Università degli Studi di Padova
Dipartimento di Scienze Statistiche

Corso di Laurea Triennale in
Statistica per l'Economia e l'Impresa

Relazione Finale

Improved likelihood inference in rater agreement models

Relatore Prof. Alessandra Salvan
Dipartimento di Scienze Statistiche

Laureanda Scattolin Alice
Matricola 1193137

Anno Accademico 2020/2021

Contents

Introduction

Log-linear models are the most commonly used to model categorical variables. In this work the focus will be on *ordinal variables* and their symmetric relationships in contingency tables. In particular, models for rater agreement will be presented to analyse *matched-pairs data.* That is, in a pair each observation uses the same ordinal scale. Matched-pairs data are usually presented in contingency tables whose row and column variables are identical.

These data occur in multiple situations:

- each subject is observed at two different times,
- each subject is observed in two different sites,
- given opinion by one person on two different topics with the same response scale,
- given opinion by two people on the same topic.

This work will focus on the last situation and describe models to analyze agreement between two judges.

Furthermore, inference based on the likelihood will be central to the analysis. The maximum likelihood estimator will be presented in view of its well-known asymptotic properties and its vast application. However, its behavior in finite samples may be unsatisfactory. Hence different approaches to improve its properties will be depicted and some of them will be examined through Monte Carlo simulations.

Chapter 1

Ordinal variables and inference based on the likelihood

1.1 Ordinal variables and contingency tables

An *ordinal variable* is a categorical variable with an ordered categorical scale. Let Y be an ordinal variable and c the number of its levels. Let n be the number of observations in a sample and $n_1, ..., n_j, ..., n_c$ the frequencies in the categories. The *sample proportions* are given by $p_j = n_j/n,\ j = 1, ..., c$.

In a square contingency table, let X be the row ordinal variable and Y be the column ordinal variable and c the number of levels each. Let n_{ij} be the frequency of the cell in the i-th row and j-th column, $i, j = 1, ..., c$, and n the total sample size. The sample proportions are given by $p_{ij} = n_{ij}/n$. Hence, $\sum_i \sum_j p_{ij} = 1$ and $\{p_{ij}\}$ is the sample *joint distribution.* The sample *marginal distributions* are given by p_{+j} and p_{i+}. It follows that $\sum_i p_{ij} = p_{+j}$ and $\sum_j p_{ij} = p_{i+}$. For a given i, $p_{j|i}$ denotes the proportion in category j. Then, $p_{j|i} = n_{ij}/n_{i+}$ and $\sum_j p_{j|i} = 1$ for all $i,\ i = 1, ..., c$.

In a square 2×2 table, the *odds ratio* is defined by

$$\hat{\theta} = \frac{p_{1|1}/p_{2|1}}{p_{1|2}/p_{2|2}}.$$

If $\hat{\theta} > 1$, the proportion of subjects that gave response 1 is greater in row 1 than in row 2.

1.2 Model specification

Let y_i be the observation of a univariate variable Y_i and $\mathbf{x}_i = (x_{i1}, ..., x_{ip})$ be the vector of observations of covariates, $i = 1, ..., n$. Let X be the $n \times p$ matrix, with rows $\mathbf{x}_1^\top, ..., \mathbf{x}_n^\top$.

The problem of specifying a regression model for Y_i can be divided into two steps:

- specify the model for the response distribution,
- link the parameters of this model to the covariates.

For the first step, it is essential to analyze the type of the response distribution and its characteristics (symmetry etc.). For the second step, the link function between the expected value of the response and covariates will be assumed as known. In particular, let $E(Y_i) = \mu_i = \theta(\mathbf{x}_i; \beta)$.

1.2.1 Model specification levels

Models can be divided into three categories, depending on the level of specification: parametric, semi-parametric and non-parametric.

Parametric models are those in which the distribution on Y_i is fully specified apart from a vector of unknown parameters. The Gaussian linear model or logistic regression are examples of parametric models.

Semi-parametric models are indexed by an unknown finite dimensional parameter and an unknown component not expressed by a finite number of constants (such as a function). The Gaussian linear regression with second order hypothesis is an example.

Finally, in *non-parametric models* the distribution of Y_i is specified except for an unknown component which cannot be described by a finite number of constants.

This work will deal with parametric models.

1.3 Inference based on the likelihood in parametric models

Let $\mathcal{F}$ be a parametric family of density functions

$$\mathcal{F} = \{p_Y(y;\theta) : \theta \in \Theta \subset \mathbb{R}^p\}.$$

When dealing with discrete data, $p_Y(y;\theta)$ is a probability mass function. Through the data y we would like to define the data-generation process Y. To uniquely identify the true parameter θ_0, two conditions must be met:

- θ must be identifiable: if $\theta_1 \neq \theta_2$, $p_Y(y;\theta_1)$ and $p_Y(y;\theta_2)$ are different distributions,
- correct specification of $\mathcal{F}$: let $p^0(y)$ be the true density of Y, we must have $p^0(y) \in \mathcal{F}$.

Some inferential procedures are valid under *regularity conditions*, which request

- the likelihood to be a regular function,
- the support of Y to not depend on the parameter θ,
- order inversions of differentiation and integration to be allowed.

1.3.1 The likelihood function

Let y be the vector of observations and $\mathcal{F}$ be the statistical model, the *likelihood function* is defined as

$$L(\theta) = L(\theta; y) = c(y)p_Y(y;\theta),$$

where $L(\theta) : \Theta \to [0, +\infty)$.

In a regression model, $Y = (Y_1, ..., Y_n)$ and Y_i are independent for all $i = 1, ..., n$. Hence, the likelihood function is

$$L(\theta) = p_Y(y;\theta) = \prod_{i=1}^{n} p_{Y_i}(y_i;\theta).$$

The *log-likelihood function* is

$$l(\theta) = \log(L(\theta)) = \log(p_Y(y;\theta)) = \log\left(\prod_{i=1}^{n} p_{Y_i}(y_i;\theta)\right) = \sum_{i=1}^{n} \log(p_{Y_i}(y_i;\theta)),$$

where $l(\theta) = -\infty$ if $L(\theta) = 0$.

1.3.2 Likelihood quantities

The first derivative of $l(\theta)$ is usually called *score*

$$l_*(\theta) = l_*(\theta; y) = \left(\frac{\partial l(\theta)}{\partial \theta_1}, ..., \frac{\partial l(\theta)}{\partial \theta_p}\right)^{\top}.$$

The opposite of the second derivative of $l(\theta)$ is called *observed information*

$$j(\theta) = j(\theta; y) = -\frac{\partial^2 l(\theta)}{\partial \theta \partial \theta^{\top}}.$$

The *Fisher information* is defined as the expected value of the observed information

$$i(\theta) = E_\theta\{j(\theta; Y)\}.$$

Under regularity conditions, the following identities hold:

- $E_\theta(l_*(\theta)) = 0, \ \theta \in \Theta$
- $E_\theta(l_* l_*^{\top}) = i(\theta)$ for each $\theta \in \Theta$.

They are also known as *Bartlett identities.*

1.3.3 Maximum likelihood estimator

A maximum likelihood estimate of θ is a value of θ that maximizes $L(\theta)$. Since the logarithm is a monotonic function, $l(\theta)$ is usually maximized with respect to θ. We assume that $\hat{\theta}$ exists and is unique. Once we consider the random variables Y_i, we obtain the maximum likelihood estimator (MLE).

Under regularity conditions, $\hat{\theta}$ is a solution of the equation

$$l_*(\theta) = \frac{\partial l}{\partial \theta} = 0.$$

This equation is called *likelihood equation*, which must in general be solved numerically.

It can be shown that under regularity conditions the MLE has the following properties as $n \to +\infty$:

- *consistency*: $\hat{\theta} \xrightarrow{p} \theta_0$
- *asymptotic normality*: $\sqrt{n}(\hat{\theta} - \theta_0) \xrightarrow{d} N_p\left(0, i(\theta_0)^{-1}\right)$, with $i(\theta_0) = \lim\limits_{n\to\infty} \frac{i(\theta_0)}{n}$.

$i(\theta_0)$ may be estimated by $i(\hat{\theta})$ or by $j(\hat{\theta})$.

1.3.4 Confidence regions and tests

Confidence regions and tests can be based on three asymptotically pivotal quantities having an asymptotic distribution χ^2_p under θ. These are the Wald, likelihood ratio and score pivotal quantities.

The *Wald* pivot is

$$W_e(\theta) = (\hat{\theta} - \theta)^\top j(\hat{\theta})(\hat{\theta} - \theta). \tag{1.1}$$

To test the null hypothesis $H_0 : \theta = \theta_0$, large values of $W_e(\theta_0)$ are considered significant against H_0.

The *likelihood ratio* pivot is

$$W(\theta) = 2(l(\hat{\theta}) - l(\theta)). \tag{1.2}$$

To test the null hypothesis $H_0 : \theta = \theta_0$, large values of $W(\theta_0)$ are considered significant against H_0.

The *score* pivot, also known as *Lagrange Multiplier Test*, has the following structure

$$W_u(\theta) = l_*(\theta)^\top i(\theta)^{-1} l_*(\theta) \tag{1.3}$$

It is based on the restricted model instead of the unrestricted. The basic idea is that, if the null hypothesis is true, the first derivatives of the log-likelihood evaluated at the restricted parameter vector will be approximately zero. To test that, the reasoning is analogous to the Wald test.

Chapter 2

Generalized linear models

2.1 Introduction

Generalized linear models (GLM) were introduced by Nelder & Wedderburn (1972) to generalize the linear model.

Let $y_1, ..., y_n$ be observations on independent random variables $Y_1, ..., Y_n$ each with probability density or mass function of the exponential dispersion family form

$$p_{Y_i}(y_i; \theta_i, \phi) = \exp\left\{\frac{\theta_i y_i - b(\theta_i)}{a_i(\phi)} + c(y_i; \phi)\right\}$$

for some sufficiently smooth functions $b(\cdot)$, $c(\cdot)$ and $a(\cdot)$. Let m_i be fixed observation weights, $i = 1, ..., n$.

The expected value and the variance of Y_i are then

$$E(Y_i) = \mu_i = b'(\theta_i),$$

$$Var(Y_i) = a_i(\phi)b''(\theta_i) = a_i(\phi)V(\mu_i),$$

where $b'(\theta_i)$ and $b''(\theta_i)$ are the first and second derivatives of $b(\theta_i)$ with respect to θ_i. $V(\mu_i)$ is called the *variance function* and allows the heteroscedasticity of the model.

A *link function* is assumed to be known and aims to link the mean μ_i of the response variable to the covariates linearly in the parameters. Let η_i be the *linear* *predictor* and $g(\cdot)$ the link function. Hence, the model assumes

$$g(\mu_i) = \eta_i = \mathbf{x}_i^\top \beta,$$

where $\mathbf{x}_i^\top$ is the i-th row of the covariates matrix and β is the parameters vector.

When $\theta_i = g(\mu_i)$, $g(\cdot)$ is called *canonical link function.*

2.2 Inference based on the likelihood in GLM

Assuming independence, the response variables $Y_1, ..., Y_n$ have joint distribution equal to the product of the marginal distributions. Hence, the log-likelihood function is

$$l(\beta, \phi) = \sum_{i=1}^{n} \frac{y_i \theta_i - b(\theta_i)}{a_i(\phi)} + \sum_{i=1}^{n} c(y_i; \phi) \tag{2.1}$$

where $\theta_i = \theta_i(\mu_i) = \theta_i(g^{-1}(\mathbf{x}_i^\top \beta))$.

When $g(\cdot)$ is canonical, the log-likelihood simplifies to

$$l(\beta, \phi) = \sum_{i=1}^{n} \frac{y_i \mathbf{x}_i^\top \beta - b(\mathbf{x}_i^\top \beta)}{a_i(\phi)} + \sum_{i=1}^{n} c(y_i; \phi).$$

When ϕ is known, the score function can be obtained by first-order differentiation of the log-likelihood. The likelihood equations for β are

$$l_r = \sum_{i=1}^{n} \frac{(y_i - \mu_i)}{Var(Y_i)} \frac{\partial \mu_i}{\partial \beta_r} = 0, \tag{2.2}$$

with $r = 1, ..., p$.

When $g(\cdot)$ is canonical, these equations simplify to

$$\sum_{i=1}^{n} \frac{1}{a_i(\phi)} y_i x_{ir} = \sum_{i=1}^{n} \frac{1}{a_i(\phi)} \mu_i x_{ir},$$

with $r = 1, ..., p$.

The score equations can be written in the matrix form as

$$D^\top V^{-1}(y - \mu) = 0, \tag{2.3}$$

where $(y-\mu) = (y_1 - \mu_1, ..., y_n - \mu_n)^\top$, $V = \text{diag}[Var(Y_i)]$, $i = 1, ..., n$, and D is a $n \times p$ matrix with generic element

$$d_{ir} = \frac{\partial \mu_i}{\partial \beta_r} = \frac{1}{g'(\mu_i)} x_{ir},$$

with $i = 1, ..., n$ and $r = 1, ..., p$.

2.3 Iterative weighted least squares

The score equations have generally no explicit solution for β, hence an iterated numeric algorithm is needed. This paragraph explores the *iterative weighted least squares*, also known as *IWLS*.

Let l_β the vector with elements l_r and $j_{\beta\beta}$ the (β, β) block of the information matrix about β and ϕ. The $(j+1)$-th iteration give the approximation

$$\hat{\beta}^{(j+1)} = \hat{\beta}^{(j)} + [j_{\beta\beta}^{(j)}]^{-1} l_\beta^{(j)}, \tag{2.4}$$

where the superscript (j) indicates the evaluation at $\hat{\beta}^{(j)}$. In this expression, $j_{\beta\beta}$ can be replaced with its expectation $i_{\beta\beta}$.

From this expression, the following can be derived

$$X^\top W X \hat{\beta}^{(j+1)} = X^\top W z^{(j)}, \tag{2.5}$$

where the generic element of $z^{(j)}$ is $z_i^{(j)} = \mathbf{x}_i^\top \hat{\beta}^{(j)} + (y_i - \mu_i) g'(\mu_i)$, $i = 1, ..., n$. The variable $z^{(j)}$ is usually called *adjusted dependent variable* and can be interpreted as the linear approximation of $g(y_i)$.

When the algorithm converges, we will have

$$\hat{\beta} = (X^\top \hat{W} X)^{-1} X^\top \hat{W} \hat{z}$$

with $\hat{z} = X\hat{\beta} + \hat{u}$.

For further details, see Salvan, Sartori & Pace (2020, Section 2.3.6).

2.4 Log-linear models

We will now focus on log-linear models. They are usually employed for count responses, such as the number of people in a queue, the number of trees alive in forest, the number of articles published by a certain journalist, etc.

In log-linear models

- the response variable Y_i is a count with mean μ_i,

- the link-function is the logarithm.

Let X be the matrix of covariates and $\mathbf{x}_i^\top$ its i-th row, $i = 1, ..., n$. Let $\mathbf{x}_i^\top \beta$ be the *linear predictor*.

Hence, a log-linear model has the following structure, $i = 1, ..., n$,

$$\log(\mu_i) = \mathbf{x}_i^\top \beta.$$

2.4.1 Poisson log-linear model

It is natural to choose the Poisson distribution for the response variable. This will lead to the Poisson log-linear model.

In this case, the log-likelihood function is

$$l(\beta) = \sum_{i=1}^{n} \{y_i \log(\mu_i) - \mu_i\},$$

where $\mu_i = g^{-1}(\mathbf{x}_i^\top \beta) = \exp(\mathbf{x}_i^\top \beta)$.

For $r = 1, ..., p$, the likelihood equations are

$$\sum_{i=1}^{n} y_i x_{ir} = \sum_{i=1}^{n} e^{\mathbf{x}_i^\top \beta} x_{ir}. \tag{2.6}$$

In this model,

$$\mu_i = \exp\left(\sum_{r=1}^{p} \beta_r x_{ir}\right) = (e^{\beta_1})^{x_{i1}} ... (e^{\beta_p})^{x_{ip}}.$$

Hence, when x_{ir} increases by one unit, the mean μ_i increases by a multiplicative factor e^{β_r}, given the others covariates.

To evaluate the goodness of fit, the deviance can be used if n is fixed and μ_i is large enough, for all $i = 1, ..., n$. In a Poisson model, the deviance is approximately equal to the Pearson's statistics

$$D(y; \hat{\mu}) = 2\sum_{i=1}^{n}\{y_i \log \frac{y_i}{\hat{\mu}_i} - y_i + \hat{\mu}_i\} = 2\sum_{i=1}^{n}\{o_i \log \frac{o_i}{a_i} - o_i + a_i\} \doteq \sum_{i=1}^{n} \frac{(o_i - a_i)^2}{a_i}, \quad (2.7)$$

with $o_i = y_i$ and $a_i = \hat{\mu}_i$. When $\hat{\mu}_i$ is sufficiently large for all $i = 1, ..., n$, the deviance follows a χ^2_{n-p} distribution under the model.

To improve the model, interactions between covariates could be added or the negative binomial distribution used instead of the Poisson one when overdispersion is present. For further details see Salvan, Sartori & Pace (2020, chap. 5).

Chapter 3

Models for rater agreement

3.1 Introduction

To model cell counts in contingency tables log-linear models are used. In this chapter we will investigate models for agreement between two observers or *raters*.

Firstly, it is crucial to highlight the difference between *agreement* and *association*. As explained by Agresti (2019, Section 8.5), strong agreement requires strong association, but strong association can exist without strong agreement. For example, if we have two raters A and B and rater B classifies subjects systematically one level higher than rater A, we will have strong positive association but poor agreement.

In the first place, the independence model will be recalled and its possible poor fit shown by an example. In fact, observed ratings on the main diagonal are usually higher than expected by chance. For this reason, a quasi-independence model will be introduced. However, for ordinal variables, given disagreement, positive association still occurs: the ordinal agreement model tries to detect this pattern.

