BANOVA: An **R** Package for Hierarchical Bayesian ANOVA

Chen Dong
University of Maryland, College Park

Michel Wedel
University of Maryland, College Park

Abstract

In this paper, we develop generalized hierarchical Bayesian ANOVA, to assist experimental researchers in the behavioral and social sciences in the analysis of experiments with within- and between-subjects factors. The method alleviates several limitations of classical ANOVA, still commonly employed in those fields of research. An accompanying R Package for BANOVA is developed. It offers statistical routines and several easy-to-use functions for estimation of hierarchical Bayesian ANOVA models that are tailored to the analysis of experimental research. MCMC simulation is used to simulate posterior samples of the parameters of each model specified by the user. The core program is written in R and **JAGS**. After preparing the data in the required format, users simply select an appropriate model, and can estimate it without any advanced coding being required. The main aim of the R package is to offer freely accessible resources for hierarchical Bayesian ANOVA analysis, which makes it easy to use for applied researchers.

Keywords: hierarchical Bayes regression, ANOVA, BANOVA, MCMC, BUGS modeling.

1. Introduction

Because of its computational attractiveness and the ease of interpretation of its statistical tests and the corresponding tables of means, ANOVA (Fisher 1921, 1925) is implemented in most statistical packages and continues to garner tremendous popularity in applied research. As it is about to celebrate its centennial, behavioral and social scientists still rely heavily on ANOVA for the analysis of their data from experiments with human subjects (e.g., Cardinal and Aitken 2005). Yet, the standard approach to ANOVA is based on several assumptions that are often critically violated for the data typically collected in those fields of research.

First, ANOVA assumes a continuous homoscedastic i.i.d. normal distributed dependent variable. Categorical variables, however, for which ANOVA has long known not to be appropriate (Cochran 1940), abound in the behavioral sciences. These include measurements of perceptions, attitudes and intentions, on categorical rating and multiple choice scales, and

binary and count measures of human attention, memory and decision making (Nunnaly 1967; Thorndike 1971; Lord and Novick 1968). In addition, continuous measures such as response times or monetary expenditures, which often have high skewness or kurtosis, are commonly used as well. The distributional properties of most of these variables violate the assumptions underlying ANOVA, and extensive research into the effect of these violations has shown that they may lead to both excess type-I and type-II errors in classical significance testing (Ito 1980; Tan 1982; Tiku 1971). Transformations of the data, such as the log and square-root (for counts), logit and arcsine (for proportions), rank (for ordered categorical variables), and Box-Cox transformations have been used to render the empirical distribution closer to the normal (Bartlett 1947; Box and Cox 1964; Draper and Hunter 1969; Conover and Iman 1976). These transformations, however, often do not provide a satisfactory solution, because they cause the ANOVA tables of means to lose some of their appealing interpretations, while significance levels of the transformed and original data often do not correspond. Modern statistical solutions are available in the form of generalized linear models (McCullagh and Nelder 1989). The application of GLMs capitalizes on the fact that ANOVA is a special case of linear regression models, and GLMs extend regression models to a wide variety of models in which the dependent variable follows one of many distributions in the exponential family. However, for applied researchers a downside of the use of GLM to analyze designed experiments that involve of multiple factors and interactions is that these need to be represented in the model through dummy variables, and the interpretation of estimates of coefficients of these dummy variables is not as easy as interpreting the output of ANOVA, and indeed may often be quite cumbersome.

Second, in the behavioral sciences experiments often involve a combination of between- and within-subjects factors, leading to nested and repeated measurement designs. For these designs established (mixed, split-plot and repeated measures) ANOVA procedures are available in most statistical packages. These mixed ANOVA models in question may have both fixed and random effects (Hartley and Rao 1967; é 1957), but assume a balanced design, a continuous dependent variable and categorical independent variables. Unbalanced designs, unequally spaced measurements, and continuous covariates such as encountered in ANCOVA, violate these assumptions. As a consequence, some experimental behavioral science researchers resort to the application of hierarchical linear models (Breslow and Clayton 1993; Longford 1987; Raudenbusch 1988; Raudenbush 1999), in particular for the analysis of quasi(field) experiments. These models, which take on various forms and are correspondingly labeled with a variety of names in the literature, are all special cases of hierarchical Bayes models (Lindley and Smith 1972; Press 2003; Gelman, Carlin, Stern, and Dunson 2013a). Extensive treatments can be found in Gelman, Carlin, Stern, Dunson, Vehtari, and Rubin (2013b) and Gelman and Hill (2006). These models allow for more general covariance structures and data hierarchies than repeated measures ANOVA. Gelman (2005) argued the importance of hierarchical Bayes formulations of ANOVA, and showed how the principles of ANOVA are helpful in understanding hierarchical linear models. In addition, hierarchical Bayes models can accommodate non-normal dependent variables that render the application of classical ANOVA and hierarchical linear models problematic. The Bayesian approach in addition offers a number of theoretical and pragmatic advantages as a framework for inference and testing that have been widely acknowledged (Bernardo J 2000; Press 2003; Savage 1954). Indeed, in behavioral research the advantages of Bayesian inference are increasingly recognized, in that it provides inferences based on finite samples, and avoids some of the pitfalls of classical hypothesis

testing (Gelman 2013; Kruschke 2013; Rouder, Speckman, Sun, Morey, and Iverson 2009). Moreover, Hierarchical Bayes models can now be relatively easily be implemented using existing statistical software, such as BUGS (Lunn, Thomas, Best, and Spiegelhalter 2000) and **JAGS** (Plummer 2016a).

Nevertheless, applied experimental researchers in the behavioral and social sciences continue to resort to standard ANOVA in many cases in spite of its limitations and availability of modern superior alternatives, because of its ease of application and interpretation, widespread availability in standard statistical packages, lack of familiarity with better alternatives, and/or the effort involved in programming alternative methods or interpreting their output. The present paper attempts to help remedy this undesirable state of affairs by developing a framework and accompanying R package for hierarchical Bayes ANOVA, called BANOVA, which addresses the most salient limitations of classical ANOVA, yet is easy to use and retains many of the familiar features of the outputs of classical ANOVA. Amongst others, it accommodates a wide range of distributions for the dependent variable, hierarchical data structures and between- and within- subjects categorical design factors, as well as continuous covariates.

The key insight behind the BANOVA approach is that ANOVA and ANCOVA are special cases of linear regression and that once an ANOVA model is formulated as a hierarchical linear model, subject-level parameters become incidental and inference focuses entirely on the population-level model, which is where main effects and interactions of within- and between-subjects factors are represented and tested. Assuming that lower-level parameters that describe subject heterogeneity follow normal distributions, it follows that variance decomposition, tables of means and significance tests of main effects and interactions can be computed, similar to how they would be computed in standard ANOVA, from the population-level parameter estimates. This then allows for the analysis of dependent variables with a wide variety of distributional forms with hierarchical models, while at the same time retaining much of the appealing output from standard ANOVA for experimental data. The underlying estimation methods in the package are Markov Chain Monte Carlo (MCMC) algorithms implemented in the **JAGS** software. The user of the package needs to input the data and set up a few parameters (or accept the defaults). The package then sets up a **JAGS** program and analyses the data with a hierarchical Bayes ANOVA, using MCMC estimation. **JAGS** was chosen as an interface, because after calling the R package, in addition to the estimation results, the **JAGS** code will be available for inspection and modification by the statistically more advanced behavioral researcher. Importantly, although the underlying models are hierarchical Bayes models, the output of these models is presented in a form that is very familiar to users of standard ANOVA, including tables of means with credible intervals, (Bayesian) p values and effect sizes.

The remainder of this paper is organized as follows. In Section 2, we discuss the hierarchical Bayesian approach to ANOVA. In Section 3, the architecture of the R package is discussed and a brief tutorial is provided. Bayesian estimation of parameters and other quantities of interest are described as well. We include examples in Section 4. The last section concludes and suggests future work.

2. BANOVA models

We assume data are collected in an experiment in which a sample of subjects have participated, and have been exposed to between-subject as well as within-subjects experimental

manipulations. Repeated measurements of one or more dependent variables are taken on each subject, while continuous or categorical covariates may have been measured as well. The hierarchical Bayesian BANOVA approach then consists of sub-models at two levels: at level 1 the subject-level model, and at level 2 the population-level model. The subject-level model represents the effects of within-subjects factors and covariates, and the population-level model represents the influence of between-subjects factors. The population-level model allows for the calculation of statistics that can be readily interpreted by applied researchers, as they would with standard ANOVA.

In the *subject-level model*, each outcome of the dependent variable, y_i, with i indexing data points, is assumed to be generated from a particular distribution in the exponential family, $f(y_i|\mu_i)$ (and even other distributions can be accommodated). The mean, μ_i, of the distribution depends on the independent variables through a suitable link function $g(\cdot)$ (McCullagh and Nelder 1989). The within-subjects factors and their interactions are indexed by $p(p = 1, 2, \ldots, P)$. Each index p represents a batch of J_p coefficients: $\beta_{j,s}^p$, $j = 1, \ldots, J_p$; where $s = 1, \ldots, S$ indexes subjects. Note that if a subject-level covariate is continuous, $J_p = 1$, so that ANCOVA models are also accommodated (but the present formulation relaxes their "constant slope" assumption). The subject-level model is expressed as a generalized linear regression model, with a design matrix X that contains all within-subjects factors and their interactions, as well as a constant term, indexed by $p = 0$:

$$\mathsf{E}(y_i) = g^{-1}(\eta_i), \tag{1}$$

$$\eta_i = \sum_{p=0}^{P} \sum_{j=1}^{J_p} X_{i,j}^p \beta_{j,s_i}^p, \tag{2}$$

where s_i is the subject index corresponding to data point i.

The *population-level model* allows for unobserved heterogeneity among subjects, because the subject-level coefficients $\beta_{j,s}^p$ are assumed to follow a multivariate normal distribution. The between-subjects factors and their interactions are indexed by $q(q = 1, 2, \ldots, Q)$; $q = 0$ denotes the constant term. Then, using the notation in Gelman (2005), the population-level BANOVA is:

$$\beta_{j,s}^p = \sum_{q=0}^{Q} \theta_{j,k_s^q}^{pq} + \delta_{j,s}^p. \tag{3}$$

Each index q represents a batch of K_q coefficients: $\theta_{j,k}^{pq}$, $k = 1, \ldots, K_q$; k_s^q indexes coefficient k in batch q corresponding to the treatment of subject s. For example, in the simple case of one 3-level within-subjects factor D ($P = 2$, $J_1 = 1$ and $J_2 = 3$) and two 2-level between-subjects factors A and B, and the AB-interaction, $Q = 3$, Equation 2 and Equation 3 reduce to (with the parameter of the last level of each factor set to zero):

$$\eta_i = \beta_{1,s}^0 + \sum_{j=1}^{2} X_{i,j}^D \beta_{j,s_i}^D, \tag{4a}$$

$$\beta_{1,s}^0 = \theta_1^0 + \theta_{1,k_s^A}^A + \theta_{1,k_s^B}^B + \theta_{1,k_s^{AB}}^{AB}, \tag{4b}$$

$$\beta_{1,s}^D = \theta_1^{D0} + \theta_{1,k_s^A}^{DA} + \theta_{1,k_s^B}^{DB} + \theta_{1,k_s^{AB}}^{D\,AB}, \tag{4c}$$

$$\beta_{2,s}^D = \theta_2^{D0} + \theta_{2,k_s^A}^{DA} + \theta_{2,k_s^B}^{DB} + \theta_{2,k_s^{AB}}^{D\,AB}. \tag{4d}$$

Equation 4b contains the overall intercept (θ_1^0) and the main effects of the between-subjects factors A ($\theta_{1,k_s^A}^A$) and B ($\theta_{1,k_s^B}^B$) and their interaction ($\theta_{1,k_s^{AB}}^{AB}$). Equation 4c contains the main effect of the (first level) of the within-subjects factor D (θ_1^{D0}), and its two-($\theta_{1,k_s^A}^{DA}, \theta_{1,k_s^B}^{DB}$) and three-way ($\theta_{1,k_s^{AB}}^{D\,AB}$) interactions with the between-subjects factors A and B. Similarly, Equation 4d contains the main effect of the second level of D (θ_2^{D0}), its two -way interactions with A ($\theta_{2,k_s^A}^{DA}$) and B ($\theta_{2,k_s^B}^{DB}$), and the three-way interaction ($\theta_{2,k_s^{AB}}^{D\,AB}$). Equations 4b to 4d are thus similar to those of a standard ANOVA model with two 2-level factors and their interaction.

The population-level BANOVA model can be expressed as a linear model with a design matrix Z that contains all between-subjects factors and their interactions (and a constant term):

$$\beta_{j,s}^p = \sum_{k=1}^{K} Z_{s,k} \theta_{j,k}^p + \delta_{j,s}^p, \tag{5}$$

where $Z_{s,k}$ is an element of Z, a $S \times K$ matrix of covariates, and K is the number of parameters. $\theta_{j,k}^p$ is a hyper-parameter that captures the effects of between-subjects factor q on the parameter $\beta_{j,s}^p$ of within-subjects factor p. The error $\delta_{j,s}^p$ is assumed to be normal: $\delta_{j,s}^p \sim N(0, \sigma_p^2)$. Proper, but diffuse priors are assumed: $\theta_{j,k}^p \sim N(0, s)$, and $\sigma_p^{-2} \sim Gamma(a, b)$, where s, a, b are hyper-parameters.

