563]

G. J. Lensvelt-Mulders, J. J. Hox, P. G. van der Heijden, and C. J. Maas. Meta-analysis of randomized response research: Thirty-five years of validation. *Sociological Methods & Research*, 33(3):319–348, 2005. URL https://doi.org/10.1177/0049124104268664. [p561]

P. McCullagh and J. Nelder. *Generalized linear models (2nd ed.)*. London, UK: Chapman & Hall, 1989. [p564]

F. Rijmen, F. Tuerlinckx, P. De Boeck, and P. Kuppens. A nonlinear mixed model framework for item response theory. *Psychological methods*, 8(2):185–205, 2003. URL https://doi.org/10.1037/1082-989X.8.2.185. [p566]

N. Scheers and C. M. Dayton. Covariate randomized response models. *Journal of the American Statistical Association*, 83(404):969–974, 1988. URL https://doi.org/10.1080/01621459.1988.10478686. [p561]

R. Thompson and R. Baker. Composite link functions in generalized linear models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 30(2):125–131, 1981. URL https://doi.org/10.2307/2346381. [p581]

R. Tourangeau and T. W. Smith. Asking sensitive questions: The impact of data collection mode, question format, and question context. *The Public Opinion Quarterly*, 60(2):275–304, 1996. URL http://www.jstor.org/stable/2749691. [p561]

R. Tourangeau and T. Yan. Sensitive questions in surveys. *Psychological Bulletin*, 135(5):859–883, 2007. URL https://doi.org/10.1037/0033-2909.133.5.859. [p561]

G. Tutz. *Regression for categorical data*, volume 34. Cambridge University Press, 2011. [p564]

A. van den Hout, U. Böckenholt, and P. G. van der Heijden. Estimating the prevalence of sensitive behaviour and cheating with a dual design for direct questioning and randomized response. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 59(4):723–736, 2010. URL https://doi.org/10.1111/j.1467-9876.2010.00720.x. [p561]

S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965. URL https://doi.org/10.1080/01621459.1965.10480775. [p561, 563]

F. Wolter and P. Preisendörfer. Asking sensitive questions: An evaluation of the randomized response technique versus direct questioning using individual validation data. *Sociological Methods & Research*, 42(3):321–353, 2013. URL https://doi.org/10.1177/0049124113500474. [p562]

J.-W. Yu, G.-L. Tian, and M.-L. Tang. Two new models for survey sampling with sensitive characteristic: design and analysis. *Metrika*, 67(3):251, 2008. URL https://doi.org/10.1007/S00184-007-0131-X. [p563]

Jean-Paul Fox
Department of Research Methodology, Measurement and Data Analysis
Faculty of Behavioural, Management and Social Sciences
University of Twente
7500AE Enschede, The Netherlands
E-mail: j.p.fox@utwente.nl

Konrad Klotzke
Department of Research Methodology, Measurement and Data Analysis
Faculty of Behavioural, Management and Social Sciences
University of Twente
7500AE Enschede, The Netherlands
E-mail: k.k.klotzke@utwente.nl

Duco Veen
Julius Global Health
Julius Center for Health Sciences and Primary Care
University Medical Center Utrecht
3584CX Utrecht, The Netherlands
Optentia Research Program
North-West University
1900 Vanderbijlpark, South Africa
E-mail: D.Veen-5@umcutrecht.nl