Eventually, overall agreement can be split up into three components: chance agreement (what would occur even if classifications were independent), agreement due to a baseline between the ratings and an increment that reflects agreement in excess of that occurring simply from chance agreement and from the baseline association. Further details about this heuristic division can be found in Agresti (1988) and about the following models in Agresti (2010, Section 8.5).

3.2 Notation

In square tables, let A be a factor whose categories are displayed on the rows and r be the number of rows. Let B be another factor whose categories are displayed on the columns and c be the number of columns. Let u_j be the score for the j-th category of factor A, $j = 1, ..., r$, and v_h be the score for the h-th category of factor B, $h = 1, ..., c$. For instance, ordered category scores may be $u_j = j$ and $v_h = h$. Y_{jh} is the random variable that describes the frequency corresponding to the j-th row and h-th column, $j = 1, ..., r$ and $h = 1, ..., c$. Let y_{jh} be the observed frequency corresponding to the j-th row and h-th column, $j = 1, ..., r$ and $h = 1, ..., c$. The total sample size is $y_{++} = \sum_{j=1}^{r} \sum_{h=1}^{c} y_{jh}$. The sample marginal distributions are $y_{j+} = \sum_{h=1}^{c} y_{jh}$, $j = 1, ..., r$, and $y_{+h} = \sum_{j=1}^{r} y_{jh}$, $h = 1, ..., c$.

Let λ, λ^A and λ^B be the parameters related respectively to the intercept, the indicator variables for factor A and the indicator variables for factor B. λ_j^A is the parameter related to the j-th level of factor A and λ_h^B is the parameter related to the h-th level of factor B, $j = 1, ..., r$ and $h = 1, ..., c$.

When dealing with matched-pairs data, A and B are ordinal variables and have the same categories. Hence, in this particular case, $r = c$ and $u_j = v_j$, $j = 1, ..., r$. A and B will be the explanatory variables to model the frequency of each cell.

3.3 Independence model

The independence model assumes independence among the ordinal variables. Although it usually fits poorly, Agresti (2010) described it as a baseline due to its possible interpretation as the grade of agreement expected if no association occurs. In fact, positive standardized residuals ideally stay on the main diagonal and reveal the presence of agreement because observed frequencies are higher than expected by the model. Oppositely, disagreement occur less than expected by chance, hence off-the-diagonal standardized residuals are expected to be negative.

Let μ_{ij} be the expected number of subjects classified as i by rater A and as j by rater B. The independence model assumes

$$\log(\mu_{jh}) = \lambda + \lambda_j^A + \lambda_h^B, \tag{3.1}$$

with $j = 1, ..., r$ and $h = 1, ..., r$.

3.3.1 Inference on independence model

In the independence model, the log-likelihood function has the following structure:

$$l(\lambda, \lambda^A, \lambda^B) = \sum_{j=1}^{r}\sum_{h=1}^{r} y_{jh} \log \mu_{jh} - \sum_{j=1}^{r}\sum_{h=1}^{r} \mu_{jh}$$

$$= y_{++}\lambda + \sum_{j=1}^{r} y_{+j}\lambda_j^A + \sum_{h=1}^{r} y_{+h}\lambda_h^B - \sum_{j=1}^{r}\sum_{h=1}^{r} \exp(\lambda + \lambda_j^A + \lambda_h^B).$$

Therefore, the first-derivatives of the log-likelihood function are the following

$$\frac{\partial l}{\partial \lambda} = y_{++} - \sum_{j=1}^{r}\sum_{h=1}^{r} \exp(\lambda + \lambda_j^A + \lambda_h^B) = y_{++} - \mu_{++}$$

$$\frac{\partial l}{\partial \lambda_j^A} = y_{j+} - \sum_{j=1}^{r}\sum_{h=1}^{r} \exp(\lambda + \lambda_j^A + \lambda_h^B) = y_{j+} - \mu_{j+}$$

$$\frac{\partial l}{\partial \lambda_h^B} = y_{+h} - \sum_{j=1}^{r}\sum_{h=1}^{r} \exp(\lambda + \lambda_j^A + \lambda_h^B) = y_{+h} - \mu_{+h}$$

The MLE satisfies $y_{j+} = \mu_{j+}$ and $y_{+h} = \mu_{+h}$ and these also imply $y_{++} = \mu_{++}$, $j = 1, ..., r$ and $h = 1, ..., r$.

3.4 Quasi-independence model

The quasi-independence model adds a parameter δ to describe agreement and has the following structure

$$\log(\mu_{jh}) = \lambda + \lambda_j^A + \lambda_h^B + \delta I(j = h), \tag{3.2}$$

where $I(j = h) = 1$ if $j = h$ and $I(j = h) = 0$ otherwise.

The model is called of *quasi-independence* because, when $j \neq h$, variables A and B are independent.

The MLE fit can be easily implemented by the usual software. The model fit can be evaluated by the deviance if μ_{jh} are sufficiently large, with degrees of freedom

$rr - [1 + (r-1) + (r-1) + 1] = rr - 2r$.

To improve the model fit, the parameter δ could be assumed as dependent of the $j - th$ category. That is, when $j = h$ and $\delta_j > 0$, the expected frequency μ_{jj} is higher than expected by chance, but these differences are not the same for all $j,\ j = 1, ..., r$.

3.5 Ordinal agreement model

The ordinal agreement model uses two parameters, β and δ, to describe the agreement. The parameter δ describes the association on the main diagonal, while β is informative on the agreement off that diagonal.

The model has the following structure

$$\log(\mu_{jh}) = \lambda + \lambda_j^A + \lambda_h^B + \beta u_j u_h + \delta I(j = h) \tag{3.3}$$

It should be noticed that, when $j = h$, μ_{jh} represents the frequency of agreement for the j-th category. In this case, an additional constant δ is added to the linear predictor: it can interpreted as the agreement in excess.

When $j \neq h$, the frequencies μ_{jh} express disagreement. The relation between $\log(\mu_{jh})$ and u_j is linear in β when u_h is fixed; viceversa the relation between $\log(\mu_{jh})$ and u_h is linear in β when u_j is fixed.

3.5.1 MLE for the ordinal agreement model

The kernel of the log-likelihood function is

$$l(\lambda, \lambda^A, \lambda^B) = \sum_{j=1}^{r}\sum_{h=1}^{r} y_{jh}\log\mu_{jh} - \sum_{j=1}^{r}\sum_{h=1}^{r}\mu_{jh}$$

$$= \sum_{j=1}^{r}\sum_{h=1}^{r} y_{jh}(\lambda + \lambda_j^A + \lambda_h^B + \beta u_j u_h + \delta I(j=h)) +$$

$$- \sum_{j=1}^{r}\sum_{h=1}^{r} \exp(\lambda + \lambda_j^A + \lambda_h^B + \beta u_j u_h + \delta I(j=h)).$$

Hence, the likelihood equations are the following:

$$\frac{\partial l}{\partial \lambda} = y_{++} - \sum_{j=1}^{r}\sum_{h=1}^{r} \exp(\lambda + \lambda_j^A + \lambda_h^B + \beta u_j u_h + \delta I(j=h)) = y_{++} - \mu_{++} = 0$$

$$\frac{\partial l}{\partial \lambda_j^A} = y_{j+} - \sum_{h=1}^{r} \exp(\lambda + \lambda_j^A + \lambda_h^B + \beta u_j u_h + \delta I(j=h)) = y_{j+} - \mu_{j+} = 0$$

$$\frac{\partial l}{\partial \lambda_h^B} = y_{+h} - \sum_{j=1}^{r} \exp(\lambda + \lambda_j^A + \lambda_h^B + \beta u_j u_h + \delta I(j=h)) = y_{+h} - \mu_{+h} = 0$$

$$\frac{\partial l}{\partial \beta} = \sum_{j=1}^{r}\sum_{h=1}^{r} y_{jh}u_j u_h - \sum_{j=1}^{r}\sum_{h=1}^{r} u_j u_h \mu_{jh} = 0.$$

Log-likelihood functions are concave for log-linear models, hence a maximum can be generally found. Infinite estimates occur for β when no pairs of observations agree or no pairs of observations disagree or, in other words, when all observations are on the main diagonal or all of them are off the diagonal. Usual software for generalized linear model can easily implement the MLE fit.

As for log-linear models, the deviance can be interpreted as goodness of fit only if y_{jh} are sufficiently high. Its degrees of freedom are

$rr - [1 + (r-1) + (r-1) + 2] = rr - 2r - 1.$

In the ordinal agreement model *odds ratios* can be interpreted as measures that the raters agree rather than disagree. For categories j and h, the odds ratio is

$$\tau_{jh} = \frac{\pi_{jj}\pi_{hh}}{\pi_{jh}\pi_{hj}}$$

where π_{jh} is the population proportion for the element j, h. For the ordinal agreement model, $\log(\tau_{jh}) = (u_j - u_h)^2\beta + 2\delta$. If $\beta > 0$ and $\delta > 0$, the odds ratio increases when categories j and h are far from each other. The sample odds ratios will be obtained by computing the corresponding sample proportions.

3.6 Example: movie critics

An application of the models presented in this chater will be illustrated. The dataset in Table 8.7 in Agresti (2010, Section 8.5) contains information about two movie critics' opinions. Table 3.1 summarizes ratings of 160 movies given by Gene Siskel and Roger Ebert for at-the-time Chicago newspaper from April 1995 to September 1996. In each cell, the absolute frequency of the matched ratings by Siskel and Ebert is displayed.

TABLE 3.1: Ratings of movies given by Gene Siskel and Roger Ebert

	Ebert Rating		
Siskel Rating	Con	Mixed	Pro
Con	24	8	13
Mixed	8	13	11
Pro	10	9	64

In table 3.2 the same data are displayed into three columns. The first one shows Siskel's rating, the second one shows Ebert's rating and the third one shows the observed frequencies of the matched-data. The following columns show the fitted values of the independence model, the quasi-independence model assuming δ constant, the quasi-independence model assuming δ varying with the level of the rating and the ordinal agreement model, respectively. In the rater agreement model, we have imposed $Con = 1$, $Mixed = 2$, $Pro = 3$ to estimate β. Both the quasi-independence models with constant δ and variable δ seem to perform the best in terms of fitted frequencies and residual deviance and indicate the presence of agreement on the main diagonal. On the other hand, the independence model has the highest AIC which suggests that its performance is poor and that the variables are not independent. In the ordinal agreement model, the parameter β is not statistically significant which means that, when one variable is fixed, there is no linear association. More details on the R code can be found in Appendix .

Siskel	Ebert	Obs.	Ind.	Quasi-ind.c	Quasi-ind.	O.A.
Con	Con	24	11.8	24.1	24.0	25.1
Mixed	Con	8	8.4	6.2	7.9	6.8
Pro	Con	10	21.8	11.6	10.1	10.1
Con	Mixed	8	8.4	6.3	8.1	7.1
Mixed	Mixed	13	6.0	14.9	13.0	13.0
Pro	Mixed	9	15.6	9.1	8.9	9.9
Con	Pro	13	24.7	14.5	12.9	12.9
Mixed	Pro	11	17.6	11.2	11.1	12.2
Pro	Pro	64	45.6	62.3	64.0	62.9

TABLE 3.2: Fitted values of ratings by Siskel and Ebert
Obs. = observed frequencies, Ind. = independence model,
Quasi-ind.c = quasi-independence model with constant δ,
Quasi-ind. = quasi-independence model with variable δ_j,
O.A. = ordinal agreement model

Chapter 4

Improvements in inference based on the likelihood

4.1 Introduction

In GLMs the MLE is usually the preferred estimator because it is asymptotically unbiased, consistent and efficient under regularity conditions and, as highlighted in chapter 1, inferential procedures based on Wald, score or likelihood ratio perform well in large samples. However, small sample sizes (or when the sample size is not enough larger than the number of parameters to estimate) may cause the spoiling of these asymptotic properties and procedures.

As described by Kosmidis (2014), a repository of methods is available to reduce mean and median bias and can be divided in explicit and implicit methods. Explicit methods of bias correction aim to estimate a function $B(\theta)$ so that

$$\tilde{\theta} = \hat{\theta} - B(\theta)$$

has the smallest bias. We will firstly discuss the so-called *location-adjusted MLE*, which is based on a higher order approximation of the MLE. Despite their theoretical stability, they generally involve tedious formulae or high computational effort. In addition, the existence of the MLE is required by explicit methods but it is not generally guaranteed. For these reasons, section 4.3 and section 4.4 will analyze implicit methods of *bias reduction* (BR) which are based on score adjustment. Many simulation studies have provided empirical support not only to the property of these methods of reducing mean and median bias but also of dealing with infinite estimates. As it will be shown, the solution of the adjusted likelihood equation can be computed through IWLS.

In section 4.5 we will study the mean BR and median BR estimators in the quasi-independence model and study their finite sample properties through Monte-Carlo simulations. In particular, δ will be our parameter of interest. The package `brglm2` at Kosmidis (2020) will be used to estimate the coefficients.

4.2 Location-adjusted mean bias correction

In regular parametric models the MLE is asymptotically unbiased and $E_\theta(\hat{\theta}) = \theta + O(n^{-1})$ by expansion of decreasing powers of n. Cox & Hinkley (1974, Section 9.2) proved that the asymptotic mean bias of the MLE can be corrected by the explicit formula

$$\tilde{\theta} = \hat{\theta} - b(\hat{\theta})/n,$$

where the solution $\tilde{\theta}$ has bias of order $o(n^{-1})$.

Cordeiro & McCullagh (1991) extended this result for GLMs, where $\theta = (\beta, \phi)$. The mean bias correction has the formulae

$$b_\beta = -i_{\beta\beta}^{-1} A_\beta^* \tag{4.1}$$

and

$$b_\phi = -i_{\phi\phi}^{-1} A_\phi^*, \tag{4.2}$$

where $A_\beta^* = X^\top W \xi$, ξ is a p-vector with generic element $\xi_i = \frac{h_i d_i'}{2 d_i w_i}$, h_i is the $i-$th diagonal element of the matrix $H = X(X^\top W X)^{-1} X^\top W$ and $d_i' = \frac{\partial^2 \mu_i}{\partial \eta_i^2}$, $i = 1, ..., n$. For details on A_ϕ^*, see Kosmidis et al. (2020, Section 2.2).

The expression of the explicit bias corrected estimator for β given by Kosmidis et al. (2020) is

$$(X^\top \hat{W} X)^{-1} X^\top \hat{W} (\hat{z} + \hat{\phi}\hat{\xi}),$$

where z is the vector of "working" covariates with generic element $z_i = \eta_i + (y_i - \mu_i)/d_i$ and $\hat{z}$ is its estimator based on MLE. As for section 2.3, the explicit correction can be performed iteratively up to convergence and then replace the covariate z by its correction $z + \phi\xi$. In Poisson models with $g(\cdot) = \log(\cdot)$, $\phi\xi = \frac{h}{2me^{\eta}}$.

Differently from MLE, the generic update of $\hat{\beta}$ depends on ϕ. Hence, when the dispersion parameter in unknown, an estimate of ϕ is necessary to compute the bias corrected estimator for β and the updates of $\hat{\beta}$ and $\hat{\phi}$ need to be simultaneous. However, neither $i_{\phi\phi}^{-1}$ nor A_ϕ^* depend on β hence the bias correction estimator for ϕ requests the knowledge of $\hat{\phi}$ only.

4.3 Mean bias-reduced adjusted score functions

Implicit mean bias reduction methods consist in modifying the score equations by a term A^* to reduce the MLE bias.

For GLMs, Firth (1993) showed that the *adjusted score equations*

$$l_\beta^* + A_\beta^* = 0 \text{ and } l_\phi^* + A_\phi^* = 0 \tag{4.3}$$

deliver estimators $\hat{\beta}_*$ and $\hat{\phi}_*$ with mean bias of smaller asymptotic order than the MLE.

To solve the adjusted score equations, quasi-Fisher scoring algorithm is commonly used. Although there is no theoretical proof of its convergence to the bias-reduced estimator, many empirical studies have shown no evidence of divergence. The generic updates for β and ϕ are

$$\beta^{(j+1)} = \beta^j + [i_{\beta\beta}^{(j)}]^{-1} l_\beta^{*(j)} - b_\beta^j, \tag{4.4}$$

$$\phi^{(j+1)} = \phi^j + [i_{\phi\phi}^{(j)}]^{-1} l_\phi^{*(j)} - b_\phi^j. \tag{4.5}$$

As for the location-adjusted mean bias corrected estimator, the algorithm requires to estimate β and ϕ simultaneously when the dispersion parameter is unknown since b_β generally depends on ϕ.

The invariance property under linear transformations applies to mean BR estimators in terms of the mean bias of the transformed estimators. This property is particularly relevant in categorical regressions interested in inference on arbitrary contrasts. However, mean BR estimators are not invariant under nonlinear transformations on the parameters.

4.4 Median bias-reducing adjusted score functions

Similarly to adjusted score equations to reduce mean bias, Kenne Pagui et al. (2017) proposed to adjust score equations in order to reduce median bias compared to the MLE estimator. The ambition is to obtain an estimator whose probability to underestimate and overestimate the corresponding parameter is as close as possible to 0.5.

For GLMs the adjusted score equations have the following structure

$$l_\beta + A_\beta^\dagger = 0 \quad \text{and} \quad l_\phi + A_\phi^\dagger = 0.$$

According to Kosmidis et al. (2020, chap 2.4) and using its notation, explicit formulae for $A_\beta^\dagger$ and $A_\phi^\dagger$ have the form

$$A_\beta^\dagger = X^\top W(\epsilon + Xu) \quad \text{and} \quad A_\phi^\dagger = \frac{p}{2\phi} + \frac{\sum_{i=1}^n m_i^3 a_i'''}{6\phi^2 + \sum_{i=1}^n m_i^2 a_i''}.$$

As for mean BR estimators, quasi-Fisher scoring algorithm is a valid method to solve the adjusted score equations and the guarantee of its convergence has vast empirical evidence although no theoretical foundation.