The BANOVA model is estimated capitalizing on the fact that it is a special case of hierarchical generalized linear models, that is, using Equations 1, 2 and 5. We use effects-coding of the factors in the matrices X and Z (Overall, Spiegel, and Cohen 1975). It is important to note that Equation 3 is the equation that is of key interest for inference. It contains the parameters that specify the population-level BANOVA model. It is as if the subject-level coefficients $\beta_{j,s}^p$ are the (normally distributed) "dependent variables" in an ANOVA, specified by the between-subjects factors in Equation 3. Thus, inferences focuses on the parameters in Equation 3. Data that comes from a between-subject design, with only a single measurement of the dependent variable for each subject, is also accommodated. In that case, Equations 2 and 5 reduce to a single Equation 2, where the subscript s_i is replaced by i. In the absence of repeated measurements for each subject, unobserved heterogeneity is not identified for most models. **BANOVA** can still be used in those cases, but reduces to a Generalized Linear Model with a canonical link function. We next specify the specific outcome variables that are accommodated in the R package.

Continuous responses: To model continuous data, a normal distribution can be assumed for y_i:

$$y_i = \eta_i + \epsilon_i; \quad \epsilon_i \sim N(0, \sigma^2), \tag{6}$$

where η_i is defined in Equation 2, and the prior for the scale parameter σ, is specified through $\sigma^{-2} \sim Gamma(\alpha, \beta)$, with α and β hyper-parameters.

To describe data with "outliers" or fatter tails than the normal, the ϵ_i in Equation 6 can be assumed to follow a t- distribution, with an unknown number of degrees of freedom, assumed to follow a Poisson distribution:

$$\epsilon_i \sim t(0, \phi^2, \nu), \tag{7}$$

The priors are $\phi^{-2} \sim Gamma(\alpha, \beta)$, and $\nu \sim Poisson(\lambda)$, with α, β, and λ hyper-parameters.

Binary responses: To model data y_i that take on the values 0 and 1, a Bernoulli distribution is assumed,

$$y_i \sim Binomial(1, p_i),\ p_i = \text{logit}^{-1}(\eta_i), \tag{8}$$

where $\text{logit}(x) = \ln\frac{x}{1-x}$ is the standard logit link-function.

If the data y_i represents the number of successes in a sequence of B independent Bernoulli experiments, then,

$$y_i \sim Binomial(B, p_i),\ p_i = \text{logit}^{-1}(\eta_i), \tag{9}$$

Count responses: To model count data y_i that can take on integer values $0, 1, 2, \ldots$, the Poisson distribution is assumed:

$$y_i \sim Poisson(\lambda_i);\ \lambda_i = \exp(\eta_i). \tag{10}$$

Ordered categorical responses: To model data y_i that are ordered categorical and can take on the values $\ell = 1, \ldots, L$, an ordered logistic model is used,

$$Pr(y_i > \ell) = \text{logit}^{-1}(\eta_i - c_{\ell-1});\ \ell = 1, \ldots, L - 1. \tag{11}$$

The cut-point parameters c_ℓ are constrained: $0 = c_1 < c_2 < \cdots < c_{L-1}$. We assume $c_1 = 0$, and the other cut-points follow a normal distribution $c_\ell \sim N(0, \bar{\sigma}_\ell^2)$. The prior distribution of $\bar{\sigma}_\ell^2$ is $\bar{\sigma}_\ell^2 \sim Uniform(0, d)$, with d a hyper-parameter.

Multinomial responses: To model data y_i that are categorical and can take on the values $1, \ldots, L$, a multinomial logistic model (MNL) is used,

$$Pr(y_i = \ell) = \frac{\exp(\eta_{i,\ell})}{\sum_{\ell=1}^{L} \exp(\eta_{i,\ell})} \tag{12}$$

where $\eta_{i,\ell} = \sum_{p=0}^{P} \sum_{j=1}^{J_p} X_{i,j,\ell}^{k,p} \beta_{j,s_i}^{p}$, and $X_{i,j,\ell}^{k,p}$ is the design matrix corresponding to each response category $\ell(\ell = 1, \ldots, L)$ of y_i.

3. BANOVA R package

3.1. Obtaining the software

The **BANOVA** software is an add-on package to the statistical software R. It is free and can be downloaded from the Comprehensive R Archive Network (CRAN, https://CRAN.R-project.org/package=BANOVA). The **BANOVA** package is implemented in R and **JAGS**. Thus, an additional system requirement is the **JAGS** software, which can be freely downloaded from http://mcmc-jags.sourceforge.net/. The package also imports three other packages, **runjags** (Denwood 2016), **rjags** (Plummer 2016b), and **coda** (Plummer, Best, Cowles, and Vines 2006) in order to connect R and **JAGS** and calculate convergence diagnostics. Note, these three imported packages do not necessarily need to be installed before installing **BANOVA**. They are automatically attached to the package and loaded when the package **BANOVA** is loaded. However, the **JAGS** software must be installed in order to estimate any of the models introduced above. The **BANOVA** package will automatically detect the location of **JAGS** software and connect it via **runjags**. Once R and **JAGS** have been installed, **BANOVA** can be loaded using the following command: `library("BANOVA")`.

3.2. Functionality

BANOVA can fit the Bayesian hierarchical ANOVA models introduced in the previous section. As explained there, the response variable may follow a wide variety of distributions including normal, student's t, Poisson, binomial, ordered and unordered multinomial distributions. Each of the corresponding models can be fitted by a specific function in the package[1]. The names of these functions have the form of '`BANOVA.Binomial`', where the first part specifies the general name and the second part after the '.' specifies the form of the likelihood. Currently, there are seven model classes included in the package.

1. `BANOVA.Bernoulli()` – the model in which the response variable follows a Bernoulli distribution (Equation 8).

2. `BANOVA.Binomial()` – the model in which the response variable follows a binomial distribution (Equation 9).

3. `BANOVA.Multinomial()` – the model in which the response variable follows an unordered multinomial distribution (Equation 12).

4. `BANOVA.Normal()` – the model in which the response variable follows a normal distribution (Equation 6).

5. `BANOVA.ordMultinomial()` – the model in which the response variable follows an ordered categorical distribution (Equation 11).

6. `BANOVA.Poisson()` – the model in which the response variable is considered a count variable which follows a Poisson distribution (Equation 10).

7. `BANOVA.T()` – the model in which the response variable follows a t distribution (Equation 7).

The predictor for each Bayesian ANOVA model is specified as a regular R object, which is similar to the `lm()` and `glm()` objects in R. This means that the `summary()`, `print()` and `predict()` functions can be applied to the object in question after fitting the model. In addition to the common R object functions, the package also includes several useful functions such as `conv.diag()`, `table.predictions()`, `table.pvalues()` and so on. Their use is illustrated in following sections.

3.3. Data input

When the data is in '.csv' or other formats, it can be loaded with the R function `read.csv()` or other import functions. The package expects the data imported to be in a long format where each row corresponds to one trial, replication, or time point per subject. Thus, each subject will have data in multiple rows. Subject ID values must be included in the data (see Figure 1), and the other columns in the data set contain the dependent variable(s), the covariates, and the between- and within- subjects experimental factors. The continuous independent variables in the data are automatically mean-centered within the package. The

---

[1]The **BANOVA** package also comes with a function `BANOVA.run()`. This function has similar functionality as the ones described in the paper but interfaces with the **Stan** (Carpenter *et al.* 2017) software. It is described in detail in the manual.

Subject_ID	Category	Brand_Accuracy	Size	Blur	Color
1	1	1	0	0	0
1	1	0	0	0	0
1	1	1	0	0	0
1	2	0	0	0	0
1	2	1			0
1	2	1			0
2	2	0	0	1	1
2	2	1	0	1	1
2	2	1	0	1	1
2	1	0	0	1	1
2	1	0	0	1	1
2	1	1	0	1	1
3	1	0	1	0	1
3	1	0	1	0	1
3	1	1	1	0	1
3	2			0	1
3	2			0	1
3	2	1	1	0	1
4	2	0	1	1	0
4	2	1	1	1	0
4	2	0	1	1	0
4	1	1	1	1	0
4	1	0	1	1	0
4	1	1	1	1	0

Response variable

Between-subject variables

Within-subject variable

Figure 1: Example of the format of input data.

between-subjects variables, which are constant within each subject, will have the same value in all rows containing the data for one subject. The attribute of each of the factors must be specified as one of the following three classes: "integer", "numeric" or "factor". The function `class()` in R can be used to check the classes of factors. For example,

```
R> class(x)
R> x <- as.factor(x)
```

For the multinomial response model (Equation 12), the required data format is different from that shown in Figure 1. The within- subjects data for each subject must be stored in one item of a 'list' in R. For example, if there are 100 subjects, then the list must contain 100 items where each item includes multiple rows that denote the values of the within-subjects variables. The between-subjects data is stored in a separate data frame where each row corresponds to one subject. The order of the between-subjects data must match the order of within-subjects data. For example, although the 'choicedata' included in the **BANOVA** package is already in a long format, both within-subjects and between-subjects data needs to be further manipulated. The following R code can be used for that purpose:

```
R> data("choicedata")
```

```
R> dataX <- list()
R> for (i in 1:nrow(choicedata)){
+   logP <- as.numeric(log(choicedata[i, 3:8]))
+   dataX[[i]] <- as.data.frame(logP)
+   }
R> dataZ <- choicedata[, 9:13]
```

Data checking for the multinomial and Poisson distributions is handled internally in the package, and the `as.integer` operation for categorical response variables is also applied by default.

3.4. Estimation of the coefficients

As explained above, the BANOVA model is estimated by capitalizing on the fact that it is a special case of a hierarchical generalized linear model, that is, using Equations 1, 2 and 5. The conditional posterior distribution, denoted by $\pi(\cdot)$, of the parameters $\beta_{j,s}^p$ is obtained from the likelihood and priors:

$$\pi(\beta_{j,s}^p|\cdot) \propto \pi(y_i|\boldsymbol{\beta}^1,\ldots,\boldsymbol{\beta}^p,x_i)\pi(\boldsymbol{\beta}^1|\boldsymbol{\theta}^1,\mathbf{Z})\cdots\pi(\boldsymbol{\beta}^p|\boldsymbol{\theta}^p,\mathbf{Z})\pi(\boldsymbol{\theta}^1)\cdots\pi(\boldsymbol{\theta}^p), \tag{13}$$

where $\pi(y_i|\boldsymbol{\beta}^1,\ldots,\boldsymbol{\beta}^p,x_i)$ is the likelihood determined by Equations 1, 2 and 6 to 12; $\boldsymbol{\beta}^p = (\boldsymbol{\beta}_1^p,\ldots,\boldsymbol{\beta}_{J_p}^p)$, $p = 1,\ldots,P$, with $\boldsymbol{\beta}_j^p = (\beta_{j,1}^p,\ldots,\beta_{j,s}^p)$, $j = 1,\ldots,J_p$, are the coefficients of factor p; $\boldsymbol{\theta}^p = (\theta_{j,k}^{p1},\theta_{j,k}^{p2},\ldots,\theta_{j,k}^{pQ})$, $j = 1,\ldots,J_p, k = 1,\ldots,K_q$ the population-level parameters; $\pi(\boldsymbol{\theta}^p), p = 1,\ldots,P$ denotes the prior of population-level parameters $\boldsymbol{\theta}^p$; and $\pi(\boldsymbol{\beta}^p|\boldsymbol{\theta}^p,\mathbf{Z})$, $p = 1,\ldots,P$, is the prior determined by the population model in Equation 3. We are interested in the effects of between-subjects factors, captured by $\theta_{j,k}^{pq}$, $k = 1,\ldots,K_q$ using the notation in Equation 3. The conditional posterior distribution of the parameters $\theta_{j,k}^{pq}$ is:

$$\begin{aligned}\pi(\theta_{j,k}^{pq}|\cdot) &\propto \pi(\boldsymbol{\beta}_j^p|\boldsymbol{\theta}^p,\mathbf{Z})\pi(\boldsymbol{\theta}^p)\\&= \pi(\boldsymbol{\beta}_j^p|\boldsymbol{\theta}^p,\mathbf{Z})\prod_{jk}\pi(\theta_{j,k}^{p1})\prod_{jk}\pi(\theta_{j,k}^{p2})\cdots\prod_{jk}\pi(\theta_{j,k}^{pQ}).\end{aligned} \tag{14}$$

where for each $\theta_{j,k}^{pq}$ a normal prior is assumed, $\theta_{j,k}^{pq} \sim N(0,s)$, with hyper-parameter s.

3.5. Parameter values for the prior distributions

All prior distributions for the parameters are conjugate diffuse priors. For the parameters in Equation 5, these priors are: $\theta_{j,k}^p \sim N(0,s)$, and $\sigma_p^{-2} \sim Gamma(a,b)$, where s,a,b are hyper-parameters. For Equation 6, the prior is: $\sigma^{-2} \sim Gamma(\alpha,\beta)$, where α and β are hyper-parameters. For Equation 7, the prior distributions are $\phi^{-2} \sim Gamma(\alpha,\beta)$ and $\nu \sim Poisson(\lambda)$, where α,β,λ are hyper-parameters. For Equation 11, the prior distributions are $\bar{\sigma}_\ell^2 \sim Uniform(0,d)$, for $\ell = 2,\ldots,L-1$.

The values of the hyper parameters a,b,s in Equation 5 can be assigned by setting the `l2_hyper` arguments of the **BANOVA** package. The values of the hyper parameters α,β in Equation 6, or the hyper parameters α,β,λ in Equation 7 can be assigned by setting the `l1_hyper` arguments for the **BANOVA** package. In absence of user input of these arguments, the following default values are assumed: $a = 1, b = 1, \alpha = 1, \beta = 1$. Note that these parameter values for the Gamma distributions imply a considerable prior probability on smaller

values near zero (Gelman *et al.* 2013b). Further, the default values of the hyper-parameters $\lambda = 1$ and $s = 10^{-4}$. For the parameter d in Equation 11 the default value $d = 100$ is assumed.

3.6. Starting values and burn-in period

Successful implementation of MCMC algorithm requires proper starting values and a sufficiently long burn-in period to ensure convergence. The default number of burn-in is set to 5000, and target draws is 2000. But users can choose different numbers via the arguments of all `BANOVA.*()` functions. The burn-in draws are discarded. Default thinning is 1 in 10 draws. The starting values of all parameters are assigned by the R package. For example, unconstrained parameters are assigned starting values drawn from a normal distribution using the `rnorm()` function (the starting values of $\theta_{j,k}^{pq}$ in Equation 14, for example).

3.7. JAGS code

Based on Equation 13 and 14, the models are built and estimated in **JAGS** (Just Another Gibbs Sampler, Plummer (2003)). The **JAGS** program allows users to write their own models and select prior distributions, and frees them from dealing with the implementation details of models and samplers. The **BANOVA** package generates the **JAGS** code fully automatically, and runs it. The **JAGS** software takes care of the work involved in estimating model parameters by constructing a MCMC algorithm to sample from the posterior distributions of the parameters. The various functions in the package capture the output of the MCMC algorithms and perform the necessary calculations to present it in a format similar to standard ANOVA. The **JAGS** code is produced as part of the output of our package, so that advanced users can inspect and modify it. The following R command outputs the **JAGS** code generated to a file:

```
R> sink("JAGScode.txt")
R> cat(res$JAGSmodel)
R> sink()
```

where `res` is a list returned from the `BANOVA.*` function. The code can then be modified and run independently from the package.

3.8. Convergence diagnostics

There is a large number of convergence diagnostics available that can be used to assess whether the MCMC chains have reached their stationary distributions (e.g., Gill 2007). In the output of the package, two convergence diagnostics are reported: the Geweke diagnostic (Geweke 1992), and the Heidelberg and Welch (Heidelberger and Welch 1983) diagnostic. These two convergence diagnostics may be applied with any MCMC method and are calculated based on only a single MCMC chain, which saves some computation time and is less cumbersome for the applied user. The functions `geweke.diag` and `heidel.diag` in R package **coda** are incorporated in our package and used to compute the convergence diagnostics. If so desired the user can apply other diagnostics from the **coda** package manually.

Geweke's convergence diagnostic is calculated by taking the difference between the means from the first m_A iterations and the last m_B iterations, where M is the total number of iterations. If the ratios $\frac{m_A}{M}$ and $\frac{m_B}{M}$ are fixed and $m_A + m_B < M$, then the distribution of

this diagnostic approaches a standard normal as $M \to \infty$. In the package, $m_A = 0.1M$ and $m_B = 0.5M$.

The Heidelberg and Welch diagnostic is based on a test statistic to accept or reject the null hypothesis that the Markov chain is at its stationary distribution. The present package reports the Cramer-von Mises statistic to test for stationarity. The test is iteratively applied on batches of draws from the posterior distributions. If the null hypothesis is rejected, the first 10% of the iterations are discarded and the stationarity test repeated. If the test fails again, an additional 10% of the iterations are discarded and the test is repeated. The process continues until 50% of the iterations have been discarded and the test still rejects the null hypothesis. Our package uses the function `heidel.diag` in the **coda** package and sets the parameters $\epsilon = 0.1$, $pvalue = 0.5$.

To obtain the convergence tests, the following R command is used:

```
R> conv.diag(res)
```

where `res` is a fitted object from any of the models.

In case the chain does not pass one or more of the convergence tests, a warning message will be automatically displayed: 'The chain may not have converged. Consider a longer burn-in, speeding up the convergence by setting `conv_speedup = T`, or modifying the model.' The default number of burn-in draws is set to 5000, which the user can change manually to a larger number. For speeding up of convergence when `conv_speedup = T` is specified, an additional scale parameter is added which avoids the sampler getting stuck due to near-zero draws of variance parameters, as suggested by Gelman *et al.* (2013b) and Gelman and Hill (2006).

3.9. Tables of predictions

One key output of the package is a table of predicted values, classified by the factors at both level 1 and level 2. In line with ANOVA terminology these tables are called 'tables of means'. As explained above, effects-coding is used to estimate the parameters of categorical variables, using the last level of each factor as the reference level. However, especially when there are multiple factors and interactions, interpretation of the parameter estimates is cumbersome. Therefore, posterior samples of each $\theta_{j,k}^{pq}$ in Equation 3 generated by MCMC are used in the calculation of 'tables of means', similar to those produced by standard ANOVA. The advantage of doing this is that this output is familiar to behavioral applied researchers and relatively easy to interpret. Because these statistics are computed for each draw from the posterior distribution of the parameters, statistics from their posterior distributions are readily available. The package therefore also computes 95% credible intervals.

Let $\theta_{j,k,m}^{pq}$ denote the posterior sample of $\theta_{j,k}^{pq}$ in mth $(m = 1, 2, \ldots, M)$ draw of the MCMC chain. Then the predicted grand mean is:

$$\bar{\mu} = \frac{\sum_m g^{-1}(\theta_m^{00})}{M}, \tag{15}$$

where θ_m^{00} is the mth draw of the level-2 intercept corresponding to the level-1 intercept, which is equal to the grand mean. The 95% credible interval is simply provided by the 2.5% and 97.5% quantiles of the posterior distribution of $\{g^{-1}(\theta^{00})\}$.

Higher order tables are computed as illustrated below. For example, in computing the one-way table of predictions of a level-1 factor A, the posterior mean of its level j is calculated as:

$$\bar{\mu}_j^A = \frac{\sum_m g^{-1}(\theta_m^{00} + \mathbf{X}_j^{A'}\boldsymbol{\theta}_m^{A0})}{M},\tag{16}$$

where $\mathbf{X}_j^A$ is the effects-coded column vector of factor levels, corresponding to level j of factor A, in which all other factors and covariates are set to be 0; $\boldsymbol{\theta}_m^{A0}$ is the mth draw of the vector of level-2 intercepts corresponding to level-1 factor A.

As another example, in computing the one-way table of predictions of a level-2 factor B, the posterior mean of its level j is calculated as:

$$\bar{\mu}_j^B = \frac{\sum_m g^{-1}(\theta_m^{00} + \mathbf{Z}_{j,B}'\boldsymbol{\theta}_m^{0B})}{M},\tag{17}$$

where $\mathbf{Z}_{j,B}$ is the effects-coded column vector of factor levels corresponding to level j of factor B, in which all other factors and covariates are set to be 0; $\boldsymbol{\theta}_m^{0B}$ is the mth draw of the vector of level-2 coefficients of the effect of factor B on the level-1 intercept.

Continuing the example, the means of the two-way table classified by A and B (level j of factor A and level k of factor B) is calculated as:

$$\bar{\mu}_{j,k}^{AB} = \frac{\sum_m g^{-1}(\theta_m^{00} + \mathbf{X}_j^{A'}\boldsymbol{\theta}_{j,k,m}^{AB}\mathbf{Z}_{k,B}')}{M},\tag{18}$$

where $\boldsymbol{\theta}_{j,k,m}^{AB}$ is the mth draw from the coefficient matrix, the kth row of which is a vector of level-2 coefficients representing the effect of factor B on to the jth level of level-1 factor A.

Based on the above formulas, the function `table.predictions()` computes the tables of predicted values, or 'means' and their posterior quantiles.

```
R> table.predictions(res)
```

3.10. Table of sums of squares and effect sizes

If the experimental design is balanced, the level-2 parameters in Equations 3 or 5 allow for a variance decomposition to produce sums-of-squares and effect sizes. Both of these sets of statistics are important in interpreting the results of experiments in behavioral and social research. For this purpose, it is convenient to consider the BANOVA as a regression as in Equation 5, so that for $\boldsymbol{\theta}_j^p = (\theta_{j,1}^p, \theta_{j,2}^p, \ldots, \theta_{j,K}^p)$ the total sum of squares can be represented as

$$SS(\boldsymbol{\theta}_j^p) = \boldsymbol{\theta}_j^{p'}\mathbf{Z}'\boldsymbol{\beta}_j^p.\tag{19}$$

In the package, Equation 19 is estimated as $SS(\boldsymbol{\theta}_j^p) = \frac{\sum \boldsymbol{\theta}_{j,m}^{p'}\mathbf{Z}'\boldsymbol{\beta}_{j,m}^p}{M}$ where m indexes the samples from the posterior distributions of $\boldsymbol{\theta}_j^{p'}$ and $\boldsymbol{\beta}_j^p$. If the design matrix $\mathbf{Z}$ is orthogonal, then the sum of squares attributable to each of the factors and their interactions can be written in terms of the sub-matrices $\mathbf{Z}_q$ of $\mathbf{Z}$, which are associated with each factor q and its coefficients $\theta_j^{pq} = (\theta_{j,1}^{pq}, \ldots, \theta_{j,K_q}^{pq})$, $q = 1, \ldots, Q$. (e.g., Draper and Smith 1998),

$$SS(\theta_j^{pq}) = \boldsymbol{\theta}_j^{pq'}\mathbf{Z}_q'\boldsymbol{\beta}_j^p,\tag{20}$$

Where $\boldsymbol{\theta}_j^{pq}$ is the vector of coefficients of the dummy variables corresponding to main effect or interaction q. Equation 20 is estimated as $SS(\theta_j^{pq}) = \frac{\sum \boldsymbol{\theta}_{j,m}^{pq'} \mathbf{Z}_q' \beta_{j,m}^p}{M}$. For orthogonal designs it holds that

$$SS(\boldsymbol{\theta}_j^p) = SS(\theta_j^{p1}) + SS(\theta_j^{p2}) + \cdots + SS(\theta_j^{pQ}). \tag{21}$$

If the design is not balanced, type III sum-of-squares are computed(interactions included). These reflect the presence of an effect after the other main effects and interactions are accounted for, and are valid in the presence of significant interactions (Fox 1997).

The effect size measures the degree of association between a model term(e.g., a main effect, an interaction, a linear contrast) and the dependent variable, and is interpreted as the proportion of variance that is attributable to that term. Effect sizes are of eminent importance in applied research, where they are used as additional information next to statistical significance levels. There are several measures of effect size (Kirk 1982; Tabachnick and Fidell 1989). Gelman and Pardoe (Gelman and Pardoe 2006) define an R^2 for multilevel models as follows. A multilevel model with m equations can be written as:

$$\theta_k^{(m)} = X^{(m)} \beta^{(m)} + \epsilon_k^{(m)}. \tag{22}$$

$X^{(m)}$ are the predictors at level m of the model; $k = 1, \ldots, K$ indexes the observations at level m; and the errors $\epsilon_k^{(m)} \sim N(0, \sigma^{(m)})$. The $\theta_k^{(m)}$ represent batches of regression coefficients from level-1. Gelman and Pardoe now define the R^2 as follows:

$$R^{2(m)} = 1 - \frac{E[S(\epsilon_k^{(m)})]}{E[S(\theta_k^{(m)})]}. \tag{23}$$

Here $E[\cdot]$ is the posterior mean, calculated by averaging across draws in the Gibbs sampler, and $S(\cdot)$ represents the finite-sample operator, $S(x_i) = \sum_{i=1}^n (x_i - \bar{x})^2 = (n-1) \times Var(x_i)$. This R^2 measure is an adjusted R^2, because it averages over the uncertainty of the parameter estimates through the expectation operator (across draws of the Gibbs sampler).