The generic updates for β and ϕ are

$$\beta^{(j+1)} = (X^\top W^{(j)} X)^{-1} X^\top W^{(j)} (z^{(j)} + \phi^{(j)} \xi^{(j)}) + \phi^{(j)} u^{(j)}, \tag{4.6}$$

$$\phi^{(j+1)} = \phi^{(j)} \left\{ 1 + \phi^{(j)} \frac{\sum (q_i^{(j)} - \rho_i^{(j)})}{\sum m_i^2 a_i^{''(j)}} + \phi^{(j)} \frac{\sum m_i^3 a_i^{'''(j)}}{3(\sum m_i^2 a_i^{''(j)})^2} + (\phi^{(j)})^2 \frac{p}{\sum m_i^2 a_i^{''(j)}} \right\}. \quad (4.7)$$

For further details, see Kosmidis et al. (2020, Section 2.4). Equation (4.7) reveals that the generic update for β of the median BR can be expressed as a translation by ϕu of the mean BR update for β.

Median BR estimators are invariant under general component-wise transformations of the parameters concerning the median bias reduction improvement, but they are not invariant under linear transformations. For instance, this feature is particularly favorable when a dispersion parameter in unknown and needs to be estimated empirically.

4.5 Simulation experiments

This paragraph will analyze the behavior of the MLE estimator (subsection 1.3.3), the mean BR estimator (section 4.3) and the median BR estimator (section 4.4) through Monte Carlo simulations. The interest will be on the quasi-independence model for contingency tables presented in section 3.4, whose general structure is Equation (3.2) and special attention will be given to δ due to its crucial interpretation in the quasi-independence model for matched-pairs data. We assume the model estimated by the maximum likelihood as the true generating process of the observed data.

The library `brglm2` (Kosmidis, 2020) will be used. The function `brglm_fit` fits a generalized linear model using explicit and implicit bias reduction methods. The `type` argument can be used to specify the type of score adjustment. Among the available options, we will use

- `type` = “`AS_mean`” which implements the mean BR score adjustment in Firth (1993),
- `type` = “`AS_median`” which implements the median BR score adjustment in Kenne Pagui, Salvan & Sartori (2017).

It is possible to apply the function `update` to a `glm` and specify the arguments `method` = “`brglm2`” and `type`, too.

The function `movies_sim` is defined to compute the main steps of the analysis given a dataframe suitable for the quasi-independence model (section 3.4). The input `int` is an integer which the frequencies are divided by; `Nsim` specifies the number of pseudo-samples to simulate; `seme` specifies the seed. Firstly, `movies_sim` fits the MLE model based on the observed data and simulate `Nsim` pseudo-samples. Notice that the sample size is not fixed because of the Poisson distribution assumption. For each pseudo-sample estimates and standard errors are computed from the MLE estimator, the mean BR estimator and the median BR estimator using the library `brglm2`, Kosmidis (2020). Afterwords we compute the sample mean and median among all pseudo-samples by each estimator. The number of iterations of the quasi-Fisher scoring algorithm and the number of times the estimated standard errors are greater than 3 are saved to detect the possibility of divergence.

At first, the true model is fitted on movies data in Table 3.1. As shown in Table 4.1, the mean BR and median BR estimators tend to be more precise than the MLE when estimating the parameters in terms of bias and have lower mean squared error (MSE). The probability of underestimation (PU) of the MLE is farther from 0.5 than that of the mean BR estimator and median BR estimator. The coverage probability is closer to the nominal value 0.95 for the mean BR estimator and median BR estimator than for the MLE.

We expect these results to be more evident when the sample size is smaller hence we analyze the data in Table 3.1 with frequencies divided by 5 and with frequencies divided by 10, rounded to integers in both cases. The sizes of the pseudo-samples will be generally smaller than in the first case because of the different values of the intercepts in the true models. As highlighted in Table 4.2, when frequencies are divided by 5 differences in the behavior between the estimators are more noticeable in terms of bias, probability of underestimation and MSE. The MAE of the three estimators takes close values due to its robustness. Table 4.3 reports the estimators sample properties when the observed frequencies are divided by 10. The bias and the MSE of the MLE are particularly high compared to those of the mean BR and median BR. However, the three estimators have similar behavior in terms of MAE and coverage probability although differences are slightly more noticeable than in the situations with higher sample sizes.

The three estimators may cause different results of the quasi-Fisher scoring algorithm, too. The presence of divergence will be discussed later in this section.

Table 4.1: Movie critics data

Estimation method	Bias	PU	MSE	MAE	Coverage	$se > 3$
MLE	0.0146	0.4705	0.0307	0.1184	0.9441	0
mean BR	0.0022	0.4987	0.0293	0.1161	0.9529	0
median BR	0.0042	0.4948	0.0296	0.1167	0.9509	0

Table 4.2: Movie critics data divided by 5

Estimation method	Bias	PU	MSE	MAE	Coverage	$se > 3$
MLE	0.0947	0.4580	1.1082	0.2725	0.9475	30
mean BR	-0.0013	0.5180	0.1526	0.2508	0.9654	0
median BR	0.0144	0.5046	0.1721	0.2588	0.9586	0

Table 4.3: Movie critics data divided by 10

Estimation method	Bias	PU	MSE	MAE	Coverage	$se > 3$
MLE	2.1205	0.4408	81.1205	0.4723	0.9810	1710
mean BR	0.0088	0.5203	0.4018	0.3798	0.9783	0
median BR	0.0866	0.4990	0.6666	0.4129	0.9766	0

To further investigate the behavior of the mean BR estimator and median BR estimator, rectangular contingency tables are analyzed, too. In particular, the factor `ebert` in section 3.6 has been converted into a dicotomic variable merging the levels `Con` and `Mixed` into a new level `Con`, such as in Table 4.4. The quasi independence model in Equation (3.4) is adapted to rectangular tables. In this case, the parameter δ is still related to a binary variable which equals to 1 if the ordinal variables are on the same level, 0 otherwise, yet only frequencies corresponding to levels existing in both variables may be affected by the influence of rater agreement.

Analogously to square tables, different sample sizes are taken into consideration by dividing the frequencies by a fixed factor. In particular, observed frequencies of Table 4.4 are divided by 5 and by 10, respectively. As expected, estimated biasness and MSE of the mean BR estimator and median BR estimator tend to be lower than those of the MLE estimator, particularly when sample size decreases. The probability of underestimation gradually drifts apart from 0.5 when sample size decreases and specifically for the MLE estimator rather than for the mean BR estimator and median BR estimator.

The presence of divergence of the algorithm when estimating the parameters is not easy to evaluate in R for GLMs. Since it usually causes high standard errors and in the true models the standard errors are small, the number of times when the estimated standard error is higher than 3 has been counted. In all the situations in the analysis, the frequency for the mean BR estimator and the median BR estimator is zero which indicates no presence of divergence. However, the estimated standard errors based on the MLE tend to grow particularly when the sample size increases and this may be a clue of divergence of the algorithm. In addition to high standard errors, the number of iterations may be informative on the effort to estimate the parameters. In particular, it can be noticed that the number of iterations generally increases when sample size decreases. Although it happens that mean BR and median BR estimations require more iterations than MLE, for the MLE the maximum number of iterations is fixed at 25.

Table 4.4: Ratings of movies given by Gene Siskel and Roger Ebert transformed in a rectangular table

	Ebert Rating	
Siskel Rating	Con	Pro
Con	32	13
Mixed	21	11
Pro	19	64

Table 4.5: Movie critics rectangular data

Estimation method	Bias	PU	MSE	MAE	Coverage	$se > 3$
MLE	0.0220	0.4764	0.0488	0.1459	0.9493	0
mean BR	0.0014	0.5129	0.0460	0.1419	0.9599	0
median BR	0.0097	0.4995	0.0471	0.1430	0.9550	0

Table 4.6: Movie critics rectangular data divided by 5

Estimation method	Bias	PU	MSE	MAE	Coverage	$se > 3$
MLE	0.2477	0.4583	3.9299	0.3319	0.9669	90
mean BR	0.0038	0.5335	0.2474	0.3036	0.9784	0
median BR	0.0518	0.4995	0.3091	0.3151	0.9730	0

Table 4.7: Movie critics rectangular data divided by 10

Estimation method	Bias	PU	MSE	MAE	Coverage	$se > 3$
MLE	3.7948	0.4184	110.8098	0.6235	0.9987	2022
mean BR	-0.0050	0.5561	0.6100	0.4694	0.9957	0
median BR	0.1996	0.4889	1.0995	0.5407	0.9954	0

Finally, the distributions of the estimators for δ have been plotted. The histograms are generally coherent with the black line, which follows a normal distribution with mean equal to the true value of δ in each simulation study and sample standard error of the estimates. It is noticeable that the MLE estimator produces outliers when the sample size decreases, while the mean BR estimator and median BR estimator have a tendency to be slightly positively skewed in rectangular tables with small sample sizes.

Further details on the R code can be found in the Appendix .

Conclusions

This work aimed to describe the presence of agreement in contingency tables relying on Agresti (2010) and analyze the studies conducted by Cox & Hinkley (1974), Firth (1993), Kenne Pagui, Salvan & Sartori (2017) and Kosmidis, Kenne Pagui & Sartori (2020) to improve MLE properties in finite samples. In setting our work ground, we recalled inference based on the likelihood and the GLMs, giving deeper details on log-linear models. We then focused on the models for rater agreement and we adapted them to an example on movies critics as proposed by Agresti (2010). The poor performance of the independence model was shown, while the introduction of a parameter δ to describe agreement revealed a great improvement in the model fit. Additionally, we recalled the ordinal agreement model which may express a linear relationship between the ordinal variables.

Furthermore, attention was paid to some improvements of the MLE based on both explicit and implicit methods. Specifically, the location-adjusted mean bias correction (Cox & Hinkley, 1974), the mean BR estimator (Firth, 1993) and median BR estimator (Kenne Pagui, Salvan & Sartori (2017) and Kosmidis, Kenne Pagui & Sartori (2020)) were described as valid methods to reduce the bias of the MLE in samples. Finally, some simulation experiments showed the decrease of the estimator bias and gave further empirical support of the convergence of the quasi-Fisher scoring algorithm when the mean BR estimator and the median BR estimator are adopted.

Appendix

The appendix reports the R chunk developed for the movies example in section 3.6 and to analyze the estimators in section 4.5.

Chapter 3

Contingency table

```
# Rows: Siskel Rating
# Col: Ebert Rating

table_movie <- matrix(c(24, 8, 13,
                     8, 13, 11,
                     10, 9, 64), byrow = TRUE, ncol = 3)

names <- c("Con", "Mixed", "Pro")

colnames(table_movie) <- names
rownames(table_movie) <- names
table_movie
```

```
##       Con Mixed Pro
## Con    24     8  13
## Mixed   8    13  11
## Pro    10     9  64
```

Build up the dataset to be used in models

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
names <- c("Con", "Mixed", "Pro")
siskel <- as.factor(c(names, names, names))
ebert <- as.factor(c("Con", "Con", "Con", "Mixed", "Mixed", "Mixed", "Pro", "Pro", "Pro"))
freq <- as.vector(table_movie)

movies <- tibble(siskel, ebert, freq)

# Create important dummy variables.
# *diag* takes the value 1 if the corresponding *freq* is on the main diagonal
# of the contigency table, 0 otherwise.
diag <- as.integer(movies$siskel == movies$ebert)
# *pro.d* takes the value 1 if both *siskel* and *ebert* are "Pro", 0 otherwise.
pro.d <- as.integer(movies$siskel == movies$ebert & movies$ebert == "Pro")
# *mixed.d* takes the value 1 if both *siskel* and *ebert* are "Mixed", 0 otherwise.
mixed.d <- as.integer(movies$siskel == movies$ebert & movies$ebert == "Mixed")
# *con.d* takes the value 1 if both *siskel* and *ebert* are "Con", 0 otherwise.
con.d <- as.integer(movies$siskel == movies$ebert & movies$ebert == "Con")

movies <- tibble(siskel, ebert, freq, diag, pro.d, mixed.d, con.d)
```

Models

```
# 1. Independence model
model.ind <- glm(freq ~ siskel + ebert, data = movies, family = poisson)
summary(model.ind)
```

```
## 
## Call:
## glm(formula = freq ~ siskel + ebert, family = poisson, data = movies)
## 
## Deviance Residuals: 
##        1         2         3         4         5         6         7         8  
##   3.1068  -0.1391  -2.8284  -0.1520   2.4704  -1.8076  -2.5998  -1.6911  
##        9  
##   2.5590  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)   2.4692     0.1995  12.380  < 2e-16 ***
## siskelMixed  -0.3409     0.2312  -1.474 0.140391    
## siskelPro     0.6122     0.1851   3.307 0.000943 ***
## ebertMixed   -0.3365     0.2390  -1.408 0.159260    
## ebertPro      0.7397     0.1875   3.944 8.02e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for poisson family taken to be 1)
## 
##     Null deviance: 102.215  on 8  degrees of freedom
## Residual deviance:  43.233  on 4  degrees of freedom
## AIC: 93.424
## 
## Number of Fisher Scoring iterations: 5
```

```
fitted(model.ind)
```

```
##       1       2       3       4       5       6       7       8       9 
## 11.8125  8.4000 21.7875  8.4375  6.0000 15.5625 24.7500 17.6000 45.6500
```

```
#2. Quasi-independence model : constant delta

model.delta <- update(model.ind, ~ . + diag)
summary(model.delta)
```

```
## 
## Call:
## glm(formula = freq ~ siskel + ebert + diag, family = poisson, 
##     data = movies)
## 
## Deviance Residuals: 
##         1          2          3          4          5          6          7          8  
## -0.02966   0.67707  -0.48692   0.63654  -0.41933  -0.03218  -0.40679  -0.05824  
##         9  
##   0.21653  
```

```
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.0904     0.1965  10.636  < 2e-16 ***
## siskelMixed  -0.2603     0.2459  -1.059  0.28980
## siskelPro     0.3623     0.2059   1.759  0.07853 .
## ebertMixed   -0.2448     0.2543  -0.963  0.33574
## ebertPro      0.5853     0.2061   2.840  0.00452 **
## diag          1.0937     0.1706   6.412 1.44e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 102.2148  on 8  degrees of freedom
## Residual deviance:   1.4942  on 3  degrees of freedom
## AIC: 53.686
##
## Number of Fisher Scoring iterations: 4
```

```
fitted(model.delta)
```

```
##         1         2         3         4         5         6         7         8
## 24.145587  6.234626 11.619787  6.332054 14.571073  9.096873 14.522359 11.194301
##         9
## 62.283340
```

```
#3. Quasi-independence model : delta based on the ii-th element

model.wdelta <- update(model.ind, ~. + pro.d + con.d + mixed.d)
summary(model.wdelta)
```

```
##
## Call:
## glm(formula = freq ~ siskel + ebert + pro.d + con.d + mixed.d,
##     family = poisson, data = movies)
##
## Deviance Residuals:
##        1         2         3         4         5         6         7         8
##  0.00000   0.03482  -0.03092  -0.03454   0.00000   0.03281   0.02727  -0.02949
##        9
##  0.00000
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.21771    0.39572   5.604 2.09e-08 ***
## siskelMixed -0.15060    0.35064  -0.430  0.66755
## siskelPro    0.09464    0.38394   0.246  0.80530
## ebertMixed  -0.12608    0.37462  -0.337  0.73645
## ebertPro     0.33967    0.37687   0.901  0.36744
## pro.d        1.50687    0.41578   3.624  0.00029 ***
## con.d        0.96035    0.44527   2.157  0.03102 *
## mixed.d      0.62392    0.48302   1.292  0.19645
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1.0221e+02  on 8  degrees of freedom
## Residual deviance: 6.0515e-03  on 1  degrees of freedom
## AIC: 56.198
##
## Number of Fisher Scoring iterations: 3
```

```
fitted(model.wdelta)
```

```
##         1         2         3         4         5         6         7         8
## 24.000000  7.901913 10.098087  8.098087 13.000000  8.901913 12.901913 11.098087
##         9
## 64.000000
```

```
#4. Linear rater agreement model
model.ra <- glm(freq ~ siskel + ebert + as.integer(siskel):as.integer(ebert)
                + diag, data = movies, family = poisson)
summary(model.ra)
```

```
##
## Call:
## glm(formula = freq ~ siskel + ebert + as.integer(siskel):as.integer(ebert) +
##     diag, family = poisson, data = movies)
##
## Deviance Residuals:
##        1         2         3         4         5         6         7         8
## -0.21286   0.44089  -0.03941   0.34357   0.00000  -0.30080   0.03482  -0.34467
##        9
##  0.13298
##
## Coefficients:
##                                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)                            2.1686     0.2136  10.151  < 2e-16 ***
## siskelMixed                           -0.6371     0.4685  -1.360  0.17386
## siskelPro                             -0.4354     0.8707  -0.500  0.61703
## ebertMixed                            -0.6010     0.4553  -1.320  0.18687
## ebertPro                              -0.1951     0.8522  -0.229  0.81888
## diag                                   0.8587     0.2997   2.865  0.00417 **
## as.integer(siskel):as.integer(ebert)   0.1939     0.2055   0.944  0.34531
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 102.21480  on 8  degrees of freedom
## Residual deviance:   0.58746  on 2  degrees of freedom
## AIC: 54.779
##
## Number of Fisher Scoring iterations: 4
```

```
fitted(model.ra)
```

```
##         1         2         3         4         5         6         7         8
## 25.057950  6.816921 10.125129  7.067179 13.000000  9.932821 12.874871 12.183079
##         9
## 62.942050
```