We generalize this to a partial effect size measure as follows. At level m, the predictor variables can be partitioned as $X^{(m)} = (X_0^{(m)} | \ldots | X_l^{(m)} | \ldots | X_L^{(m)})$, into L subsets of predictors, where each subset corresponds to, for example, a set of dummy variables representing a single factor, or representing an interaction. We let $X_0^{(m)}$ denote the intercept dummy. The corresponding subsets of coefficients are denoted as $\beta_k^{(m,l)}$, and $\beta_0^{(m)}$ is the intercept, which represents the average effect of within-subject factor m. It holds that:

$$\epsilon_k^{(m)} = \theta_k^{(m)} - X^{(m)} \beta^{(m)}, \tag{24}$$

Which also defines how $\epsilon_k^{(m)}$ can be computed for each draw in the MCMC chain. We now define a set of 'partial residuals':

$$\epsilon_k^{(m,-l)} = \theta_k^{(m)} - X^{(m)} \beta^{(m)} + X_l^{(m)} \beta^{(m,l)}, \tag{25}$$

Note that at level-1 of hierarchical models for which data points, $y_{i,j}$ are non-normal, these residuals are not well-defined. Therefore, all computations are at level 2 of the model. Various sum-of-squares(SS) are defined as follows,

the total SS for equation m is:

$$SS_{Tot}^{(m)} = E[S(\theta_k^{(m)})]. \tag{26}$$

The residual SS is:

$$SS_{Res}^{(m)} = E[S(\epsilon^{(m)})]. \tag{27}$$

The SS for term l in equation m is:

$$SS_l^{(m)} = E[S(\epsilon^{(m,-l)}) - S(\epsilon^{(m)})]. \tag{28}$$

The SS for the intercept in equation m is:

$$SS_0^{(m)} = E[\theta_k^{(m)'}\theta_k^{(m)} - S(\theta_k^{(m)})]. \tag{29}$$

The latter SS of the intercept is the average effect of the within-subject factor. It is calculated from the difference of the raw and adjusted sum-of-squares of $\theta_k^{(m)}$. From these SS we can calculate a partial effect size, η_P^2, for each of the M equations at level-2 (between-subject). Partial η_p^2 have been argued to be a preferable measure (Lakens 2013). We define these partial effect sizes as:

$$\eta_{p,l}^2 = \frac{SS_l^{(m)}}{SS_l^{(m)} + SS_{Res}^{(m)}} = E\left[\frac{S(\epsilon^{(m,-l)}) - S(\epsilon^{(m)})}{S(\epsilon^{(m,-l)})}\right], \tag{30}$$

and for the intercept $\eta_{p,0}^2 = \frac{SS_0^{(m)}}{SS_0^{(m)} + SS_{Res}^{(m)}}$.

In the case of 1-level (between-subject) models, similar measures of effect size, partial η_p^2 are calculated, but using approximations to the level-1 error sum of squares (SS_{Res}) by Nakagawa and Schielzeth (2013). In the package, the function `BAnova()` performs all computations discussed above and outputs a table of sums of squares and effect sizes,

```
R> BAnova(res)
```

3.11. Table of p values

The **BANOVA** package computes Bayesian p values for posteriors of each factor (Gill 2007), which enables significance testing.

The two-sided Bayesian p value is obtained by first finding the one sided p value, or the posterior probability that the coefficient is positive, $P(\theta_{j,k}^{pq} > 0)$, or that it is negative $P(\theta_{j,k}^{pq} < 0)$, which are estimated from posterior samples as, respectively, $\frac{\sum_m (I(\theta_{j,k,m}^{pq} > 0))}{M}$, and $\frac{\sum_m (I(\theta_{j,k,m}^{pq} < 0))}{M}$. Then, the two sided p value is calculated as follows:

$$P_\theta(\theta_{j,k}^{pq}) = 2 \times \min(P(\theta_{j,k}^{pq} < 0), P(\theta_{j,k}^{pq} > 0)). \tag{31}$$

If the coefficients $\theta_{j,k_1}^{pq}, \theta_{j,k_2}^{pq}, \ldots, \theta_{j,k_J}^{pq}$ represent $J > 2$ levels of a factor, we calculate a single p value to represent the significance differences among all levels, as in standard ANOVA.

We compute the Bayesian p values in this case as follows[2]. Let $\theta^{pq}_{j,k_{min}}$ and $\theta^{pq}_{j,k_{max}}$ denote the coefficients with the smallest and largest posterior mean. Then the value is defined as

$$\min(P_\theta(\theta^{pq}_{j,k_{min}}), P_\theta(\theta^{pq}_{j,k_{max}})). \tag{32}$$

The function `table.pvalues()` in the **BANOVA** package computes p values for all factors and outputs a table of p values.

```
R> table.pvalues(res)
```

It should be noted that p values are often misinterpreted and misused, especially when "levels of significance" based on cutoff values such as 0.05 or 0.01 are reported (see for example, Nuzzo (2014); Gelman (2013)). It is beyond the scope of this article to provide a detailed guide on the proper use and interpretation of the Bayesian p values produced by the **BANOVA** package, but we emphasize that conducting repeated analyses of the same data until a certain level of significance is reached renders the reported significance levels useless. P-values should be interpreted as measures of evidence of an effect in conjunction with other statistical measures such as credible intervals and effect-sizes (See the American Statistical Association's statement on p values, Wasserstein and Lazar 2016).

4. Applications

In this section, we provide three applications of the **BANOVA** package to the analysis of previously published experimental studies. The first study by Etkin and Ratner (2012) investigated how the perceived variety among products, as means to a goal, affects peoples' motivation to pursue that goal. In this application we illustrate a between-subjects BANOVA, with dependent variables that are, respectively normal and ordered categorical. The second study, by Ferraro, Kirmani, and Matherly (2013), examines the effects of conspicuous brand usage on consumers' attitudes toward a brand. In this application, we illustrate hierarchical BANOVA models, with continuous covariates normal and t distributed dependent variables. The third study by Wedel and Pieters (2015) investigates the effects of color on the rapid gist perception of advertising. In this study, we illustrate the application of a hierarchical BANOVA with both within- and between-subjects factors, and a binomial dependent variable.

4.1. Application 1: Impact of the variety among means on motivation

In this examples we illustrate the application of the **BANOVA** package on a data from a study on goal attainment (Etkin and Ratner 2012). The study investigated how the perceived variety (high vs. low) among products, as means to a subjects' goal, affects the motivation to pursue that goal. The hypothesis was that only when progress toward a goal is low, product variety increases motivation to pursue the goal. In the study, 105 subjects were randomly assigned to one of four conditions in a 2 (goal progress: low vs. high) × 2 (variety among means: low vs. high) between-subjects design. The study used a "fitness goal", and the products used were protein bars; variety was manipulated by asking subjects to think about how the products were similar (low) or different (high); goal progress was primed by asking subjects questions

---

[2]It borrows the idea of Sheffé F test for multiple testing: the F statistic for testing the contrast with maximal difference from zero.

regarding the frequency of their recent workouts on low (0, 1, ..., 5 or more) versus high (5 or less, 6, 7, ..., 10) frequency scales. Subjects were asked questions regarding the similarity of protein bars as a manipulation check, and the bid they were willing to make for the bars, which are used as dependent variables in the study.

The data can be loaded by the following R command:

```
R> data("goalstudy")
```

The structure of the data is shown below:

```
R> head(goalstudy)

  id perceivedsim progress prodvar bid
1 1             5        1       2   5
2 2             7        1       1   0
3 3             2        2       2   1
4 4             2        2       1  15
5 5             5        2       1   3
6 6             5        1       1  10
```

The between-subjects variables are: progress, which denotes the progress toward a goal (1:low, 2:high); prodvar, which denotes the amount of variety within the means to goal attainment (1:low, 2:high); perceivedsim, which is a seven-point scale dependent variable measuring the perceived similarity of the products (1 = not at all similar, 7 = very similar); and bid which denotes the amount that subjects would be willing to pay for the products.

In the first analysis, we consider the log transform of the bid amount $(\log(\text{bid} + 1))$ as the dependent variable, assumed to follow a normal distribution. This analysis comprises a 2 (goal progress: low vs. high) $\times$ 2 (variety among means: low vs. high) between-subjects BANOVA of the bid amount. Since the study only involves a between-subjects design, the single level model should be applied. A full two-level model that incorporates a within-subject model with only an intercept is not identified. The function `BANOVA.Normal()` is used to execute the analysis (the single level model is specified through the level 1 formula). Level 1 hyper-parameters can be specified through the argument `l1_hyper` (see Section 3.5). For single level models, `l1_hyper` also includes the hyper-parameter γ, where $\beta_{j,i}^{p} \sim N(0, \gamma)$ (see Equation 2 where s_i is replaced by i).

```
R> set.seed(123)
R> goalstudy$logbid <- log(goalstudy$bid + 1)
R> app_1 <- BANOVA.Normal(logbid ~ progress * prodvar, l1_hyper =
+ c(1,1,0.0001), data = goalstudy, id = goalstudy$id, burnin = 5000,
+ sample = 1000, thin = 20)
```

The posterior means and standard deviations of the hyper parameters are reported from 1000 target samples, with a thinning factor of 20 to reduce autocorrelation, and with 5,000 samples being discarded as the burn-in period, for a total of 25,000 samples. To confirm that the chain has converged after the burn-in, the following R command outputs the Geweke's and the Heidelberg and Welch convergence diagnostics. The results are shown below,

```
R> conv.diag(app_1)
```

```
Geweke Diag. & Heidelberger and Welch's Diag.
                  Geweke stationarity test Geweke convergence p value
(Intercept)                        passed                     0.9207
progress1                          passed                     0.4004
prodvar1                           passed                     0.3560
progress1:prodvar1                 passed                     0.3451
                  H. & W. stationarity test H. & W. convergence p value
(Intercept)                         passed                      0.7390
progress1                           passed                      0.0641
prodvar1                            passed                      0.0984
progress1:prodvar1                  passed                      0.6115
```

```
The Chain has converged.
```

The above result indicates that the chains converged well before the end of the burn-in. The function `trace.plot()` provides visual diagnostics of convergence, some of the results are shown in Figure 2.

The posterior means, standard deviations, 95% credible intervals and Bayesian p values of hyper parameters are computed as follows, and the results are shown below. Note that effect sizes are calculated here using Nakagawa and Schielzeth (2013). Following standard conventions, we will call an effect 'significant' if the 95% posterior credible interval of the parameter does not cover zero.

```
R> summary(app_1)
```

```
Call:
BANOVA.Normal(l1_formula = logbid ~ progress * prodvar, data = goalstudy,
    id = goalstudy$id, l1_hyper = c(1, 1, 1e-04), burnin = 5000,
    sample = 1000, thin = 20)
```

```
Convergence diagnostics:
Geweke Diag. & Heidelberger and Welch's Diag.
                  Geweke stationarity test Geweke convergence p value
(Intercept)                        passed                     0.9207
progress1                          passed                     0.4004
prodvar1                           passed                     0.3560
progress1:prodvar1                 passed                     0.3451
                  H. & W. stationarity test H. & W. convergence p value
(Intercept)                         passed                      0.7390
progress1                           passed                      0.0641
prodvar1                            passed                      0.0984
progress1:prodvar1                  passed                      0.6115
```

```
The Chain has converged.
```

Figure 2: Trace plots of (selected) posterior distributions of the parameters of the goalstudy application.

```
Table of sum of squares:
  (Intercept) progress prodvar progress:prodvar Residuals    Total
     121.4737   0.0000  0.0000           6.6300   80.2442 207.1871

Table of effect sizes (95% credible interval):
        (Intercept)             progress            prodvar    progress:prodvar
  0.6023 (0.59,0.61) 0.0000 (-0.05,0) 0.0000 (-0.06,0) 0.0763 (0.04,0.09)

Table of p-values (Multidimensional):
  (Intercept) progress prodvar progress:prodvar
     <0.0001    0.5720  0.6560          <0.0001

Table of coefficients:
```

```
                     mean      SD Quantile0.025 Quantile0.975 p.value
(Intercept)        1.0842 0.0859        0.9182        1.2480 <0.0001
progress1          0.0490 0.0881       -0.1251        0.2262  0.5720
prodvar1           0.0373 0.0885       -0.1468        0.1987  0.6560
progress1:prodvar1 -0.2665 0.0859      -0.4260       -0.0883 <0.0001
                   Signif.codes
(Intercept)             ***
progress1
prodvar1
progress1:prodvar1      ***
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Table of predictions:

Grand mean:
1.0842
   2.5%  97.5%
 0.9182 1.2480


 progress mean    2.5%   97.5%
 1        1.1331 0.8711 1.3674
 2        1.0352 0.7906 1.261


 prodvar mean    2.5%   97.5%
 1       1.1215 0.8774 1.3419
 2       1.0469 0.7855 1.2995


 progress prodvar mean    2.5%   97.5%
 1        1         0.9039 0.5647 1.2239
 2        1         1.339  0.9716 1.6646
 1        2         1.3624 0.9966 1.7243
 2        2         0.7313 0.3961 1.0531
```

Based on the table of p values and coefficients in the above results, the interaction between variety among means (prodvar) and goal progress (progress) is significant, and the effect size is 0.076. The table of predictions, which can also be produced with the command below, shows that when goal progress was low (progress = 1), participants bid more for the products when perceived variety was high (prodvar = 2) versus low (prodvar = 1). On the contrary, when goal progress was high, participants bid more when perceived variety was low versus high.

```
R> table.predictions(app_1)
```

To predict specific values of the dependent variable, the function `predict()` in R can be applied to the objects returned by `BANOVA.*()`. For example, to predict the value of the

dependent variable for the 3rd subject in the data set, with respectively a situation of low goal progress and a high variety, the following R commands can be used:

```
R> predict(app_1, goalstudy[3, ])


      Median    2.5%    97.5%
[1,] 0.72915 0.39608 1.05315
```

To predict the value of the dependent variable for a situation of low goal progress (progress: 1) and a high variety (prodvar: 2), respectively using the entire data set, the following R commands can be used (the results are not shown):

```
R> predict(app_1, c(0, 0, 1, 2, 0, 0))
```

where all variables must have values, but only the values of 'progress' and 'prodvar' are considered. To obtain the prediction for the entire training data,

```
R> predict(app_1, goalstudy)
```