```
tibble(siskel = siskel, ebert = ebert, observed = freq,
       independence = fitted(model.ind),
       ind.delta = fitted(model.delta),
       ind.wdelta = fitted(model.wdelta),
       ordinal.agreement = fitted(model.ra))
```

```
## # A tibble: 9 x 7
##   siskel ebert observed independence ind.delta ind.wdelta ordinal.agreement
##   <fct>  <fct>    <dbl>        <dbl>     <dbl>      <dbl>             <dbl>
## 1 Con    Con         24        11.8      24.1       24.0              25.1
## 2 Mixed  Con          8         8.4       6.23       7.90              6.82
## 3 Pro    Con         10        21.8      11.6       10.1              10.1
## 4 Con    Mixed        8         8.44      6.33       8.10              7.07
## 5 Mixed  Mixed       13         6.00     14.6       13.0              13
## 6 Pro    Mixed        9        15.6       9.10       8.90              9.93
## 7 Con    Pro         13        24.7      14.5       12.9              12.9
## 8 Mixed  Pro         11        17.6      11.2       11.1              12.2
## 9 Pro    Pro         64        45.6      62.3       64                62.9
```

Chapter 4

Improvements of likelihood on the quasi-indep. model with constant delta

```
library(tidyverse)
library(brglm2)
library(ggplot2)
library(gridExtra)
```

```
##
## Caricamento pacchetto: 'gridExtra'

## Il seguente oggetto è mascherato da 'package:dplyr':
##
##     combine
```

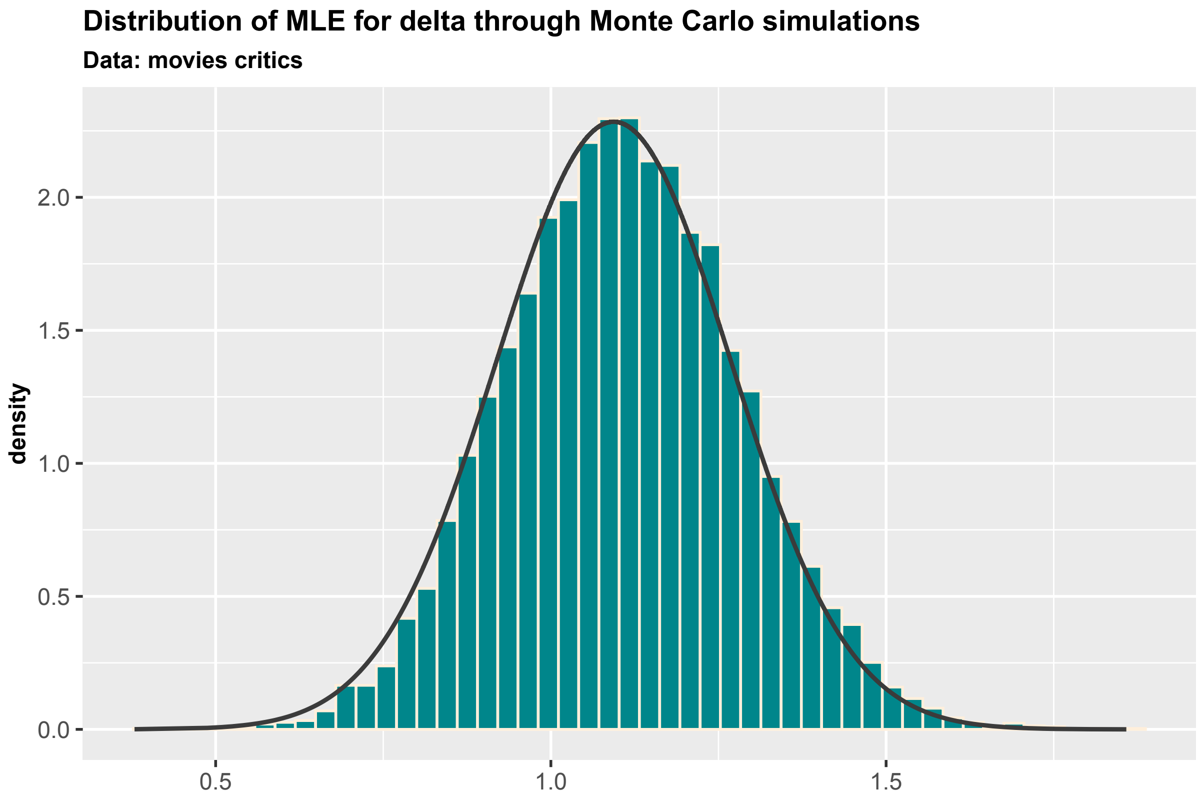
```
movies_sim <- function(dati, int = 1, Nsim = 10000, seme = 123){

    # reduce the frequences by *int*
    movies <- dati %>% mutate(freq = round(freq/int))

    # models based on MLE, mean BR and median BR
    movies.ML <- glm(freq ~ ebert + siskel + diag, family = poisson, data = movies)
    movies.br.m <- update(movies.ML, method = "brglmFit", type = "AS_mean")
    movies.br.med <- update(movies.ML, method = "brglmFit", type = "AS_median")

    # save coefficients and simulate data
    beta <- coef(movies.ML)
    betasd <- summary(movies.ML)$coefficients[,2]
    set.seed(seme)
    simdata <- simulate(movies.ML, nsim = Nsim)

    # define matrix (Nsim x n.coef)
    ml <- br.m <- br.med <- ml.se <- br.m.se <- br.med.se <-
```

```
  matrix(NA, nrow = Nsim, ncol = length(beta))
iter <- matrix(NA, nrow = Nsim, ncol = 3)
noconv <- rep(0,3)

# save estimates of each coef based on each sample
for(i in 1:Nsim){
  current_data <- within(movies, freq <- simdata[[i]])

  ml_fit <- update(movies.ML, data = current_data)
  brm_fit <- update(movies.br.m, data = current_data)
  brmed_fit <- update(movies.br.med, data = current_data)

  ml[i,] <- summary(ml_fit)$coef[,1]
  ml.se[i,] <- summary(ml_fit)$coef[,2]

  br.m[i,] <- summary(brm_fit)$coef[,1]
  br.m.se[i,] <- summary(brm_fit)$coef[,2]

  br.med[i,] <- summary(brmed_fit)$coef[,1]
  br.med.se[i,] <- summary(brmed_fit)$coef[,2]

  iter[i,1] <- ml_fit$iter
  iter[i,2] <- brm_fit$iter
  iter[i,3] <- brmed_fit$iter

  if( max(summary(ml_fit)$coef[,2]) > 5)
    noconv[1] <- noconv[1] + 1
  if( max(summary(brm_fit)$coef[,2]) > 5)
    noconv[2] <- noconv[2] + 1
  if(max(summary(brmed_fit)$coef[,2]) > 5)
    noconv[3] <- noconv[3] + 1
}


# Comparison of the mean of the MLE estimators
ml_mean <- colMeans(ml)
br.m_mean <- colMeans(br.m)
br.med_mean <- colMeans(br.med)

mean_coef <- tibble(coef = beta, MLE = ml_mean, BRmean = br.m_mean,
                    BRmedian = br.med_mean)

# Comparison of the median of the MLE estimators
ml_med <- apply(ml, 2, median)
br.m_med <- apply(br.m, 2, median)
br.med_med <- apply(br.med, 2, median)

med_coef <- tibble(coef = beta, MLE = ml_med, BRmean = br.m_med,
                   BRmedian = br.med_med)

# tibble for coef[6] = delta
tib <- tibble(ml = ml[,length(beta)], ml.se = ml.se[,length(beta)],
              br.m = br.m[,length(beta)], br.m.se = br.m.se[,length(beta)],
                  br.med = br.med[,length(beta)], br.med.se = br.med.se[,length(beta)])

    return(list(dati = movies, mod1 = movies.ML, mod2 = movies.br.m, mod3 = movies.br.med,
                beta = beta, betasd = betasd, ml = ml, br.m = br.m, br.med = br.med,
                ml.se = ml.se, br.m.se = br.m.se, br.med.se = br.med.se,
                mean_coef = mean_coef, med_coef = med_coef, delta = tib,
                iter = iter, noconv = noconv))

}

# Estimates for the model based on *movies*
obj1 <- movies_sim(movies)

obj1$delta
```

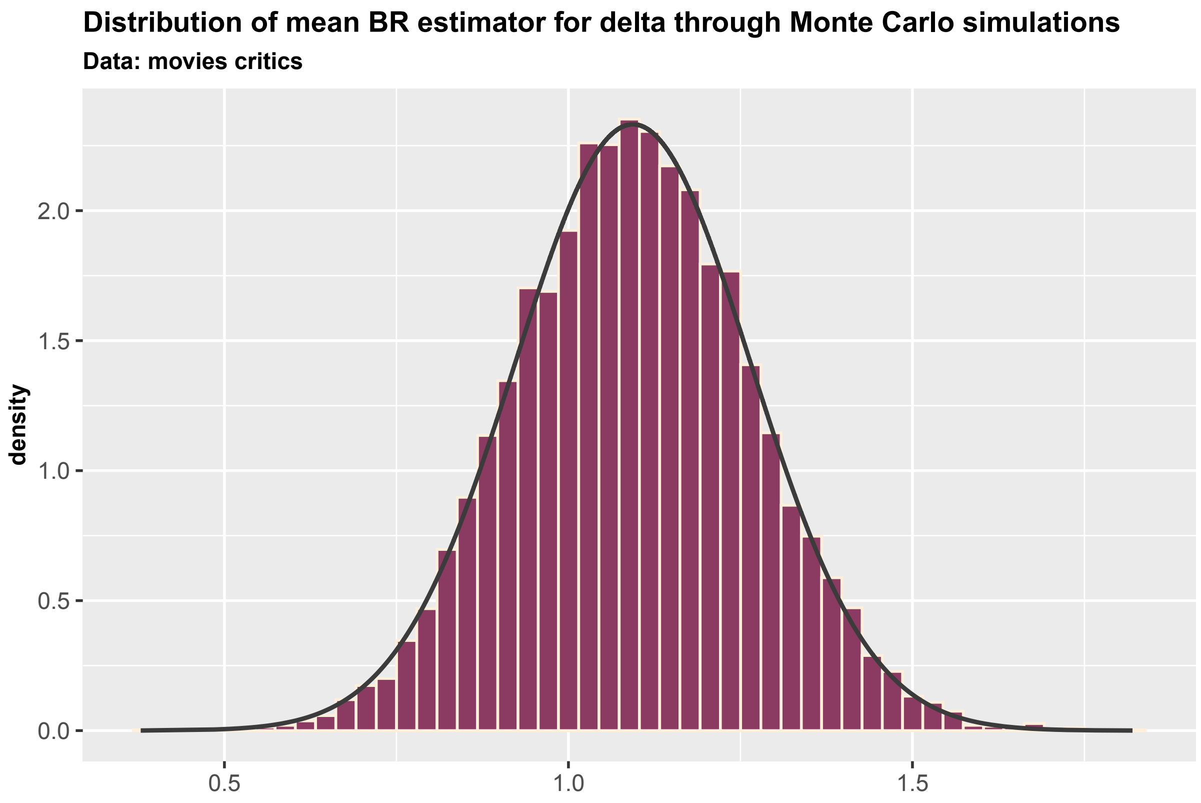
```
## # A tibble: 10,000 x 6
##       ml ml.se  br.m br.m.se br.med br.med.se
##    <dbl> <dbl> <dbl>   <dbl>  <dbl>     <dbl>
##  1 0.833 0.162 0.827   0.161  0.827     0.161
##  2 0.935 0.168 0.926   0.166  0.927     0.166
##  3 1.18  0.192 1.17    0.189  1.17      0.189
##  4 1.25  0.176 1.24    0.173  1.24      0.174
##  5 1.26  0.170 1.25    0.168  1.25      0.168
##  6 0.939 0.166 0.930   0.164  0.931     0.165
##  7 1.18  0.173 1.17    0.171  1.17      0.171
##  8 1.33  0.185 1.32    0.182  1.32      0.183
##  9 1.05  0.171 1.04    0.169  1.04      0.170
## 10 1.31  0.185 1.30    0.183  1.30      0.183
## # ... with 9,990 more rows
```

```
p1 <- ggplot(obj1$delta, aes(x = ml)) +
  geom_histogram(aes(y = ..density..), bins = 50,  fill = "turquoise4", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(ml, mean = as.numeric(obj1$beta[6]), sd(ml))),
            size = 0.8, color = "grey23") +
  labs(title = "Distribution of MLE for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics") +
  theme(title = element_text(size = 9, face = "bold"))

p2 <- ggplot(obj1$delta, aes(x = br.m)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "hotpink4", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(br.m, obj1$beta[6], sd(br.m))),
            size = 0.8, color = "grey23") +
  labs(title = "Distribution of mean BR estimator for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics") +
  theme(title = element_text(size = 9, face = "bold"))

p3 <- ggplot(obj1$delta, aes(x = br.med)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "darkgoldenrod3", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(br.med, obj1$beta[6], sd(br.med))),
            size = 0.8, color = "grey23") +
  labs(title = "Distribution of median BR estimator for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics") +
  theme(title = element_text(size = 9, face = "bold"))
```

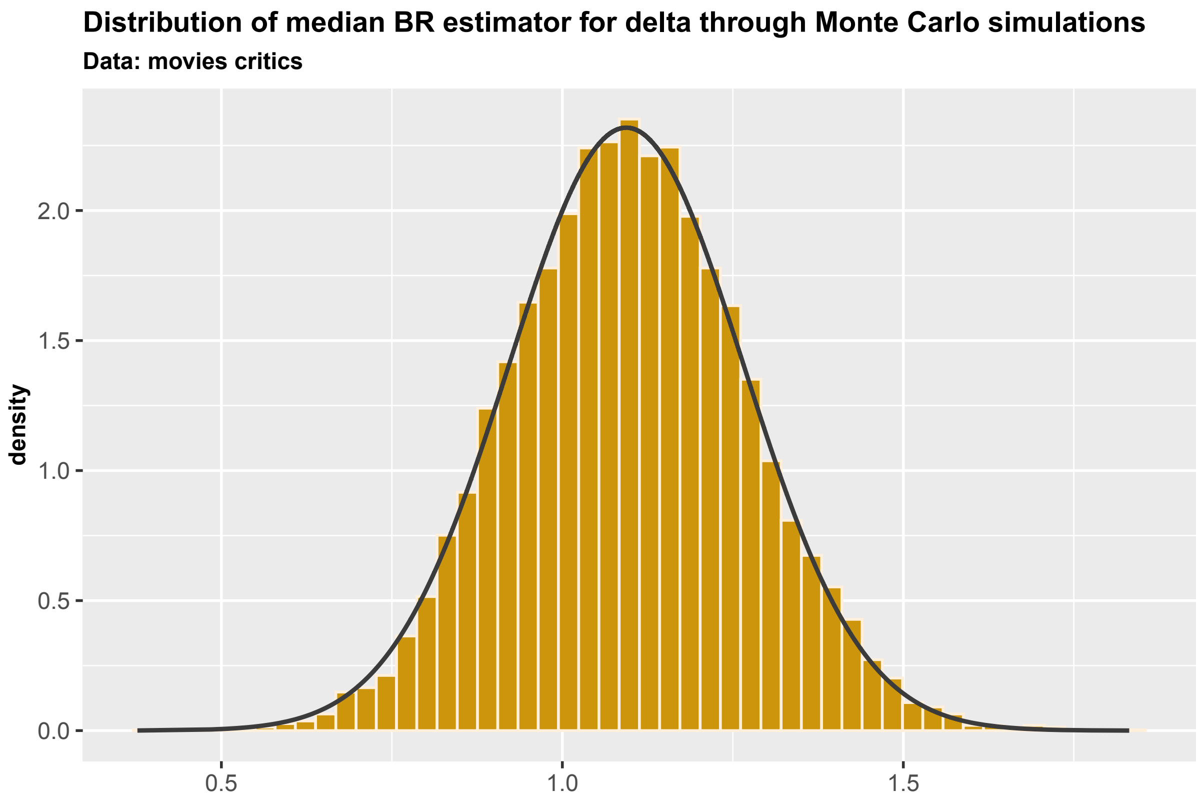
```
p1
```


Distribution of MLE for delta through Monte Carlo simulations
Data: movies critics
density
2.0
1.5
1.0
0.5
0.0
0.5
1.0
1.5

```
p2
```


Distribution of mean BR estimator for delta through Monte Carlo simulations
Data: movies critics
density
0.0
0.5
1.0
1.5
2.0
0.5
1.0
1.5