Since even the log-normal distribution may not describe the bid data very well, it could also be analyzed assuming a Poisson distribution for the bid amounts (there are only a few non-integer values which are rounded, the program will automatically convert them to integers). The following R commands constructs the BANOVA model (a two-level model is used since the model is identified in this case) and summarizes the results shown below .

```
R> set.seed(200)
R> goalstudy$bid <- as.integer(goalstudy$bid + 0.5)
R> app_1a <- BANOVA.Poisson(bid ~ 1, ~ progress * prodvar, goalstudy,
+    goalstudy$id, burnin = 5000, sample = 1000, thin = 20)
R> summary(app_1a)

Call:
BANOVA.Poisson(l1_formula = bid ~ 1, l2_formula = ~progress *
    prodvar, data = goalstudy, id = goalstudy$id, burnin = 5000,
    sample = 1000, thin = 20)

Convergence diagnostics:
Geweke Diag. & Heidelberger and Welch's Diag.
                               Geweke stationarity test
(Intercept)  :  (Intercept)                    passed
(Intercept)  :  progress1                      passed
(Intercept)  :  prodvar1                       passed
(Intercept)  :  progress1:prodvar1             passed
                               Geweke convergence p value
(Intercept)  :  (Intercept)                    0.8673
(Intercept)  :  progress1                      0.1538
(Intercept)  :  prodvar1                       0.7579
```

```
(Intercept)  :  progress1:prodvar1                           0.0121
                                H. & W. stationarity test
(Intercept)  :  (Intercept)                                  passed
(Intercept)  :  progress1                                    passed
(Intercept)  :  prodvar1                                     passed
(Intercept)  :  progress1:prodvar1                           passed
                                H. & W. convergence p value
(Intercept)  :  (Intercept)                                  0.4975
(Intercept)  :  progress1                                    0.4302
(Intercept)  :  prodvar1                                     0.9295
(Intercept)  :  progress1:prodvar1                           0.3547


The Chain has converged.


Table of sum of squares:
           (Intercept) progress prodvar progress:prodvar Residuals
(Intercept)    36.1956   0.1547  0.0000          18.8228  147.6913
               Total
(Intercept) 202.4099


Table of effect sizes (95% credible interval):
               (Intercept)           progress                 prodvar
(Intercept) 0.2044 (0.06,0.37) 8e-04 (-0.04,0.03) 0.0000 (-0.04,0.02)
            progress:prodvar
(Intercept) 0.1132 (0.04,0.2)


Table of p-values (Multidimensional):
           (Intercept) progress prodvar progress:prodvar
(Intercept)     <0.0001   0.5040  0.6740           0.0020


Table of coefficients:
                                   mean      SD Quantile0.025
(Intercept)  :  (Intercept)      0.5881 0.1564        0.2508
(Intercept)  :  progress1        0.0956 0.1390       -0.1741
(Intercept)  :  prodvar1         0.0583 0.1449       -0.2255
(Intercept)  :  progress1:prodvar1 -0.4320 0.1439    -0.7181
                                 Quantile0.975 p.value Signif.codes
(Intercept)  :  (Intercept)             0.8683 <0.0001          ***
(Intercept)  :  progress1               0.3720  0.5040
(Intercept)  :  prodvar1                0.3464  0.6740
(Intercept)  :  progress1:prodvar1     -0.1574  0.0020           **
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1


Table of predictions:


Grand mean:
```

```
3.7094
   2.5%  97.5%
 2.7128 5.3568


progress mean    2.5%   97.5%
1         4.0816 2.6784 6.5408
2         3.3711 2.2097 5.3548


prodvar mean    2.5%    97.5%
1        3.9321 2.6417 6.331
2        3.4992 2.3334 5.813



progress prodvar mean    2.5%    97.5%
1        1          2.8089 1.6215 5.2839
2        1          5.5046 3.1436 10.6318
1        2          5.9309 3.3223 11.6468
2        2          2.0645 1.1538 3.8519
```

We next analyze the manipulation check variable: the perceived similarity of the products is the dependent variable, and is expected to be different between the levels of the varmeans factor. Since it is a seven-point scale variable, an ordered multinomial distribution is used. The 2 (goal progress: low vs. high) × 2 (variety among means: low vs. high) BANOVA is executed using the function `BANOVA.ordMultinomial()` in the **BANOVA** package. Since the study only involves a between-subjects design, the single level model is used. All between-subjects factors are included in the model. The analysis is done with the following commands, the results are provided below.

```
R> set.seed(300)
R> app_2 <- BANOVA.ordMultinomial(perceivedsim ~ progress * prodvar,
+    data = goalstudy, id = goalstudy$id, burnin = 3000, sample = 1000,
+    thin = 5)
R> summary(app_2)

Call:
BANOVA.ordMultinomial(l1_formula = perceivedsim ~ progress *
    prodvar, data = goalstudy, id = goalstudy$id, burnin = 3000,
    sample = 1000, thin = 5)


Convergence diagnostics:
Geweke Diag. & Heidelberger and Welch's Diag.
                 Geweke stationarity test Geweke convergence p value
(Intercept)                        passed                     0.2757
progress1                          passed                     0.3009
prodvar1                           passed                     0.0870
progress1:prodvar1                 passed                     0.5713
```

```
                     H. & W. stationarity test H. & W. convergence p value
(Intercept)                          passed                        0.8585
progress1                            passed                        0.7689
prodvar1                             passed                        0.2051
progress1:prodvar1                   passed                        0.7204
```

The Chain has converged.

```
Table of sum of squares:
  (Intercept) progress prodvar progress:prodvar Residuals     Total
     987.7214   4.2523 78.4044           6.6463  171.0731 1248.0976
```

```
Table of effect sizes (95% credible interval):
        (Intercept)           progress             prodvar progress:prodvar
  0.8438 (0.76,0.91) 0.0234 (0,0.1) 0.3017 (0.11,0.47)  0.0355 (0,0.14)
```

```
Table of p-values (Multidimensional):
  (Intercept) progress prodvar progress:prodvar
      <0.0001   0.5700 <0.0001           0.2840
```

```
Table of coefficients:
                      mean      SD Quantile0.025 Quantile0.975 p.value
(Intercept)         3.0133 0.4309        2.2665        4.0040 <0.0001
progress1          -0.1021 0.1749       -0.4397        0.2363  0.5700
prodvar1            0.8468 0.1936        0.4622        1.2150 <0.0001
progress1:prodvar1  0.1855 0.1723       -0.1341        0.5168  0.2840
Cutpoint[2]          1.408 0.3629        0.8159        2.2446    <NA>
Cutpoint[3]         2.2899 0.4144        1.6157        3.2244    <NA>
Cutpoint[4]         3.1432 0.4489        2.3404        4.1418    <NA>
Cutpoint[5]         4.3124 0.4915        3.4314        5.4008    <NA>
Cutpoint[6]         5.7013 0.5637        4.7653        6.8732    <NA>
                    Signif.codes
(Intercept)                  ***
progress1
prodvar1                     ***
progress1:prodvar1
Cutpoint[2]                 <NA>
Cutpoint[3]                 <NA>
Cutpoint[4]                 <NA>
Cutpoint[5]                 <NA>
Cutpoint[6]                 <NA>
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1
[...]
```

The chain converged well within the burn-in period. The posterior means and standard deviations of the hyper parameters are reported from a total of 8,000 samples, with 3,000

being discarded as the burn-in period, and the remaining 5,000 samples thinned by a factor 5. From the *p* values, we can see that only the variety factor (prodvar) has a significant effect on the perceived similarity of the products and the table of sums of squares shows that the effect size is relatively large. The table of predictions (not shown in the summary above) can be produced with the following command, and the result is provided below.

```
R> table.predictions(app_2)

Table of means of the response
------------------------------

Grand mean:
4.2031
   2.5%  97.5%
 3.8889 4.5099


 progress mean   2.5%   97.5%
 1        4.113  3.6628 4.562
 2        4.2915 3.8614 4.6916

 prodvar mean   2.5%   97.5%
 1       4.9194 4.4981 5.3237
 2       3.4561 3.0323 3.8977


 progress prodvar mean   2.5%   97.5%
 1        1       4.9832 4.3987 5.5714
 2        1       4.8483 4.2947 5.377
 1        2       3.2103 2.6672 3.796
 2        2       3.7096 3.1145 4.3472
 [...]
```

From the table of predictions of the factor prodvar, we conclude that subjects perceived the products as more similar when asked to think about how they were similar versus different, which supports the experimental manipulation and is consistent with Etkin and Ratner (2012). The function `table.predictions()` in the case of an ordered categorical variable also provides a more detailed table with the probabilities of each response category. For example, the table of probabilities corresponding to response 1 (not at all similar) of the variable prodvar, is shown below (the remainder of the output for the other six response categories is not shown). From the table, we can see that subjects are more likely to provide the response value 1 when perceived variety was high, which is in line with the results above.

```
Table of probabilities for each category of the response
--------------------------------------------------------

Response :  1
```

```
Grand mean:
0.0504
   2.5%  97.5%
 0.0176 0.0939


 progress mean    2.5%    97.5%
 1        0.056  0.0193 0.1054
 2        0.0466 0.015  0.0964


 prodvar mean    2.5%    97.5%
 1       0.0231 0.0071 0.0492
 2       0.1094 0.0413 0.1954



 progress prodvar mean    2.5%    97.5%
 1        1       0.0219 0.006  0.0502
 2        1       0.0257 0.007  0.0607
 1        2       0.142  0.0489 0.2717
 2        2       0.087  0.0278 0.1786
```

To predict means corresponding to the first two data points in `goalstudy`, the `predict()` command is used, which outputs the probabilities for each category of each data point.

```
R> predict(app_2, goalstudy[1:2, ])
```

```
       Sample number Response  Median    2.5%   97.5%
 [1,]              1        1 0.13388 0.04889 0.27168
 [2,]              1        2 0.24176 0.13774 0.38059
 [3,]              1        3 0.20875 0.12556 0.31200
 [4,]              1        4 0.17230 0.09704 0.26221
 [5,]              1        5 0.13830 0.07083 0.22450
 [6,]              1        6 0.05772 0.02522 0.12334
 [7,]              1        7 0.02146 0.00724 0.05610
 [8,]              2        1 0.01944 0.00599 0.05017
 [9,]              2        2 0.05326 0.02177 0.11453
[10,]              2        3 0.08512 0.03998 0.16265
[11,]              2        4 0.14373 0.07451 0.23643
[12,]              2        5 0.26749 0.18634 0.37241
[13,]              2        6 0.25146 0.15141 0.38959
[14,]              2        7 0.14832 0.06601 0.29467
```

The **JAGS** code for the above model generated by the program can be easily retrieved by the following R command.

```
R> cat(app_2$JAGSmodel)
```

```
model{
  for (i in 1:n){
    y[i] ~ dcat(P[i,])
    P[i,1] <- 1 - Q[i,1]
    for (i.cut in 2: n.cut) {
      P[i,i.cut] <- Q[i,i.cut-1] - Q[i,i.cut]
    }
    P[i,n.cut+1] <- Q[i,n.cut]
    for (i.cut in 1:n.cut){
      logit(Q[i,i.cut]) <- z[i,i.cut]
      z[i,i.cut] <-beta1*X[i,1]+beta2*X[i,2]+beta3*X[i,3]+beta4*X[i,4]-
      cutp[i.cut]*(1-equals(i.cut,1))
    }
  }
  beta1~dnorm(0,1e-04)
  beta2~dnorm(0,1e-04)
  beta3~dnorm(0,1e-04)
  beta4~dnorm(0,1e-04)
  for (i.cut in 1: n.cut) {
    cutp0[i.cut] ~ dnorm(0,tau.cut)
  }
  tau.cut ~ dunif(0,100)
  cutp[1:n.cut] <- sort(cutp0)
}
```

Note that, for the convenience of generation of the **JAGS** code, the program uses a uniform naming scheme for all level 1 and level 2 parameters which are different from the names in the original data.

4.2. Application 2: Conspicuous brand usage

We next illustrate the **BANOVA** package on data from a study that examines consumers' attitudes toward a brand after seeing another consumer conspicuously using it (Ferraro *et al.* 2013). Conspicuous brand usage occurs when a consumer draws attention to a brand she uses by flaunting (Ferraro *et al.* 2013). 154 subjects from an online panel participated in the study. Conspicuousness was manipulated as a between-subjects factor, by exposing subjects to a forty-five seconds video in which the conspicuous usage of the brand (Apple ipad) was manipulated (low vs high conspicuousness). In addition, the so called self-brand connection was measured: this refers to the extent to which a consumers' own self-concept matches the image she has of a certain brand. Brand attitude was calculated as the average of three seven-point scale questions (dislike/like, unfavorable/favorable and bad/good). The relation between conspicuousness, self-brand connection, and brand attitudes was investigated. The analysis aims to test the hypothesis that there are negative effects of conspicuous brand usage on the attitudes toward the brand, only for subjects that have a low self-brand connection.

This example illustrates a BANOVA model with continuous covariates. Brand attitude is considered as a continuous dependent variable, assumed to follow a normal distribution. Data of this study can be loaded by the following R command:

```
R> data("ipadstudy")
```

The data is displayed in the long format including only responses and between-subjects variables.

```
R> head(ipadstudy)
```

```
  id attitude owner age gender conspic  selfbrand apple_dl
1  1 3.000000     0  19      0       0 -2.3042672        3
2  2 5.333333     0  33      0       1  1.6957328        6
3  3 5.666667     0  25      1       0 -0.1614100        6
4  4 5.333333     1  41      0       1 -0.4471243        5
5  5 6.000000     1  38      1       1  0.2671614        6
6  6 4.000000     0  33      1       0  0.6957328        4
```

The between-subjects variables are: selfbrand, which is a numerical variable (mean centered) representing self-brand connection; and conspic, which is a two-level factor corresponding to the two levels of conspicuousness (0: low, 1: high). There are a number of control variables: owner denotes whether participants owned the product (1: yes, 0: no); age in years; and gender (1: female, 0: male). Finally id is the identification number of the subjects.

The BANOVA uses as a dependent variable the attitude toward the brand, measured by averaging answers on three seven-point scales. The model can be estimated using the function `BANOVA.Normal()` in the **BANOVA** package. Since it is a between-subjects design without repeated measurements, the within-subjects model is not included. The between-subjects covariates include owner, age, gender, selfbrand and the interaction between conspic and selfbrand. The two-level factor conspic is effect-coded and the one-way between-subjects BANOVA is specified as follows (note that the covariates must be mean centered: the program will automatically do so).

```
R> set.seed(400)
R> app_3 <- BANOVA.Normal(attitude ~ owner + age + gender + selfbrand *
+    conspic, l1_hyper = c(1,1,0.0001), data = ipadstudy, id = ipadstudy$id,
+    burnin = 5000, sample = 1000, thin = 10)
```

The posterior means and standard deviations of the hyper parameters are reported from a total 15,000 samples, with 5,000 being discarded as the burn-in period, and the remainder thinned by a factor 10. The chain converged well within the burn-in period. The table of sums-of-squares, effect sizes and p values, as well as the posterior means, standard deviations, 95% credible intervals, and p values of hyper parameters are computed as follows, the results are shown below.

```
R> summary(app_3)
```

```
Call:
BANOVA.Normal(l1_formula = attitude ~ owner + age + gender +
    selfbrand * conspic, data = ipadstudy, id = ipadstudy$id,
    l1_hyper = c(1, 1, 1e-04), burnin = 5000, sample = 1000, thin = 10)
```

```
Convergence diagnostics:
Geweke Diag. & Heidelberger and Welch's Diag.
                    Geweke stationarity test Geweke convergence p value
(Intercept)                          passed                     0.8680
owner1                               passed                     0.8201
age                                  passed                     0.0341
gender1                              passed                     0.3474
selfbrand                            passed                     0.3301
conspic1                             passed                     0.3339
selfbrand:conspic1                   passed                     0.9964
                    H. & W. stationarity test H. & W. convergence p value
(Intercept)                          passed                     0.5686
owner1                               passed                     0.9509
age                                  passed                     0.0581
gender1                              passed                     0.4634
selfbrand                            passed                     0.4829
conspic1                             passed                     0.3842
selfbrand:conspic1                   passed                     0.9657

The Chain has converged.

Table of sum of squares:
  (Intercept)   owner    age gender selfbrand conspic selfbrand:conspic
   4270.0702 10.8921 0.0000 0.0000  151.1170  6.8708            4.9632
  Residuals     Total
   191.5908 4677.7778

Table of effect sizes (95% credible interval):
        (Intercept)                 owner                 age           gender
  0.9571 (0.95,0.96) 0.0535 (0.02,0.09) 0.0000 (-0.03,0) 0.0000 (-0.04,0)
          selfbrand              conspic selfbrand:conspic
  0.4403 (0.39,0.48) 0.0346 (0.01,0.04)   0.0252 (0,0.04)

Table of p-values (Multidimensional):
  (Intercept)  owner    age gender selfbrand conspic selfbrand:conspic
      <0.0001 0.0060 0.8840 0.8540   <0.0001  0.0160            0.0200

Table of coefficients:
                    mean     SD Quantile0.025 Quantile0.975 p.value
(Intercept)       5.3832 0.1010        5.1850        5.5782 <0.0001
owner1           -0.3058 0.1088       -0.5084       -0.0854  0.0060
age              -0.0009 0.0100       -0.0190        0.0191  0.8840
gender1           0.0180 0.0972       -0.1786        0.2080  0.8540
selfbrand         0.6272 0.0629        0.4995        0.7476 <0.0001
conspic1          0.2304 0.0892        0.0564        0.3922  0.0160
selfbrand:conspic1 -0.1298 0.0583      -0.2433       -0.0198  0.0200
```

```
                Signif.codes
(Intercept)               ***
owner1                     **
age
gender1
selfbrand                 ***
conspic1                    *
selfbrand:conspic1          *
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1


Table of predictions:


Grand mean:
5.3832
   2.5%  97.5%
 5.1850 5.5782



 owner mean    2.5%    97.5%
 0     5.0774 4.8465 5.3033
 1     5.689  5.3597 6.0168


 gender mean   2.5%    97.5%
 0      5.4012 5.1007 5.6919
 1      5.3652 5.1031 5.6141


 conspic mean   2.5%    97.5%
 0       5.6136 5.344  5.8727
 1       5.1528 4.8807 5.4065
```

Based on the above estimates, conspicuousness, self-brand connection and the interaction: conspicuousness × self-brand connection, significantly affect the attitude towards the brand, consistent with Ferraro *et al.* (2013). Note that since since there are continuous covariates, a type III analysis of variance is used, so that the sum of squares (and effect size) for each effect is computed conditional upon all other effects. Thus they do not add up to the total sum of squares.

In this application, the distribution of the dependent variable is continuous, but it may have fatter tails than the normal. The function `BANOVA.T()` can be applied to construct a BANOVA model in which the response variable is assumed to follow a student's t distribution. This permits (weakly) robust inference (Bernardo and Giron 1992), as it allows for fatter tails and outliers in the data. The results, shown below, are similar to those in the results above which supports their robustness, and are not discussed further here.

```
R> set.seed(600)
R> app_4 <- BANOVA.T(attitude ~ owner + age + gender + selfbrand * conspic,
+    l1_hyper = c(1, 1, 1, 0.0001), data = ipadstudy, id = ipadstudy$id,
```

```
+     burnin = 3000, sample = 1000, thin = 5)
R> summary(app_4)

Call:
BANOVA.T(l1_formula = attitude ~ owner + age + gender + selfbrand *
    conspic, data = ipadstudy, id = ipadstudy$id, l1_hyper = c(1,
    1, 1, 1e-04), burnin = 3000, sample = 1000, thin = 5)
```

Convergence diagnostics:
Geweke Diag. & Heidelberger and Welch's Diag.

	Geweke stationarity test	Geweke convergence p value
(Intercept)	passed	0.5343
owner1	passed	0.7222
age	passed	0.1984
gender1	passed	0.9594
selfbrand	passed	0.7945
conspic1	passed	0.9094
selfbrand:conspic1	passed	0.2856
	H. & W. stationarity test	H. & W. convergence p value
(Intercept)	passed	0.7958
owner1	passed	0.3437
age	passed	0.2535
gender1	passed	0.6414
selfbrand	passed	0.5905
conspic1	passed	0.7354
selfbrand:conspic1	passed	0.3141

The Chain has converged.

Table of sum of squares:

(Intercept)	owner	age	gender	selfbrand	conspic	selfbrand:conspic
4270.0702	12.8796	0.0000	0.0000	148.2134	6.9809	5.2258

Residuals	Total
192.3997	4677.7778

Table of effect sizes (95% credible interval):

(Intercept)	owner	age	gender
0.9569 (0.95,0.96)	0.0624 (0.03,0.09)	0.0000 (-0.04,0)	0.0000 (-0.04,0)

selfbrand	conspic	selfbrand:conspic
0.4345 (0.39,0.47)	0.0350 (0.01,0.04)	0.0264 (0,0.04)

Table of p-values (Multidimensional):

(Intercept)	owner	age	gender	selfbrand	conspic	selfbrand:conspic
<0.0001	<0.0001	0.8320	0.6920	<0.0001	0.0060	0.0500

Table of coefficients:

	mean	SD	Quantile0.025	Quantile0.975	p.value

```
(Intercept)             5.4312 0.0967        5.2456        5.6154 <0.0001
owner1                 -0.3402 0.0987       -0.5329       -0.1460 <0.0001
age                     0.0019 0.0097       -0.0169        0.0202  0.8320
gender1                -0.0366 0.0892       -0.2182        0.1300  0.6920
selfbrand               0.5849 0.0614        0.4704        0.7053 <0.0001
conspic1                0.2305 0.0911        0.0633        0.4140  0.0060
selfbrand:conspic1     -0.1058 0.0562       -0.2174        0.0000  0.0500
                    Signif.codes
(Intercept)              ***
owner1                   ***
age
gender1
selfbrand                ***
conspic1                  **
selfbrand:conspic1         *
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Table of predictions:

Grand mean:
5.4312
   2.5%  97.5%
 5.2456 5.6154


 owner mean    2.5%    97.5%
 0     5.091  4.8411 5.3216
 1     5.7714 5.4671 6.0782

 gender mean   2.5%    97.5%
 0      5.3946 5.1216 5.6701
 1      5.4678 5.2095 5.6989

 conspic mean   2.5%    97.5%
 0       5.6617 5.4181 5.9206
 1       5.2007 4.9239 5.4655
```

4.3. Application 3: Gist perception of advertising

Finally, we illustrate the application of the **BANOVA** package in a study into the influence of color on gist perception of advertising, which is the very rapid identification of ads during brief exposures. Specifically, we analyze the effect of color on the perception of the gist of ads when the advertising exposure is brief and blurred (Wedel and Pieters 2015). In the study, 116 subjects were randomly assigned to one condition of a 5 (blur: normal, low, medium, high, very high) × 2 (color: full color, grayscale) between-subjects, × 2 (image: typical ads,

atypical ads) within-subjects, mixed design. Participants were exposed to 40 images, 32 full-page ads and 8 editorial pages. There were 8 ads per product category, with 4 typical and 4 atypical ones. Blur was manipulated by processing the advertising images with a Gaussian blur filter of different radius. Subjects were flashed an image for 100msec. and then asked to identify whether the image was an ad or not.

The data included in the package can be loaded into R using the `data()` function, i.e., using the following R code:

```
R> data("colorad")
```

The structure of `colorad` is shown below using the `head()` function. It is in long format including both within- subjects and between- subjects variables.

```
R> head(colorad)
```

```
   id typic  y blurfac color   blur
1  1     0  8       2     1 3.6889
2  1     1  6       2     1 3.6889
3  2     0 12       4     0 4.7875
4  2     1  6       4     0 4.7875
5  3     0 11       2     0 3.6889
6  3     1  9       2     0 3.6889
```

Here, the within-subjects variable typic is a factor with 2 levels '0' (typical ads) and '1'(atypical ads); between-subjects variables are: blur, a numerical variable representing the blur of the image (the log-radius of a Gaussian blur filter used to produce the images); blurfac, a factor variable with the five levels of blur; and color, a factor representing the color of the ads with 2 levels '0'(full color) and '1'(grayscale). id is the subject identification number. The dependent variable is the number of times ads were correctly identified as an ad, out of the 16 ads, for each subject for each level of typic.

We are interested in the effects of within- and between- subjects factors typic and color, and the variable blur, as well as their interactions. The factor typic varies within individuals; the factors blur, color and blur × color interaction vary between subjects.

The analysis of this experiment is executed with the function `BANOVA.Binomial()` in the **BANOVA** package (the continuous covariate blur is mean centered by default). The R code to implement the analysis is shown below. For this model, the **BANOVA** function call needs to include the binomial total (16) as an additional argument.

```
R> data("colorad")
R> set.seed(700)
R> app_5 <- BANOVA.Binomial(y ~ typic, ~ color * blur, colorad, colorad$id,
+    as.integer(16), burnin = 3000, sample = 2000, thin = 5)
```

The posterior means and standard deviations of the hyper parameters are reported from a total of 13,000 samples, with 3,000 being discarded as the burn-in period, and the remainder thinned by a factor 5 to reduce autocorrelation. To confirm that the chain has converged after the burn-in, the `conv.diag(app_5)` command outputs the convergence diagnostics and

`trace.plot(app_5)` provides visual diagnostics of convergence (the results are not shown here, but indicate that the chains converged well before the end of the burn-in).