```
p3
```

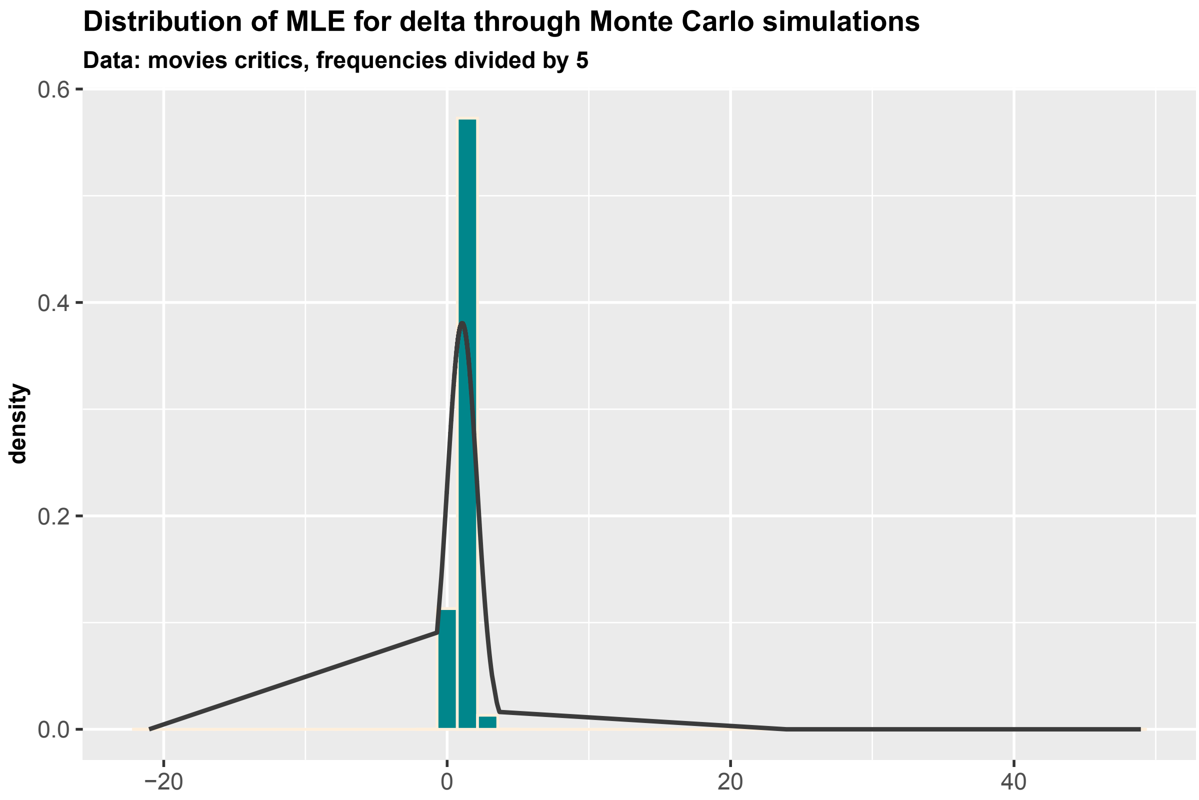


```
# Estimates for the model based on *movies* with *freq* divided by 5

obj2 <- movies_sim(movies, 5)

obj2$delta
```

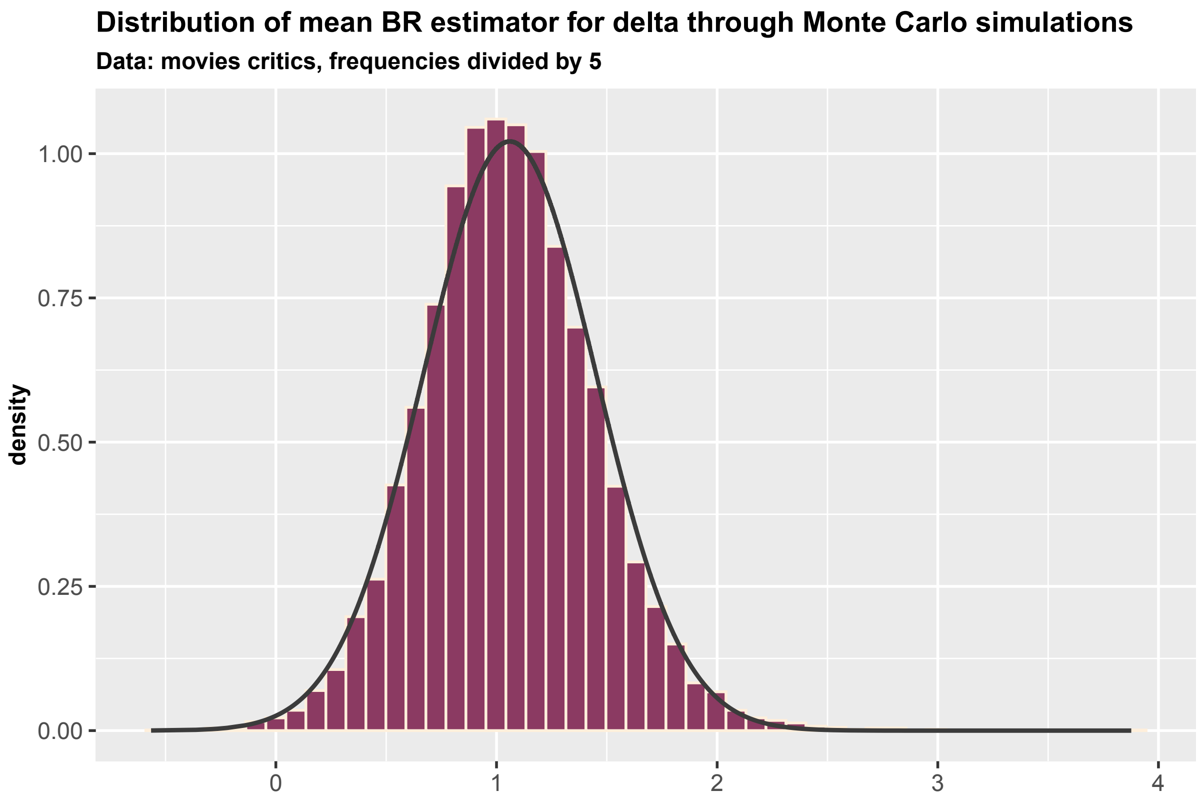
```
## # A tibble: 10,000 x 6
##       ml ml.se  br.m br.m.se br.med br.med.se
##    <dbl> <dbl> <dbl>   <dbl>  <dbl>     <dbl>
##  1 1.19  0.368 1.13    0.347  1.14      0.351
##  2 1.26  0.436 1.18    0.401  1.20      0.411
##  3 0.753 0.357 0.724   0.340  0.727     0.344
##  4 1.68  0.467 1.54    0.424  1.57      0.432
##  5 1.71  0.571 1.52    0.490  1.58      0.512
##  6 0.828 0.415 0.787   0.390  0.789     0.394
##  7 1.05  0.321 1.01    0.308  1.02      0.311
##  8 1.31  0.445 1.22    0.409  1.24      0.417
##  9 0.874 0.363 0.838   0.345  0.842     0.349
## 10 1.02  0.405 0.971   0.379  0.976     0.385
## # ... with 9,990 more rows
```

```
p4 <- ggplot(obj2$delta, aes(x = ml)) +
  geom_histogram(aes(y = ..density..), bins = 50,  fill = "turquoise4", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(ml, obj2$beta[6], sd(ml))), size = 0.8, color = "grey23") +
  labs(title = "Distribution of MLE for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics, frequencies divided by 5") +
  theme(title = element_text(size = 9, face = "bold"))
```

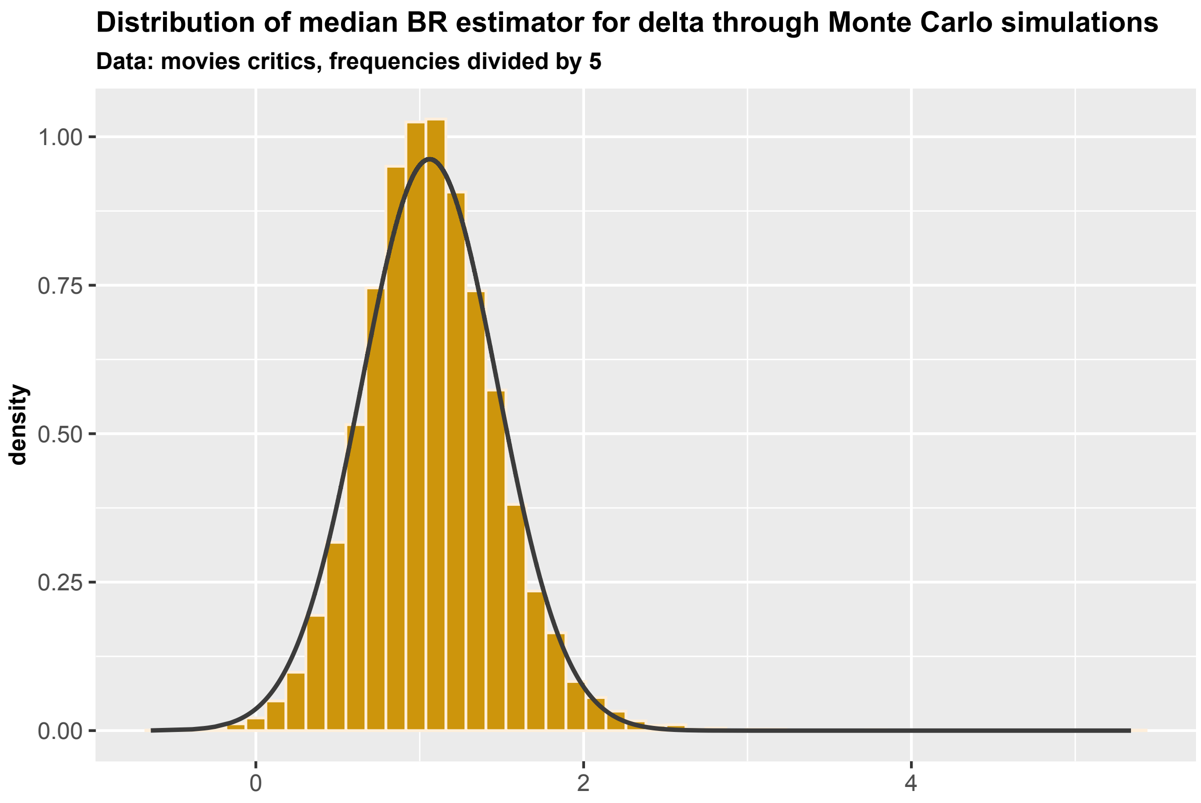
```
p5 <- ggplot(obj2$delta, aes(x = br.m)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "hotpink4", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(br.m, obj2$beta[6], sd(br.m))), size = 0.8, color = "grey23") +
  labs(title = "Distribution of mean BR estimator for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics, frequencies divided by 5") +
  theme(title = element_text(size = 9, face = "bold"))

p6 <- ggplot(obj2$delta, aes(x = br.med)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "darkgoldenrod3", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(br.med, obj2$beta[6], sd(br.med))), size = 0.8, color = "grey23") +
  labs(title = "Distribution of median BR estimator for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics, frequencies divided by 5") +
  theme(title = element_text(size = 9, face = "bold"))

p4
```



```
p5
```


Distribution of mean BR estimator for delta through Monte Carlo simulations
Data: movies critics, frequencies divided by 5
density
1.00
0.75
0.50
0.25
0.00
0
1
2
3
4

```
p6
```

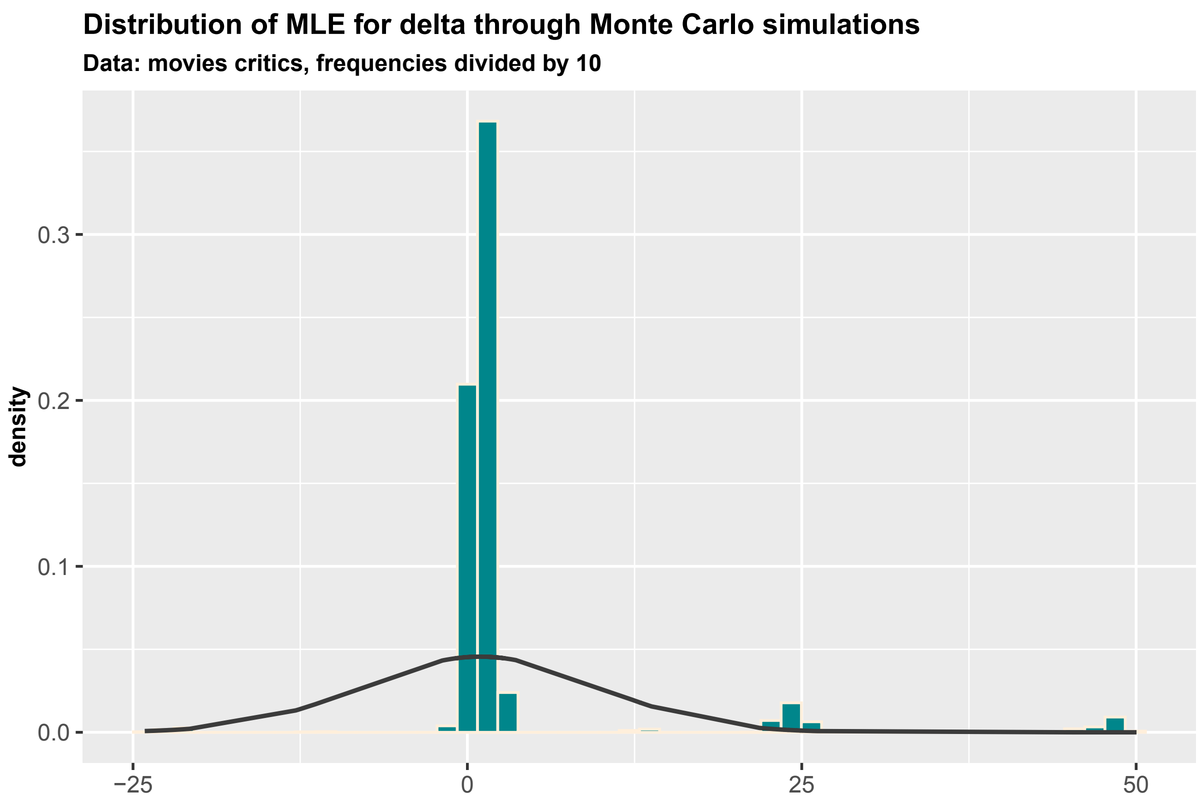


```
# Estimates for the model based on *movies* with *freq* divided by 10

obj3 <- movies_sim(movies, 10)

obj3$delta
```

```
## # A tibble: 10,000 x 6
##       ml ml.se  br.m br.m.se br.med br.med.se
##    <dbl> <dbl> <dbl>   <dbl>  <dbl>     <dbl>
##  1 1.02  0.522 0.931   0.471  0.943     0.479
##  2 0.300 0.624 0.302   0.549  0.284     0.570
##  3 0.229 0.473 0.240   0.437  0.226     0.447
##  4 2.44  1.08  1.83    0.729  2.06      0.827
##  5 0.818 0.681 0.727   0.583  0.728     0.610
##  6 1.43  0.760 1.17    0.606  1.23      0.625
##  7 2.35  1.02  1.79    0.689  1.98      0.769
##  8 0.619 0.515 0.589   0.469  0.583     0.478
##  9 1.66  0.766 1.35    0.608  1.41      0.638
## 10 0.133 0.512 0.156   0.468  0.137     0.482
## # ... with 9,990 more rows
```

```
p7 <- ggplot(obj3$delta, aes(x = ml)) +
  geom_histogram(aes(y = ..density..), bins = 50,  fill = "turquoise4", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(ml, obj3$beta[6], sd(ml))), size = 0.8, color = "grey23") +
  labs(title = "Distribution of MLE for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics, frequencies divided by 10") +
  theme(title = element_text(size = 9, face = "bold"))
```

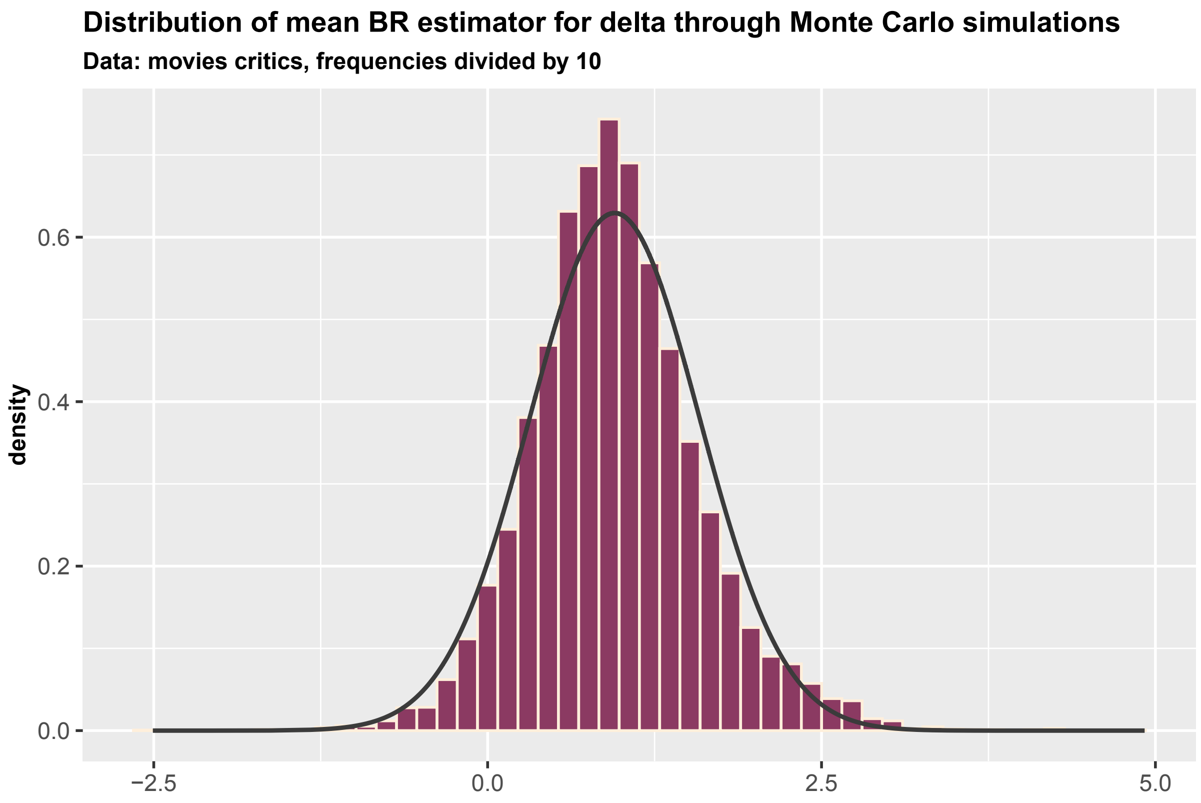
```
p8 <- ggplot(obj3$delta, aes(x = br.m)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "hotpink4", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(br.m, obj3$beta[6], sd(br.m))), size = 0.8, color = "grey23") +
  labs(title = "Distribution of mean BR estimator for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics, frequencies divided by 10") +
  theme(title = element_text(size = 9, face = "bold"))