The posterior means, standard deviations, 95% credible intervals and p values of hyper parameters are computed with the following command, and the results are shown below. Note that each of these tables now have two rows, one for each between-subject model (intercept and typic).

```
R> summary(app_5)

Call:
BANOVA.Binomial(l1_formula = y ~ typic, l2_formula = ~color *
    blur, data = colorad, id = colorad$id, num_trials = as.integer(16),
    burnin = 3000, sample = 2000, thin = 5)

Convergence diagnostics:
Geweke Diag. & Heidelberger and Welch's Diag.
                              Geweke stationarity test
(Intercept)  :  (Intercept)                   passed
(Intercept)  :  color1                        passed
(Intercept)  :  blur                          passed
(Intercept)  :  color1:blur                   passed
typic1  :  (Intercept)                         passed
typic1  :  color1                              passed
typic1  :  blur                                passed
typic1  :  color1:blur                         passed
                              Geweke convergence p value
(Intercept)  :  (Intercept)                   0.5803
(Intercept)  :  color1                        0.1613
(Intercept)  :  blur                          0.2185
(Intercept)  :  color1:blur                   0.0761
typic1  :  (Intercept)                         0.8919
typic1  :  color1                              0.2779
typic1  :  blur                                0.1391
typic1  :  color1:blur                         0.3443
                              H. & W. stationarity test
(Intercept)  :  (Intercept)                   passed
(Intercept)  :  color1                        passed
(Intercept)  :  blur                          passed
(Intercept)  :  color1:blur                   passed
typic1  :  (Intercept)                         passed
typic1  :  color1                              passed
typic1  :  blur                                passed
typic1  :  color1:blur                         passed
                              H. & W. convergence p value
(Intercept)  :  (Intercept)                   0.5140
(Intercept)  :  color1                        0.3020
(Intercept)  :  blur                          0.3020
```

```
(Intercept)  :  color1:blur                    0.0762
typic1  :  (Intercept)                          0.1179
typic1  :  color1                               0.5091
typic1  :  blur                                 0.0536
typic1  :  color1:blur                          0.5938


The Chain has converged.


Table of sum of squares:
           (Intercept)  color    blur color:blur Residuals    Total
(Intercept)    59.2977 2.4447 25.9849     8.7501  138.0372 228.8865
typic1         21.9561 1.8509  5.4538     6.9776   20.0264  50.5066


Table of effect sizes (95% credible interval):
                (Intercept)              color              blur
(Intercept) 0.3010 (0.25,0.36) 0.0164 (-0.01,0.1) 0.1586 (0.11,0.21)
typic1       0.5228 (0.4,0.64) 0.0769 (-0.01,0.3)  0.2134 (0.1,0.36)
                  color:blur
(Intercept) 0.0559 (-0.01,0.21)
typic1         0.2372 (0,0.51)


Table of p-values (Multidimensional):
           (Intercept)  color    blur color:blur
(Intercept)    <0.0001 0.9960 <0.0001     0.1840
typic          <0.0001 0.3970 <0.0001     0.0290


Table of coefficients:
                            mean     SD Quantile0.025 Quantile0.975
(Intercept)  :  (Intercept)  0.4963 0.0566        0.3857        0.6093
(Intercept)  :  color1      -0.0029 0.1243       -0.2568        0.2432
(Intercept)  :  blur        -0.1710 0.0300       -0.2301       -0.1106
(Intercept)  :  color1:blur  0.0394 0.0300       -0.0219        0.0984
typic1  :  (Intercept)       0.3012 0.0343        0.2335        0.3681
typic1  :  color1           -0.0598 0.0710       -0.1990        0.0823
typic1  :  blur             -0.0774 0.0176       -0.1122       -0.0431
typic1  :  color1:blur       0.0385 0.0170        0.0058        0.0712
                           p.value Signif.codes
(Intercept)  :  (Intercept) <0.0001          ***
(Intercept)  :  color1       0.9960
(Intercept)  :  blur        <0.0001          ***
(Intercept)  :  color1:blur  0.1840
typic1  :  (Intercept)      <0.0001          ***
typic1  :  color1            0.3970
typic1  :  blur             <0.0001          ***
typic1  :  color1:blur       0.0290            *
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1
```

```
Table of predictions:

Grand mean:
0.6216
   2.5%  97.5%
 0.5953 0.6478


 typic mean    2.5%    97.5%
 0      0.6894 0.6595 0.7172
 1      0.5486 0.5178 0.5794


 color mean    2.5%    97.5%
 0      0.6209 0.5531 0.6828
 1      0.6223 0.5561 0.6827


 typic color mean    2.5%    97.5%
 0     0      0.6759 0.5977 0.7426
 0     1      0.7027 0.6319 0.7656
 1     0      0.5627 0.4873 0.6323
 1     1      0.5345 0.4634 0.605
```

Based on the above estimates, ad identification is significantly influenced by ad typicality (typic): typical ads are identified more accurately as ads, compared to atypical ads. The accuracy of ad identification is also affected by the degree of blur and its interaction with typic. The three-factor interaction (blur $\times$ color $\times$ typic) is significant, which reveals that color protects the identification of typical ads against blur, which is in line with the findings of Wedel and Pieters (2015).

These results are based on the BANOVA model with blur as a continuous covariate. To further understand the effects of blur, we can use the discrete variable blur (blurfac) in a two-way BANOVA at the between-subjects level (and the factor typic again within-subjects), using the following command:

```
R> set.seed(900)
R> app_6 <- BANOVA.Binomial(y ~ typic, ~ color * blurfac, colorad,
+    colorad$id, as.integer(16), burnin = 20000, sample = 3000, thin = 5)
```

Since the above model involves more parameters, a larger number of burn-in and target samples are used to ensure the chains for all parameters converge: a total of 35,000. The table of sums of squares, effect sizes and p values, as well as the posterior means, standard deviations, 95% credible intervals and p values of the parameters are produced with the function `summary()`, the results are presented below.

```
R> summary(app_6)
```

```
Call:
BANOVA.Binomial(l1_formula = y ~ typic, l2_formula = ~color *
    blurfac, data = colorad, id = colorad$id, num_trials = as.integer(16),
    burnin = 20000, sample = 3000, thin = 5)
```

```
Convergence diagnostics:
Geweke Diag. & Heidelberger and Welch's Diag.
                              Geweke stationarity test
(Intercept)  :  (Intercept)                   passed
(Intercept)  :  color1                        passed
(Intercept)  :  blurfac1                      passed
(Intercept)  :  blurfac2                      passed
(Intercept)  :  blurfac3                      passed
(Intercept)  :  blurfac4                      passed
(Intercept)  :  color1:blurfac1               passed
(Intercept)  :  color1:blurfac2               passed
(Intercept)  :  color1:blurfac3               passed
(Intercept)  :  color1:blurfac4               passed
typic1  :  (Intercept)                         passed
typic1  :  color1                              passed
typic1  :  blurfac1                            passed
typic1  :  blurfac2                            passed
typic1  :  blurfac3                            passed
typic1  :  blurfac4                            passed
typic1  :  color1:blurfac1                     passed
typic1  :  color1:blurfac2                     passed
typic1  :  color1:blurfac3                     passed
typic1  :  color1:blurfac4                     passed
                              Geweke convergence p value
(Intercept)  :  (Intercept)                    0.2356
(Intercept)  :  color1                         0.3940
(Intercept)  :  blurfac1                       0.3836
(Intercept)  :  blurfac2                       0.8752
(Intercept)  :  blurfac3                       0.0345
(Intercept)  :  blurfac4                       0.3804
(Intercept)  :  color1:blurfac1                0.5241
(Intercept)  :  color1:blurfac2                0.3314
(Intercept)  :  color1:blurfac3                0.0385
(Intercept)  :  color1:blurfac4                0.4250
typic1  :  (Intercept)                          0.7581
typic1  :  color1                               0.5452
typic1  :  blurfac1                             0.5190
typic1  :  blurfac2                             0.0406
typic1  :  blurfac3                             0.3353
typic1  :  blurfac4                             0.7254
typic1  :  color1:blurfac1                      0.1974
typic1  :  color1:blurfac2                      0.1124
```

```
typic1  :  color1:blurfac3                            0.2070
typic1  :  color1:blurfac4                            0.0933
                                 H. & W. stationarity test
(Intercept)  :  (Intercept)                            passed
(Intercept)  :  color1                                 passed
(Intercept)  :  blurfac1                               passed
(Intercept)  :  blurfac2                               passed
(Intercept)  :  blurfac3                               passed
(Intercept)  :  blurfac4                               passed
(Intercept)  :  color1:blurfac1                        passed
(Intercept)  :  color1:blurfac2                        passed
(Intercept)  :  color1:blurfac3                        passed
(Intercept)  :  color1:blurfac4                        passed
typic1  :  (Intercept)                                 passed
typic1  :  color1                                      passed
typic1  :  blurfac1                                    passed
typic1  :  blurfac2                                    passed
typic1  :  blurfac3                                    passed
typic1  :  blurfac4                                    passed
typic1  :  color1:blurfac1                             passed
typic1  :  color1:blurfac2                             passed
typic1  :  color1:blurfac3                             passed
typic1  :  color1:blurfac4                             passed
                               H. & W. convergence p value
(Intercept)  :  (Intercept)                            0.4061
(Intercept)  :  color1                                 0.5476
(Intercept)  :  blurfac1                               0.8945
(Intercept)  :  blurfac2                               0.4246
(Intercept)  :  blurfac3                               0.0972
(Intercept)  :  blurfac4                               0.3817
(Intercept)  :  color1:blurfac1                        0.3015
(Intercept)  :  color1:blurfac2                        0.2675
(Intercept)  :  color1:blurfac3                        0.2377
(Intercept)  :  color1:blurfac4                        0.8629
typic1  :  (Intercept)                                 0.6026
typic1  :  color1                                      0.1678
typic1  :  blurfac1                                    0.3973
typic1  :  blurfac2                                    0.1046
typic1  :  blurfac3                                    0.4756
typic1  :  blurfac4                                    0.4426
typic1  :  color1:blurfac1                             0.6945
typic1  :  color1:blurfac2                             0.1133
typic1  :  color1:blurfac3                             0.4469
typic1  :  color1:blurfac4                             0.1558


The Chain has converged.
```

```
Table of sum of squares:
          (Intercept) color blurfac color:blurfac Residuals    Total
(Intercept)     59.2418 5.2392 29.3149         1.0919  137.2747 232.0625
typic1          21.5893 2.0443 13.7377         1.9909   16.6845  55.9163


Table of effect sizes (95% credible interval):
                (Intercept)                color          blurfac
(Intercept) 0.3020 (0.25,0.36) 0.0369 (0.01,0.07) 0.1762 (0.12,0.23)
typic1      0.5634 (0.44,0.68) 0.1083 (0.02,0.24) 0.4504 (0.31,0.59)
                color:blurfac
(Intercept) 0.0080 (-0.03,0.04)
typic1      0.1058 (0.01,0.24)


Table of p-values (Multidimensional):
          (Intercept)  color blurfac color:blurfac
(Intercept)     <0.0001 0.0073 <0.0001        0.1527
typic           <0.0001 0.0020 <0.0001        0.0220


Table of coefficients:
                               mean     SD Quantile0.025
(Intercept)  :  (Intercept)    0.4966 0.0572        0.3878
(Intercept)  :  color1         0.1542 0.0561        0.0457
(Intercept)  :  blurfac1       0.5556 0.1103        0.3468
(Intercept)  :  blurfac2       0.0511 0.1188       -0.1911
(Intercept)  :  blurfac3       0.0204 0.1090       -0.1929
(Intercept)  :  blurfac4      -0.0489 0.1110       -0.2608
(Intercept)  :  color1:blurfac1 -0.1618 0.1114     -0.3784
(Intercept)  :  color1:blurfac2  0.1162 0.1179     -0.1072
(Intercept)  :  color1:blurfac3 -0.0798 0.1080     -0.2902
(Intercept)  :  color1:blurfac4  0.0142 0.1098     -0.2022
typic1  :  (Intercept)          0.3105 0.0321        0.2505
typic1  :  color1               0.0917 0.0319        0.0313
typic1  :  blurfac1             0.1602 0.0664        0.0296
typic1  :  blurfac2             0.3655 0.0686        0.2328
typic1  :  blurfac3            -0.0254 0.0613       -0.1461
typic1  :  blurfac4            -0.1670 0.0618       -0.2874
typic1  :  color1:blurfac1     -0.1480 0.0664       -0.2793
typic1  :  color1:blurfac2      0.0357 0.0677       -0.0913
typic1  :  color1:blurfac3     -0.0201 0.0614       -0.1395
typic1  :  color1:blurfac4      0.0959 0.0599       -0.0219
                              Quantile0.975 p.value Signif.codes
(Intercept)  :  (Intercept)          0.6146 <0.0001          ***
(Intercept)  :  color1               0.2657  0.0073           **
(Intercept)  :  blurfac1             0.7732 <0.0001          ***
(Intercept)  :  blurfac2             0.2795  0.6487
(Intercept)  :  blurfac3             0.2316  0.8333
(Intercept)  :  blurfac4             0.1727  0.6607
```

```
(Intercept)  :  color1:blurfac1        0.0612  0.1527
(Intercept)  :  color1:blurfac2        0.3467  0.3280
(Intercept)  :  color1:blurfac3        0.1415  0.4607
(Intercept)  :  color1:blurfac4        0.2306  0.9160
typic1  :  (Intercept)                 0.3764 <0.0001       ***
typic1  :  color1                      0.1564  0.0020       **
typic1  :  blurfac1                    0.2873  0.0113        *
typic1  :  blurfac2                    0.5010 <0.0001       ***
typic1  :  blurfac3                    0.0931  0.6900
typic1  :  blurfac4                   -0.0454  0.0067       **
typic1  :  color1:blurfac1           -0.0225  0.0220        *
typic1  :  color1:blurfac2            0.1686  0.5993
typic1  :  color1:blurfac3            0.0997  0.7473
typic1  :  color1:blurfac4            0.2134  0.1120
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Table of predictions:

Grand mean:
0.6217
   2.5%  97.5%
 0.5957 0.6490


 typic mean    2.5%    97.5%
 0     0.6915 0.664   0.7195
 1     0.5464 0.5147  0.5789


 color mean    2.5%    97.5%
 0     0.6572 0.6208  0.6933
 1     0.5848 0.5461  0.6209

 blurfac mean    2.5%    97.5%
 1       0.7412 0.6924 0.7851
 2       0.6336 0.5689 0.6925
 3       0.6265 0.5704 0.6804
 4       0.6101 0.5528 0.6656
 5       0.4796 0.4181 0.5409


 typic color mean     2.5%    97.5%
 0     0      0.7414 0.7049 0.7762
 0     1      0.6367 0.5942 0.6764
 1     0      0.5618 0.517  0.6059
 1     1      0.5309 0.4856 0.5738
```