p9 <- ggplot(obj3$delta, aes(x = br.med)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "darkgoldenrod3", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(br.med, obj3$beta[6], sd(br.med))), size = 0.8, color = "grey23") +
  labs(title = "Distribution of median BR estimator for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics, frequencies divided by 10") +
  theme(title = element_text(size = 9, face = "bold"))

p7
```



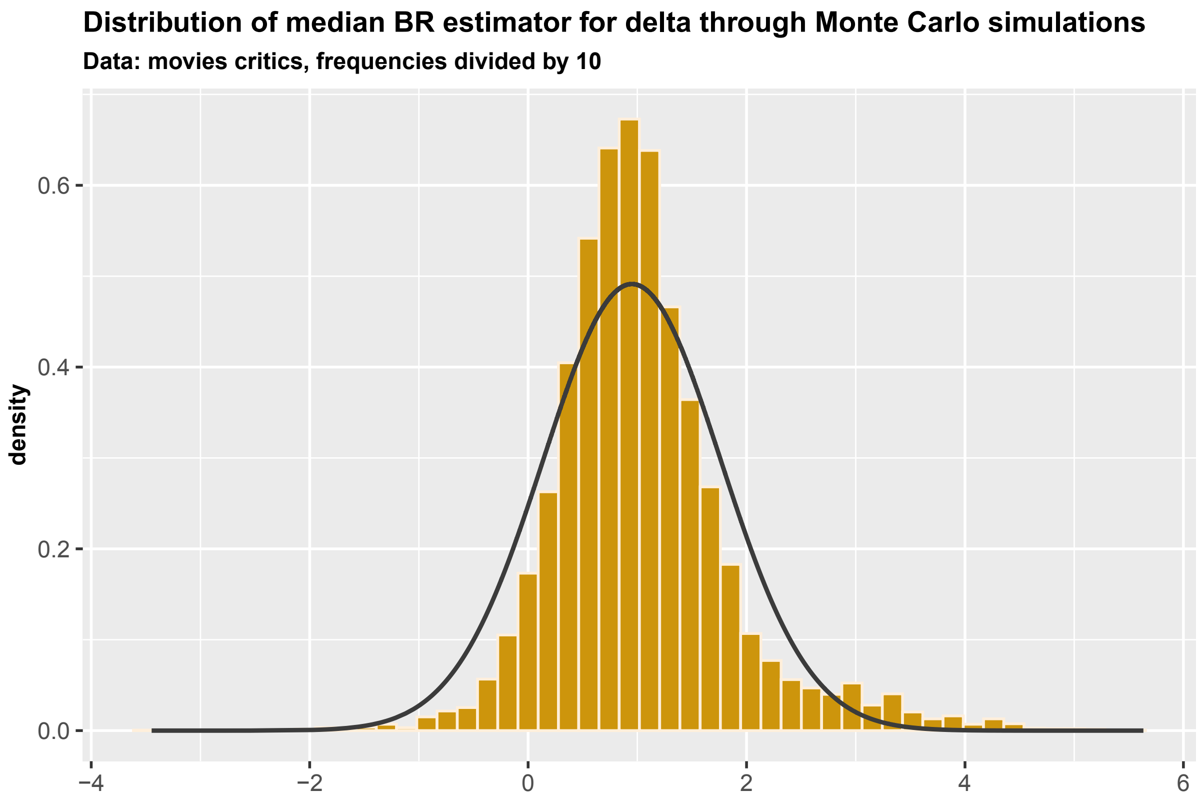
Distribution of MLE for delta through Monte Carlo simulations

Data: movies critics, frequencies divided by 10

```
p8
```


Distribution of mean BR estimator for delta through Monte Carlo simulations
Data: movies critics, frequencies divided by 10
density
0.0
0.2
0.4
0.6
−2.5
0.0
2.5
5.0

```
p9
```


Distribution of median BR estimator for delta through Monte Carlo simulations

Data: movies critics, frequencies divided by 10

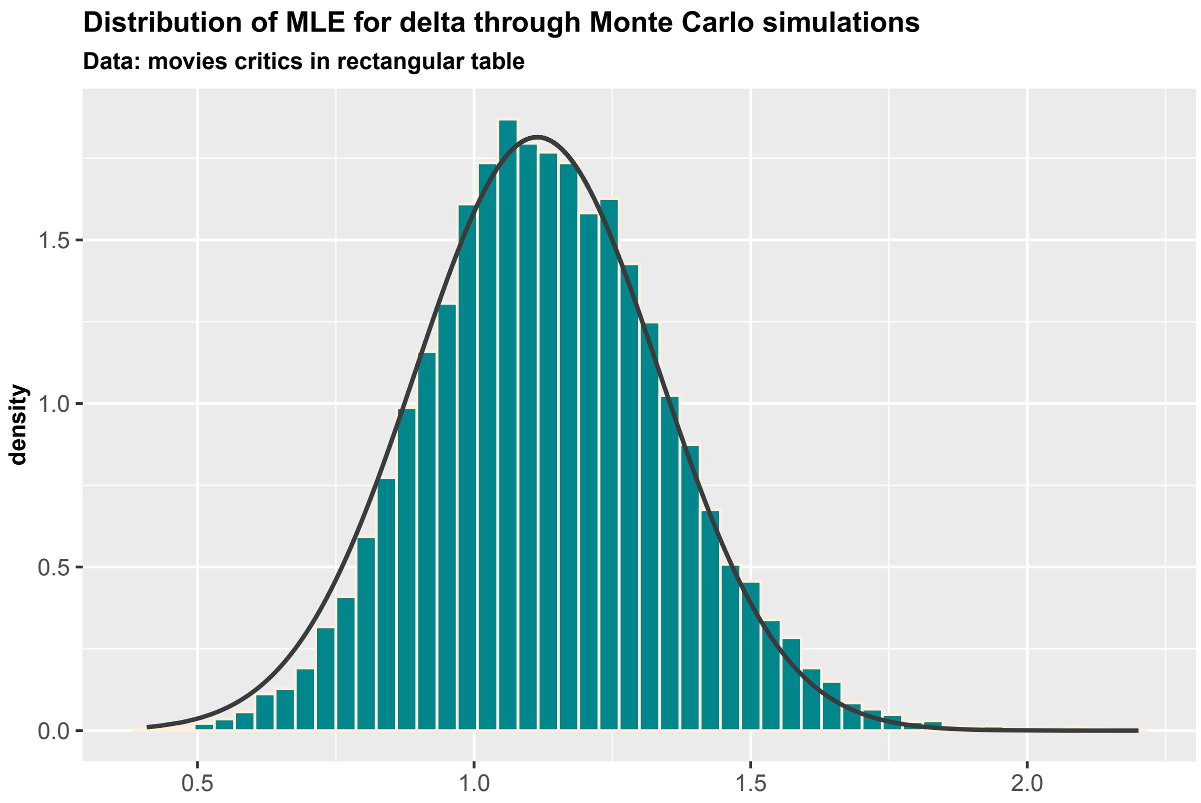
```
# Let *ebert* be a binary variable. Column *mixed* has been incorporated in column *Con*

movies32 <- movies
movies32$freq[movies32$ebert == "Con"] <-
  movies32$freq[movies32$ebert == "Con"] + movies32$freq[movies32$ebert == "Mixed"]
movies32 <- movies32[movies32$ebert != "Mixed",]
movies32
```

```
## # A tibble: 6 x 7
##   siskel ebert  freq  diag pro.d mixed.d con.d
##   <fct>  <fct> <dbl> <int> <int>   <int> <int>
## 1 Con    Con      32     1     0       0     1
## 2 Mixed  Con      21     0     0       0     0
## 3 Pro    Con      19     0     0       0     0
## 4 Con    Pro      13     0     0       0     0
## 5 Mixed  Pro      11     0     0       0     0
## 6 Pro    Pro      64     1     1       0     0
```

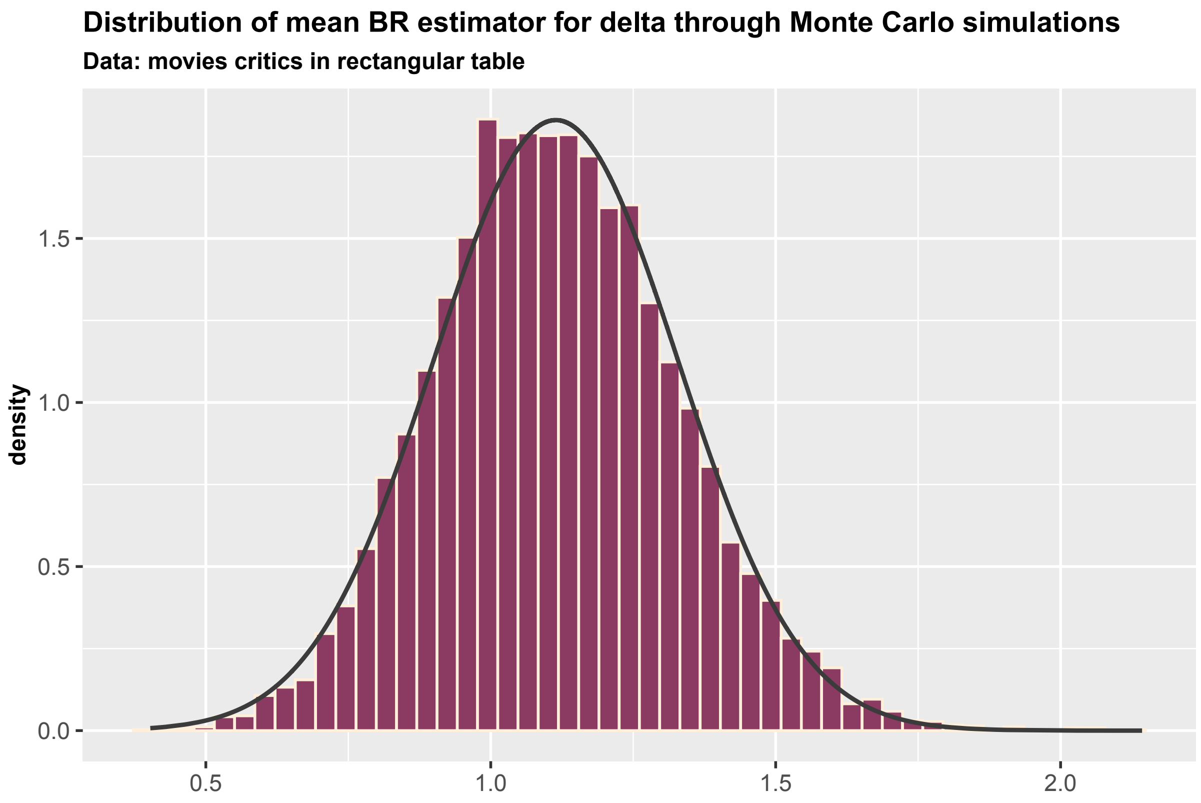
```
obj32 <- movies_sim(movies32, 1)

obj32$delta
```

```
## # A tibble: 10,000 x 6
##       ml ml.se  br.m br.m.se br.med br.med.se
##    <dbl> <dbl> <dbl>   <dbl>  <dbl>     <dbl>
##  1 1.34  0.232 1.31    0.228  1.32      0.229
##  2 0.811 0.207 0.798   0.205  0.803     0.206
##  3 0.983 0.218 0.964   0.215  0.972     0.216
##  4 0.930 0.214 0.912   0.211  0.920     0.212
##  5 0.932 0.199 0.918   0.196  0.923     0.197
##  6 1.36  0.244 1.33    0.239  1.35      0.241
##  7 1.43  0.230 1.40    0.226  1.41      0.227
##  8 0.889 0.209 0.873   0.207  0.880     0.207
##  9 0.990 0.224 0.970   0.220  0.979     0.221
## 10 1.03  0.203 1.01    0.201  1.02      0.201
## # ... with 9,990 more rows
```

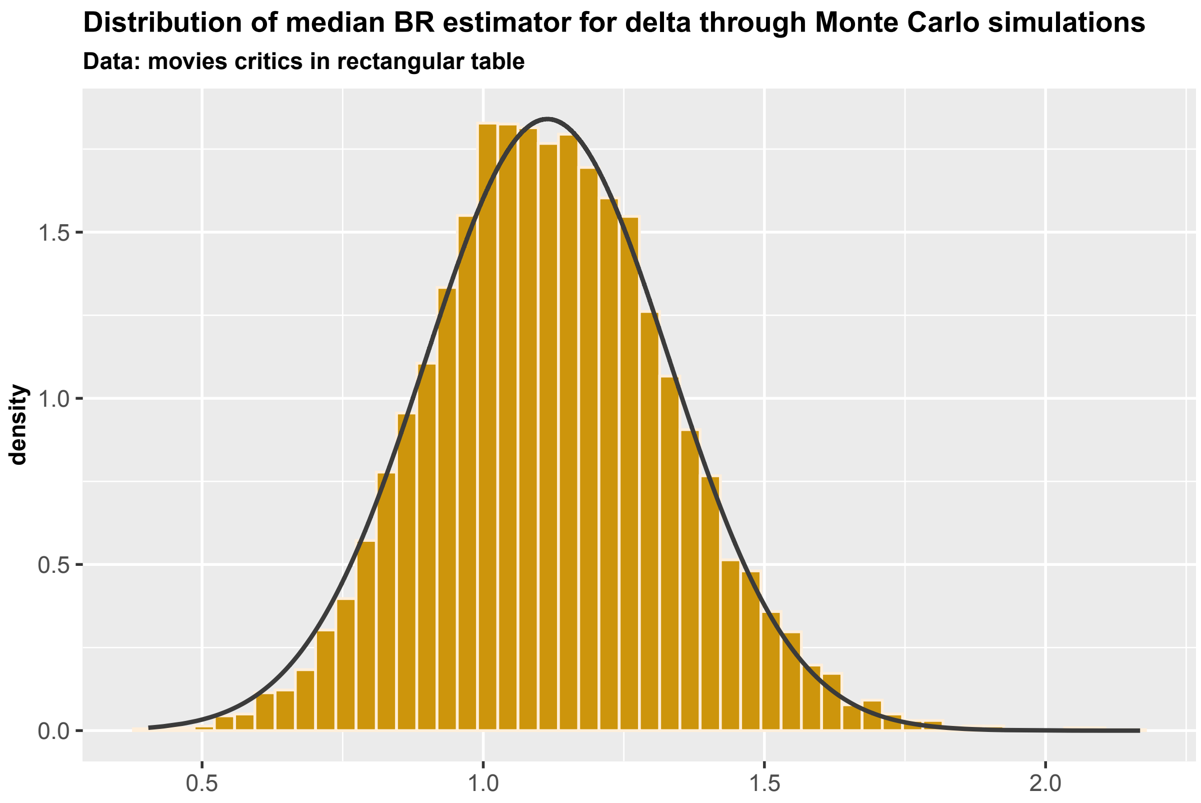
```
p10 <- ggplot(obj32$delta, aes(x = ml)) +
  geom_histogram(aes(y = ..density..), bins = 50,  fill = "turquoise4", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(ml, obj32$beta[5], sd(ml))), size = 0.8, color = "grey23") +
  labs(title = "Distribution of MLE for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics in rectangular table") +
  theme(title = element_text(size = 9, face = "bold"))

p11 <- ggplot(obj32$delta, aes(x = br.m)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "hotpink4", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(br.m, obj32$beta[5], sd(br.m))), size = 0.8, color = "grey23") +
  labs(title = "Distribution of mean BR estimator for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics in rectangular table") +
  theme(title = element_text(size = 9, face = "bold"))

p12 <- ggplot(obj32$delta, aes(x = br.med)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "darkgoldenrod3", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(br.med, obj32$beta[5], sd(br.med))), size = 0.8, color = "grey23") +
  labs(title = "Distribution of median BR estimator for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics in rectangular table") +
  theme(title = element_text(size = 9, face = "bold"))

p10
```


Distribution of MLE for delta through Monte Carlo simulations
Data: movies critics in rectangular table
density
1.5
1.0
0.5
0.0
0.5
1.0
1.5
2.0

```
p11
```


Distribution of mean BR estimator for delta through Monte Carlo simulations
Data: movies critics in rectangular table
density
0.0
0.5
1.0
1.5
0.5
1.0
1.5
2.0

```
p12
```



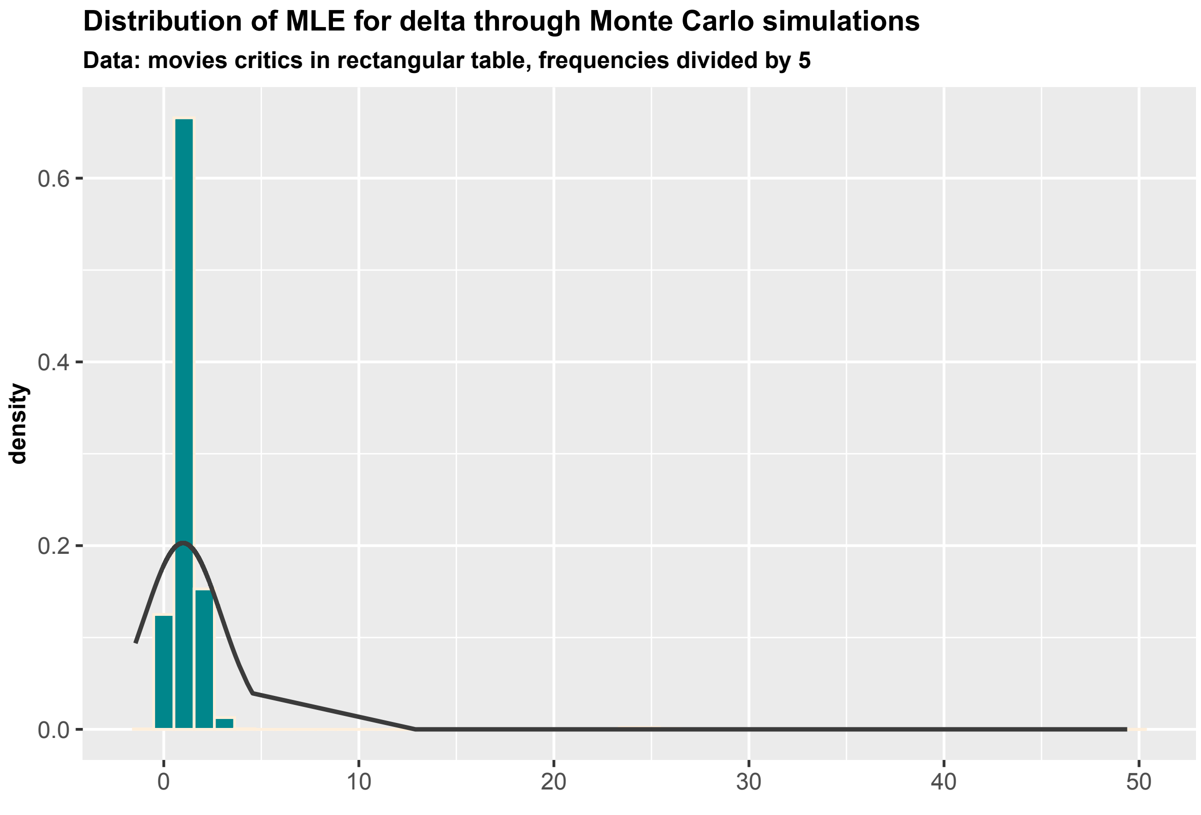
Distribution of median BR estimator for delta through Monte Carlo simulations

Data: movies critics in rectangular table

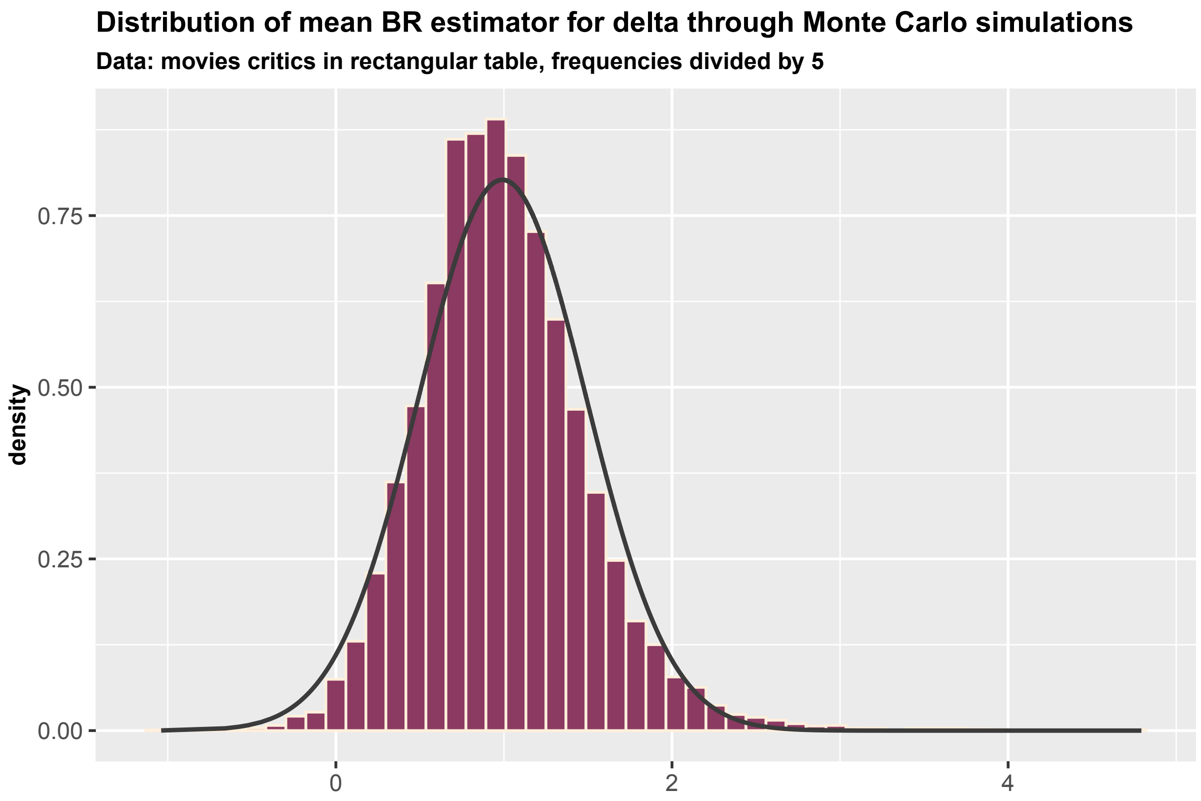
```
# Let *ebert* be a binary variable. Column *mixed* has been incorporated in column *Con*.
# Frequencies divided by 5.

obj32b <- movies_sim(movies32, 5)

obj32b$delta
```

```
## # A tibble: 10,000 x 6
##         ml ml.se    br.m br.m.se  br.med br.med.se
##      <dbl> <dbl>   <dbl>   <dbl>   <dbl>     <dbl>
##  1  0.311  0.467  0.286    0.445  0.295      0.449
##  2  0.783  0.427  0.726    0.405  0.749      0.411
##  3  2.27   0.665  2.03     0.585  2.12       0.611
##  4  1.01   0.447  0.942    0.423  0.972      0.430
##  5 -0.0157 0.477 -0.0186   0.453 -0.0165     0.459
##  6  1.34   0.622  1.17     0.557  1.24       0.575
##  7  1.17   0.530  1.07     0.491  1.10       0.499
##  8  1.03   0.521  0.929    0.486  0.965      0.495
##  9  1.63   0.564  1.47     0.515  1.53       0.529
## 10  1.10   0.440  1.02     0.416  1.05       0.423
## # ... with 9,990 more rows
```

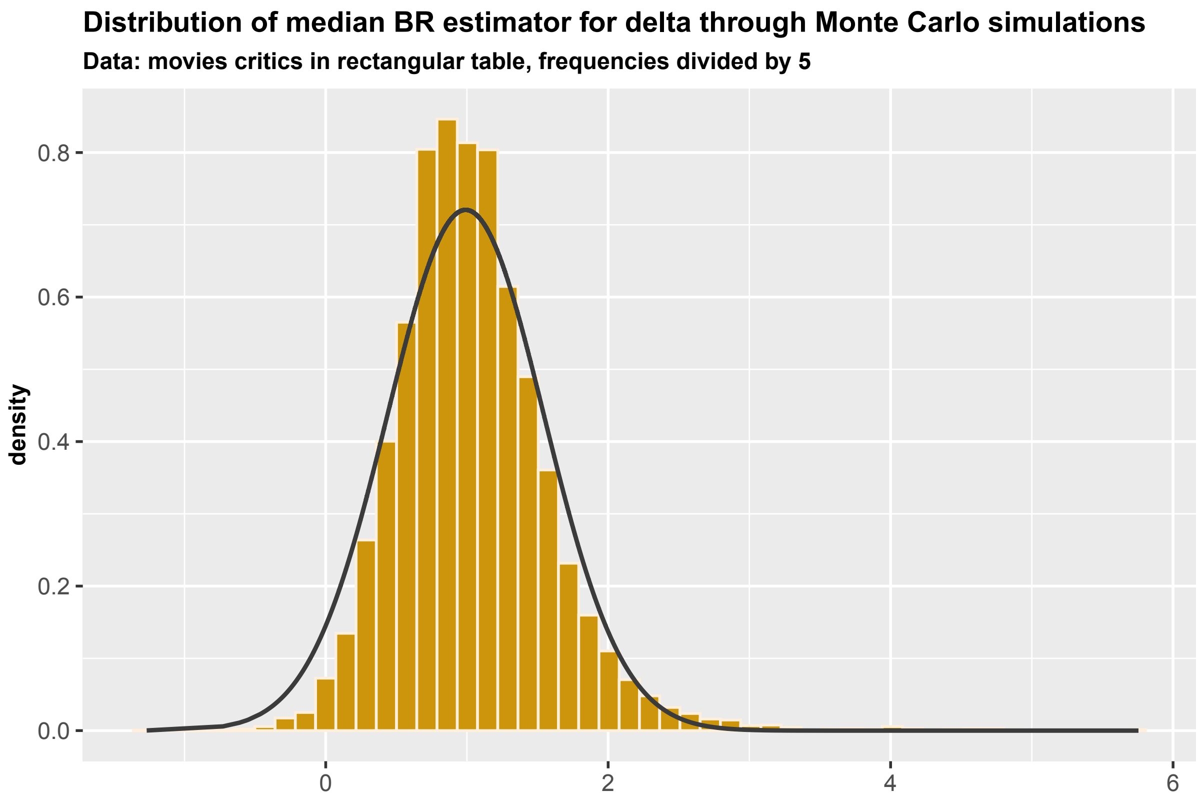
```
p13 <- ggplot(obj32b$delta, aes(x = ml)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "turquoise4", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(ml, obj32b$beta[5], sd(ml))), size = 0.8, color = "grey23") +
  labs(title = "Distribution of MLE for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics in rectangular table, frequencies divided by 5") +
  theme(title = element_text(size = 9, face = "bold"))

p14 <- ggplot(obj32b$delta, aes(x = br.m)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "hotpink4", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(br.m, obj32b$beta[5], sd(br.m))), size = 0.8, color = "grey23") +
  labs(title = "Distribution of mean BR estimator for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics in rectangular table, frequencies divided by 5") +
  theme(title = element_text(size = 9, face = "bold"))

p15 <- ggplot(obj32b$delta, aes(x = br.med)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "darkgoldenrod3", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(br.med, obj32b$beta[5], sd(br.med))), size = 0.8, color = "grey23") +
  labs(title = "Distribution of median BR estimator for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics in rectangular table, frequencies divided by 5") +
  theme(title = element_text(size = 9, face = "bold"))


p13
```



```
p14
```


Distribution of mean BR estimator for delta through Monte Carlo simulations
Data: movies critics in rectangular table, frequencies divided by 5
density
0.75
0.50
0.25
0.00
0
2
4

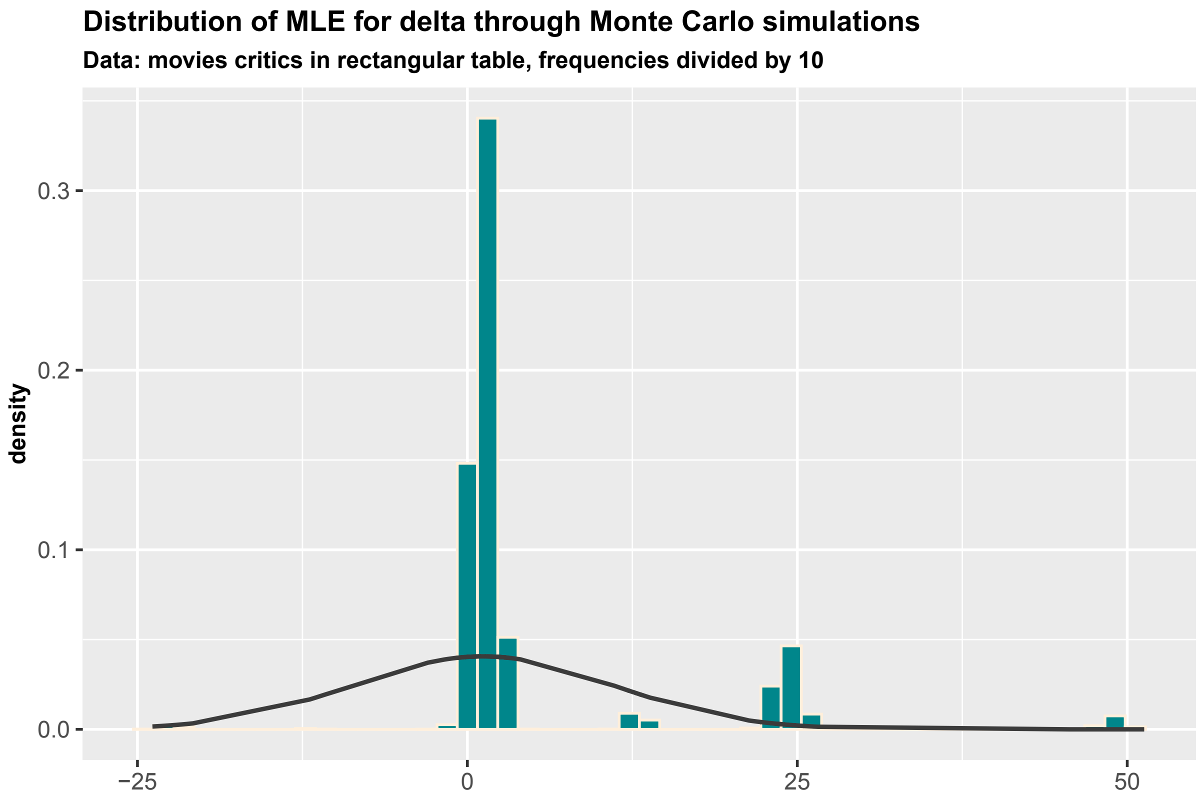
```
p15
```



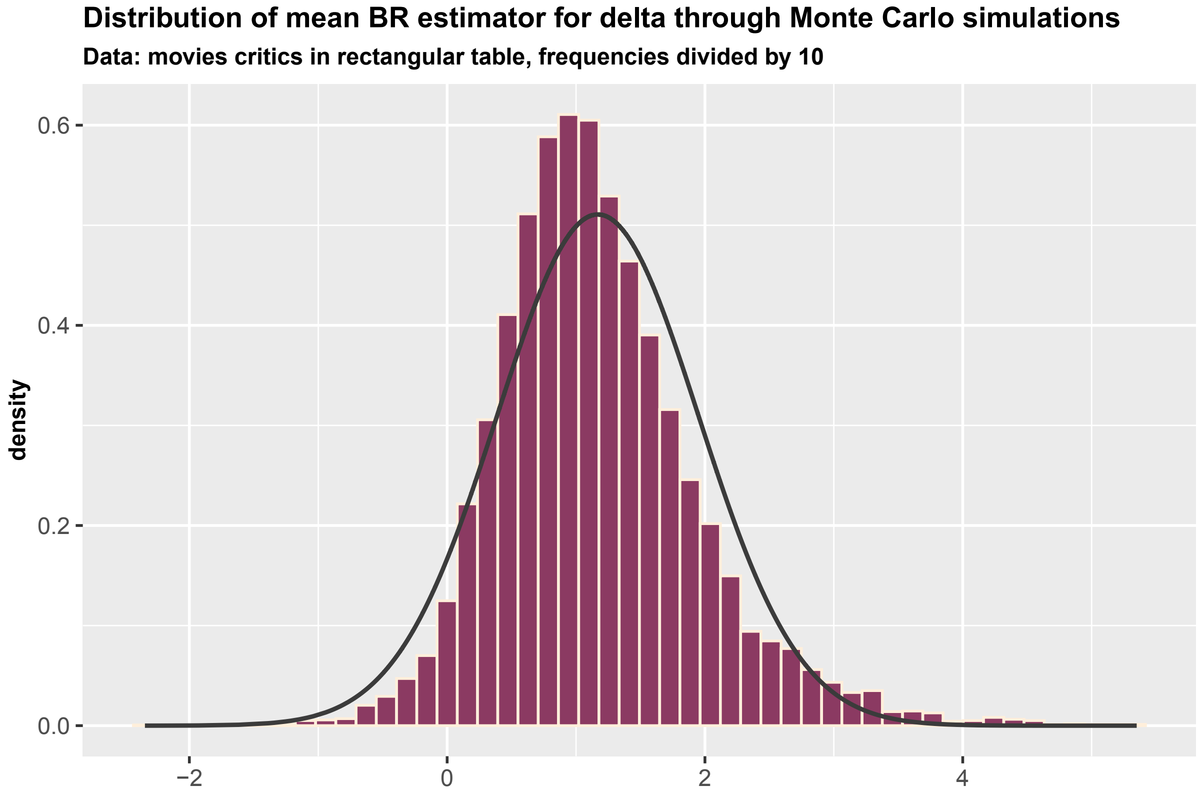
```
# Let *ebert* be a binary variable. Column *mixed* has been incorporated in column *Con*
# Frequencies divided by 10.

obj32c <- movies_sim(movies32, 10)

obj32c$delta
```

```
## # A tibble: 10,000 x 6
##           ml     ml.se      br.m br.m.se   br.med br.med.se
##        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
##  1 -1.29e-16     0.707 -5.34e-17   0.639 3.97e-17     0.653
##  2  9.59e- 1     0.688  8.22e- 1   0.618 8.75e- 1     0.635
##  3  1.10e+ 0     0.906  8.46e- 1   0.742 9.31e- 1     0.779
##  4  1.21e+ 0     0.662  1.06e+ 0   0.588 1.12e+ 0     0.612
##  5  9.07e- 1     0.724  7.47e- 1   0.637 8.04e- 1     0.658
##  6  1.51e+ 0     0.721  1.26e+ 0   0.609 1.36e+ 0     0.647
##  7  2.42e+ 1 49916.     2.04e+ 0   1.03  2.79e+ 0     1.71
##  8  2.42e+ 1 48441.     1.95e+ 0   1.05  2.75e+ 0     1.79
##  9  1.63e+ 0     1.11   1.20e+ 0   0.855 1.37e+ 0     0.920
## 10  1.36e+ 0     0.814  1.11e+ 0   0.693 1.20e+ 0     0.720
## # ... with 9,990 more rows
```

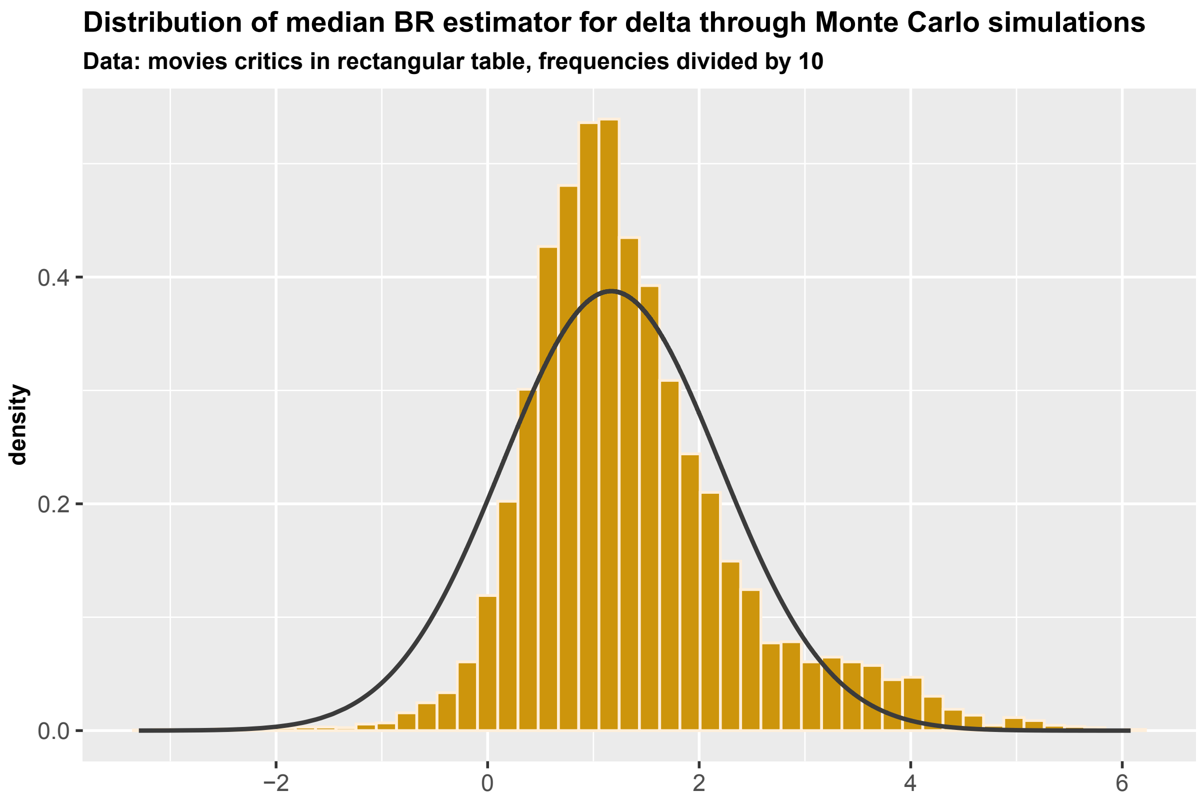
```
p16 <- ggplot(obj32c$delta, aes(x = ml)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "turquoise4", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(ml, obj32c$beta[5], sd(ml))), size = 0.8, color = "grey23") +
  labs(title = "Distribution of MLE for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics in rectangular table, frequencies divided by 10") +
  theme(title = element_text(size = 9, face = "bold"))

p17 <- ggplot(obj32c$delta, aes(x = br.m)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "hotpink4", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(br.m, obj32c$beta[5], sd(br.m))), size = 0.8, color = "grey23") +
  labs(title = "Distribution of mean BR estimator for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics in rectangular table, frequencies divided by 10") +
  theme(title = element_text(size = 9, face = "bold"))

p18 <- ggplot(obj32c$delta, aes(x = br.med)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "darkgoldenrod3", color = "antiquewhite1") +
  geom_line(aes(y = dnorm(br.med, obj32c$beta[5], sd(br.med))), size = 0.8, color = "grey23") +
  labs(title = "Distribution of median BR estimator for delta through Monte Carlo simulations",
       x = "", subtitle = "Data: movies critics in rectangular table, frequencies divided by 10") +
  theme(title = element_text(size = 9, face = "bold"))

p16
```



```
p17
```


Distribution of mean BR estimator for delta through Monte Carlo simulations
Data: movies critics in rectangular table, frequencies divided by 10
density
0.6
0.4
0.2
0.0
−2
0
2
4

```
p18
```



```
# Analysis

analysis <- function(object){

    # Biasness
    bias <- apply(object$mean_coef, 2, '-', object$beta)

    ## Coverage probability
    # proportion of (beta_hat - beta) / se(beta) <= z(1-alpha/2)
      cp.ml <- ((object$delta$ml - object$beta[length(object$beta)]) / object$delta$ml.se
                <= qnorm(0.95)) %>% mean
      cp.brm <- ((object$delta$br.m - object$beta[length(object$beta)]) / object$delta$br.m.se
                  <= qnorm(0.95)) %>% mean
      cp.brmed <- ((object$delta$br.med - object$beta[length(object$beta)]) /
                     object$delta$br.med.se
                    <= qnorm(0.95)) %>% mean

    ## Probability of underestimation
      pu.ml <- ((object$delta$ml - object$beta[length(object$beta)]) /
                  object$delta$ml.se <= 0) %>% mean
      pu.brm <- ((object$delta$br.m - object$beta[length(object$beta)]) /
                   object$delta$br.m.se <= 0) %>% mean
      pu.brmed <- ((object$delta$br.med - object$beta[length(object$beta)]) /
                     object$delta$br.med.se <= 0) %>% mean

    ## MSE
```

```
      mse.ml <- (object$delta$ml - object$beta[length(object$beta)])^2 %>% mean
      mse.br <- (object$delta$br.m- object$beta[length(object$beta)])^2 %>% mean
      mse.brmed <- (object$delta$br.med - object$beta[length(object$beta)])^2 %>% mean


    ## MAE
      mae.ml <- abs(object$delta$ml - object$beta[length(object$beta)]) %>%
        median
      mae.brm <- abs(object$delta$br.m- object$beta[length(object$beta)]) %>%
        median
      mae.brmed <- abs(object$delta$br.med - object$beta[length(object$beta)]) %>%
        median

    ## Infinite estimates
    iter.table <- list(ml.iter = table(object$iter[,1]),
                       brm.iter = table(object$iter[,2]),
                       brmed.iter = table(object$iter[,3]))
    noconv <- object$noconv

    return(list(bias = bias, "Coverage probability" = c(cp.ml, cp.brm, cp.brmed),
                "PU" = c(pu.ml, pu.brm, pu.brmed),
                "MSE" = c(mse.ml, mse.br, mse.brmed),
                "MAE" = c(mae.ml, mae.brm, mae.brmed),
                "iter tables" = iter.table,
                "frequency of se greater than 3" = noconv
                )
           )
}

analysis(obj1)
```

```
## $bias
##               coef          MLE        BRmean      BRmedian
## (Intercept)    0 -0.031260699 -7.163959e-05 -0.0066991749
## ebertMixed     0 -0.003614745  7.496866e-04 -0.0016722739
## ebertPro       0  0.004937764 -2.640192e-03  0.0005378334
## siskelMixed    0 -0.003920953  4.947952e-04 -0.0018765928
## siskelPro      0  0.003275974  8.542530e-05  0.0012689776
## diag           0  0.014633493  2.227490e-03  0.0042116638
##
## $`Coverage probability`
## [1] 0.9441 0.9529 0.9509
##
## $PU
## [1] 0.4705 0.4987 0.4948
##
## $MSE
## [1] 0.03071772 0.02926750 0.02961955
##
## $MAE
## [1] 0.1184029 0.1161062 0.1167387
##
## $`iter tables`
## $`iter tables`$ml.iter
```

```
##
##    3    4    5
##  547 9129  324
##
## $`iter tables`$brm.iter
##
##    2    3    4
## 5969 4030    1
##
## $`iter tables`$brmed.iter
##
##    2    3    4
##  796 9202    2
##
##
## $`frequency of se greater than 3`
## [1] 0 0 0
```

```
analysis(obj2)
```

```
## $bias
##             coef          MLE       BRmean      BRmedian
## (Intercept)    0 -0.208537854 -0.006708122 -0.0460677517
## ebertMixed     0 -0.006387040  0.009948878  0.0027086000
## ebertPro       0  0.072183067  0.005496901  0.0239594444
## siskelMixed    0 -0.052695647 -0.004583374 -0.0206706022
## siskelPro      0 -0.006916033 -0.004138811 -0.0008238712
## diag           0  0.094673451 -0.001324998  0.0144170083
##
## $`Coverage probability`
## [1] 0.9475 0.9654 0.9586
##
## $PU
## [1] 0.4580 0.5180 0.5046
##
## $MSE
## [1] 1.1082014 0.1526082 0.1720791
##
## $MAE
## [1] 0.2725476 0.2508369 0.2588350
##
## $`iter tables`
## $`iter tables`$ml.iter
##
##    3    4    5    6    7    8   18   19   20   21   22   23   24
##   43 3544 5364  974   44    1    5    7    3    2    3    8    2
##
## $`iter tables`$brm.iter
##
##    3    4    5    6    7    8    9   10   11   12   13   18   27
##  915 5845 2451  575  138   36   17    8    4    6    3    1    1
##
## $`iter tables`$brmed.iter
##
##    3    4    5    6    7    8    9   10   11   12   13   15   16   17   18   19
##   80 3859 4434 1192  286   80   31   16    5    3    1    3    3    2    1    2
##   20   42
##    1    1
##
##
## $`frequency of se greater than 3`
## [1] 30  0  0
```

```
analysis(obj3)
```

```
## $bias
##             coef        MLE      BRmean    BRmedian
## (Intercept)    0 -3.4650157 -0.033570222 -0.20020800
## ebertMixed     0 -0.6589267  0.013982477 -0.02892473
## ebertPro       0  0.5194667  0.014891414  0.05642192
## siskelMixed    0 -0.6465212 -0.003706696 -0.04661002
## siskelPro      0  0.7017859  0.008594486  0.05565563
## diag           0  2.1204974  0.008786705  0.08658163
##
## $`Coverage probability`
## [1] 0.9810 0.9783 0.9766
##
## $PU
## [1] 0.4408 0.5203 0.4990
##
## $MSE
## [1] 81.1205151  0.4018240  0.6666159
##
## $MAE
## [1] 0.4723059 0.3797605 0.4128591
##
## $`iter tables`
## $`iter tables`$ml.iter
##
##    3    4    5    6    7   17   18   19   20   21   22   23   24   25
##    3  793 4841 2491  162   25  283  304   94   93  213  425  260   13
##
## $`iter tables`$brm.iter
##
##    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
##    8  688 2380 2497 1639  815  595  326  265  171  140  102   84   67   56   27
##   19   20   21   22   23   24   25   26   27   28   29   30   31   32   33   34
##   20   27   19   15    9    7    6    5    1    4    6    1    1    4    1    3
##   35   37   38   42   47   49
##    2    2    3    1    1    2
##
## $`iter tables`$brmed.iter
##
##    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19
##  125 1113 2295 2120 1546  549  457  363  236  197  106   52   84   77  107  116
##   20   21   22   23   24   25   26   27   28   29   30   31   32   33   34   35
##   95   74   38   36   31   32   16   11    8    9    3    5    8    4    2    9
##   36   37   38   39   40   41   42   43   44   45   46   47   48   49   50   51
##    7    2    6    2    4    2    3    4    2    7    4    2    1    7    1    1
##   52   53   54   55   57   58   59   62   64   73
##    4    1    1    4    3    1    1    3    2    1
##
##
## $`frequency of se greater than 3`
## [1] 1710    0    0
```

```
analysis(obj32)
```

```
## $bias
##             coef          MLE        BRmean     BRmedian
## (Intercept)    0 -0.036570734 -0.0011779374 -0.011975927
## ebertPro       0 -0.006274213 -0.0012933087 -0.004401333
## siskelMixed    0  0.020407498  0.0030418848  0.009527812
## siskelPro      0  0.012284839  0.0002748273  0.005400394
## diag           0  0.022035009  0.0014008022  0.009661926
##
## $`Coverage probability`
## [1] 0.9493 0.9599 0.9550
##
## $PU
## [1] 0.4764 0.5129 0.4995
##
## $MSE
## [1] 0.04884427 0.04596938 0.04707862
##
## $MAE
## [1] 0.1459329 0.1419001 0.1430497
##
## $`iter tables`
## $`iter tables`$ml.iter
##
##    3    4    5    6
## 4776 5203   20    1
##
## $`iter tables`$brm.iter
##
##    2    3    4
## 9357  642    1
##
## $`iter tables`$brmed.iter
##
##    2    3    4
## 3458 6538    4
##
##
## $`frequency of se greater than 3`
## [1] 0 0 0
```

```
analysis(obj32b)
```

```
## $bias
##             coef        MLE        BRmean    BRmedian
## (Intercept)    0 -0.3429989 -0.0128296950 -0.07619771
## ebertPro       0 -0.1182435 -0.0003900982 -0.01901512
## siskelMixed    0  0.1477840  0.0076533368  0.03228415
## siskelPro      0  0.1736656  0.0101414537  0.04147001
## diag           0  0.2477334  0.0038002261  0.05181261
##
## $`Coverage probability`
## [1] 0.9669 0.9784 0.9730
##
## $PU
## [1] 0.4583 0.5335 0.4995
##
## $MSE
## [1] 3.9299333 0.2474123 0.3091238
##
## $MAE
## [1] 0.3318644 0.3035779 0.3150959
##
## $`iter tables`
## $`iter tables`$ml.iter
##
##    3    4    5    6    7    8   21   22   23
## 1799 5806 1800  464   40    1   33   44   13
##
## $`iter tables`$brm.iter
##
##    3    4    5    6    7    8    9   10   11   12   13   14   15   16   18
## 3907 4947  631  227  122   69   40   14    9   12    5    6    7    2    2
##
## $`iter tables`$brmed.iter
##
##    3    4    5    6    7    8    9   10   11   12   13   14   16   17   18   36
##  294 5809 2848  731  217   37   27   11   15    4    1    2    1    1    1    1
##
##
## $`frequency of se greater than 3`
## [1] 90  0  0
```

```
analysis(obj32c)
```

```
## $bias
##             coef       MLE        BRmean    BRmedian
## (Intercept)    0 -4.689088 -8.742518e-03 -0.26174042
## ebertPro       0 -1.702997  1.680372e-02 -0.09087244
## siskelMixed    0  2.854158  8.828512e-05  0.12809593
## siskelPro      0  2.488106 -5.717505e-03  0.14203831
## diag           0  3.794779 -4.993344e-03  0.19963438
##
## $`Coverage probability`
## [1] 0.9987 0.9957 0.9954
##
## $PU
## [1] 0.4184 0.5561 0.4889
##
## $MSE
## [1] 110.809871   0.609961   1.099466
##
## $MAE
## [1] 0.6235227 0.4693928 0.5406610
##
## $`iter tables`
## $`iter tables`$ml.iter
##
##    3    4    5    6    7   21   22   23   24
##  484 3385 2734 1335   40  592 1118  277   35
##
## $`iter tables`$brm.iter
##
##    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
##   61 3395 2706  964  618  532  402  314  235  154  168   84   84   69   58   36
##   19   20   21   22   23   24   25   26   28   32   33   37   38   39   40   42
##   30   33   11   12    9    4    1    6    2    2    2    1    1    1    1    1
##   45   48
##    2    1
##
## $`iter tables`$brmed.iter
##
##    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
##    2  300 2197 2397 1702 1118  507  364  429  257  180  140   73   40   61   59
##   19   20   21   22   23   24   25   26   27   29   30   32   34   35   36   37
##   40   15   16    4    5    3    2    1    1    1    1    1    1    8    7   10
##   38   39   41   42   43   44   45   46   47   48   50   51   52   53
##    8    6    4    5    2    3    8    3    4    3    4    2    5    1
##
##
## $`frequency of se greater than 3`
## [1] 2022    0    0
```

Bibliography

Agresti, A. (1988). A model for agreement between ratings on an ordinal scale. *International Biometric Society* **44**, 539–548.

Agresti, A. (2010). *Analysis of Ordinal Categorical Data*. Wiley, 2nd ed.

Agresti, A. (2019). *An Introduction to Categorical Data Analysis, 3rd Edition*. Wiley.

Cordeiro, G. & McCullagh, P. (1991). Bias correction in generalized linear models. *Journal of the Royal Statistical Society. Series B (Methodological)* **53**, 629–643.

Cox, D. & Hinkley, D. (1974). *Theoretical Statistics*. Chapman and Hall.

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika* **80**, 27–38.

Kenne Pagui, C., Salvan, A. & Sartori, N. (2017). Median bias reduction of maximum likelihood estimates. *Biometrika* **104(4)**, 923–938.

Kosmidis, I. (2014). Bias in parametric estimation: reduction and useful side-effects. *WIREs Comput Stat* **6**, 185–196.

Kosmidis, I. (2020). *brglm2: Bias Reduction in Generalized Linear Models*. R package version 0.6.2.

Kosmidis, I., Kenne Pagui, C. & Sartori, N. (2020). Mean and median bias reduction in generalized linear models. *Statistics and Computing* **30**, 43–59.

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

Salvan, A., Sartori, N. & Pace, L. (2020). *Modelli Lineari Generalizzati.* Springer.