typic	blurfac	mean	2.5%	97.5%
0	1	0.821	0.7737	0.8593
0	2	0.7727	0.7116	0.8213
0	3	0.6905	0.6291	0.7469
0	4	0.6437	0.5777	0.7047
0	5	0.4739	0.4036	0.5452
1	1	0.6414	0.5757	0.7017
1	2	0.4679	0.3944	0.5439
1	3	0.5577	0.4919	0.6225
1	4	0.5755	0.5095	0.6389
1	5	0.4853	0.4136	0.5554

color	blurfac	mean	2.5%	97.5%
0	1	0.7397	0.6687	0.8002
1	1	0.7427	0.6693	0.8044
0	2	0.6938	0.6062	0.7691
1	2	0.5689	0.4783	0.657
0	3	0.6437	0.5616	0.7161
1	3	0.6089	0.5273	0.6845
0	4	0.6493	0.5674	0.7217
1	4	0.5694	0.4852	0.6491
0	5	0.5459	0.4594	0.6289
1	5	0.4141	0.3319	0.502

typic	color	blurfac	mean	2.5%	97.5%
0	0	1	0.8114	0.7414	0.8678
0	1	1	0.8302	0.7624	0.8793
0	0	2	0.835	0.7587	0.8893
0	1	2	0.6955	0.6008	0.7766
0	0	3	0.7208	0.6342	0.7913
0	1	3	0.6584	0.5682	0.7387
0	0	4	0.7206	0.6333	0.7929
0	1	4	0.5585	0.4631	0.6526
0	0	5	0.5718	0.4754	0.6612
0	1	5	0.378	0.2852	0.4804
1	0	1	0.6525	0.5611	0.7313
1	1	1	0.6301	0.5357	0.7166
1	0	2	0.5037	0.3973	0.6069
1	1	2	0.4326	0.3351	0.5386
1	0	3	0.5584	0.4619	0.6499
1	1	3	0.557	0.4647	0.6443
1	0	4	0.5708	0.4749	0.6607
1	1	4	0.5802	0.4853	0.6681
1	0	5	0.5197	0.4195	0.6171
1	1	5	0.4511	0.354	0.5524

We first inspect the tables of sums of squares and effect sizes. In these tables, the columns denote between-subjects factors and the rows denote the within-subjects factors. The values in the table present the sum-of-squares and effect sizes of the effects of these factors. Again, the accuracy of ad identification is affected by blur, and to a lesser extent by color. From the tables of p values, ad typicality (the value corresponding to the row name 'typic' and column name '(Intercept)') and the degree of blur (the value corresponding to the row name '(Intercept)' and column name 'blurfac') are again highly significant. There is also support for the main effect of color. The three-factor interaction (blurfac $\times$ color $\times$ typic) is also significant, which again shows that color protects the identification of typical ads against blur (Wedel and Pieters 2015). The conclusions from the table of estimates are similar to those from the results of the previous model, but this table for nonlinear effects of blur allows us to inspect the effects of each level of bur, and the interactive effects with color and typicality.

Through the tables of predictions for all factors and their interactions, we can inspect these effects in more detail. We can see that typical color ads (typic = 0, color = 0) are always more accurately identified than atypical color ads (typic = 1, color = 0). Typical grayscale ads (typic = 0, color = 1, blur = 1, . . . ,5), however, are only more accurately identified than atypical grayscale ads (typic = 1, color = 1, blur = 1, . . . ,5) when there is no blur, or a low level of blur (Wedel and Pieters 2015).

5. Discussion

This paper has introduced a general framework for Bayesian analysis of variance, and an accompanying R package **BANOVA**. This package can be used to analyze experimental data with a wide variety of hierarchical Bayesian models. The response variable can be normal, student's t, binomial, multinomial or Poisson. The between-subjects model follows a traditional ANOVA or ANCOVA structure that allows the estimation of sums of squares and effect sizes of experimental factors and their interactions. These are easy to interpret for the applied researcher because of their similarity with standard ANOVA output. In the social and behavioral sciences there is an abundance of research that lends itself to the application of **BANOVA** analyses, which are in most cases considerably more appropriate than classical ANOVA. We hope that the availability of user-friendly software in the form of the **BANOVA** package will stimulate the analysis of these studies under more reasonable assumptions on the distribution of the data and its hierarchical structure. If the data was obtained from a between subject design and the dependent variable is approximately normal, the results of **BANOVA** will be very similar to those of ANOVA.

There are a number of other R packages that can be used to fit hierarchical Bayes models, including **BACC** (Geweke 1999), **bayesm** (Rossi 2017) and **MCMCpack** (Martin, Quinn, and Park 2011) as well as the **WinBUGS**, **OpenBUGS** and **JAGS** software. However, all these either require the user to be familiar with Bayesian statistical modeling and/or the BUGS programming language, while some are not suitable for ANOVA of experimental data. The motivation for the development of the **BANOVA** package is to overcome these limitations and to offer user-friendly routines for the applied researcher. The ease of use was illustrated in three applications : (1) models can be run by only a single function call, (2) only a small number of settings are required to run each model, (3) outputs are included in summary tables, such as the table of effect sizes, table of p values and tables of means, which are easy to interpret for the applied researcher.

The package utilizes the **JAGS** software, which offers many advantages from a modeling perspective, while in addition the **JAGS** code is produced as a by-product of the analysis and available for the applied researcher to inspect and modify. Another function (`BANOVA.run`) is available in the package that generates and runs **Stan** code. However, a disadvantage of this choice is that this decreases the computation efficiency. Future developments could focus on reprogramming the MCMC code in languages such as C++ or Java. In addition, the package can be extended with other models, for example using different distributions of the dependent variable.

Acknowledgments

References

Bartlett MS (1947). "The Use of Transformations." *Biometrics*, **3**, 39–52. doi:10.2307/3001536.

Bernardo JM, Giron FJ (1992). "Robust Sequential Prediction from Nonrandom Samples: The Election Night Forecasting Case." In JM Bernardo, JO Berger, AP Dawid, AFM Smith (eds.), *Bayesian Statistics 4*. Oxford University Press, Oxford.

Bernardo J SA (2000). *Bayesian Theory*. John Wiley & Sons.

Box GEP, Cox DR (1964). "An Analysis of Transformation." *Journal of the Royal Statistical Society B*, **26**(2), 211–252.

Breslow NE, Clayton DG (1993). "Approximate Inference in Generalized Linear Mixed Models." *Journal of the American Statistical Association*, **88**(421), 9–25. doi:10.1080/01621459.1993.10594284.

Cardinal RN, Aitken MRF (2005). *ANOVA for the Behavioral Sciences Researcher*. Psychology Press. doi:10.4324/9780203763933.

Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A (2017). "Stan: A Probabilistic Programming Language." *Journal of Statistical Software*, **76**(1), 1–32. doi:10.18637/jss.v076.i01.

Cochran WG (1940). "The Analysis of Variances When Experimental Errors Follow the Poisson or Binomial Laws." *The Annals of Mathematical Statistics*, **11**, 335–347. doi:10.1214/aoms/1177731871.

Conover WJ, Iman RL (1976). "On Some Alternative Procedures Using Ranks for the Analysis of Experimental Designs." *Communications in Statistics*, **A5**, 1349–1368. doi:10.1080/03610927608827447.

Denwood MJ (2016). "**runjags**: An R Package Providing Interface Utilities, Parallel Computing Methods and Additional Distributions for MCMC Models in **JAGS**." *Journal of Statistical Software*, **71**(9), 1–25. doi:10.18637/jss.v071.i09.

Draper NR, Hunter WG (1969). "Transformations. Some Examples Revisited." *Technometrics*, **11**, 23–40. doi:10.2307/1266762.

Draper NR, Smith H (1998). *Applied Regression Analysis.* 3 edition. John Wiley & Sons. doi:10.1002/9781118625590.

é HS (1957). *The Analysis of Variance.* John Wiley & Sons.

Etkin J, Ratner RK (2012). "The Dynamic Impact of Variety among Means on Motivation." *Journal of Consumer Research*, **38**(6), 1076–1092. doi:10.1086/661229.

Ferraro R, Kirmani A, Matherly T (2013). "Look at Me! Look at Me! Conspicuous Brand Usage, Self-Brand Connection, and Dilution." *Journal of Marketing Research*, **50**, 477–488. doi:10.1509/jmr.11.0342.

Fisher RA (1921). "On the "Probable Error" of a Coefficient of Correlation Deduced from a Small Sample." *Metron*, **1**, 3–32.

Fisher RA (1925). *Statistical Methods for Research Workers.* Oliver and Boyd, Edinburgh.

Fox J (1997). *Applied Regression Analysis, Linear Models and Related Methods.* Sage.

Gelman A (2005). "Analysis of Variance: Why It Is More Important Than Ever?" *The Annals of Statistics*, **1**(33), 43–47. doi:10.1214/009053604000001048.

Gelman A (2013). "*P*-Values and Statistical Practice." *Epidemiology*, **24**(1), 69–72. doi:10.1097/ede.0b013e31827886f7.

Gelman A, Carlin JB, Stern HS, Dunson DB (2013a). *Subjective and Objective Bayesian Statistics: Principles, Models, and Applications.* John Wiley & Sons.

Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB (2013b). *Bayesian Data Analysis.* 3rd edition. Cambridge University Press.

Gelman A, Hill J (2006). *Data Analysis Using Regression and Multilevel/Hierarchical Models.* Chapman & Hall/CRC.

Gelman A, Pardoe I (2006). "Bayesian Measures of Explained Variance and Pooling in Multilevel (Hierarchical) Models." *Technometrics*, **48**(2), 241–251. doi:10.1198/004017005000000517.

Geweke J (1992). "Evaluating the Accuracy of Sampling-Based Approaches to Calculating Posterior Moments." In JM Bernardo, JO Berger, AP Dawid, AFM Smith (eds.), *Bayesian Statistics 4*. Oxford University Press, Oxford.

Geweke J (1999). "Using Simulation Methods for Bayesian Econometric Models: Inference, Development, and Commnication." *Econometric Reviews*, **18**, 1–126. doi:10.1080/07474939908800428.

Gill J (2007). *Bayesian Methods for the Social and Behavioral Sciences.* 2nd edition. Chapman and Hall.

Hartley HO, Rao JN (1967). "Maximum Likelihood Estimation for the Mixed Analysis of Variance Model." *Biometrika*, **54**, 93–108. `doi:10.2307/2333854`.

Heidelberger P, Welch PD (1983). "Simulation Run Length Control in the Presence of an Initial Transient." *Operations Research*, **31**, 1109–1144. `doi:10.1287/opre.31.6.1109`.

Ito PK (1980). "Robustness of ANOVA and MANOVA Test Procedures." In PR Krishnainh (ed.), *Handbook of Statistics, Volume 1*, pp. 199–236. North-Holland Publishing Company, Amsterdam.

Kirk RE (1982). *Experimental Design: Procedures for the Behavioral Sciences.* 2nd edition. Brooks/Cole, Belmont.

Kruschke JK (2013). "Bayesian Estimation Supersedes the *t*-Test." *Journal of Experimental Psychology: General*, **142**(2), 573–603. `doi:10.1037/a0029146`.

Lakens D (2013). "Calculating and Reporting Effect Sizes to Facilitate Cumulative Science: a Practical Primer for *t*-Tests and ANOVAs." *Frontiers in Psychology*, **4**, 863. `doi:10.3389/fpsyg.2013.00863`.

Lindley DV, Smith AFM (1972). "Bayes Estimates for the Linear Model." *Journal of the Royal Statistical Society B*, **34**(1), 1–41.

Longford NT (1987). "A Fast Scoring Algorithm for Maximum Likelihood Stimation in Unbalanced Models with Nested Random Effects." *Biometrika*, **74**, 817–827. `doi:10.2307/2336476`.

Lord FM, Novick MR (1968). *Statistical Theories of Mental Test Scores.* Addison-Wesley, Reading.

Lunn D, Thomas A, Best NG, Spiegelhalter DJ (2000). "**WinBUGS** – A Bayesian Modelling Framework: Concepts, Structure, and Extensibility." *Statistics and Computing*, **10**(4), 325–337. `doi:10.1023/A:1008929526011`.

Martin AD, Quinn KM, Park JH (2011). "**MCMCpack**: Markov Chain Monte Carlo in R." *Journal of Statistical Software*, **42**(9), 22. `doi:10.18637/jss.v042.i09`.

McCullagh P, Nelder JA (1989). *Generalized Linear Models.* 2nd edition. Chapman and Hall, New York.

Nakagawa S, Schielzeth H (2013). "A General and Simple Method for Obtaining R^2 from Generalized Linear Mixed-Effects Models." *Methods in Ecology and Evolution*, **4**, 133–142. `doi:10.1111/j.2041-210x.2012.00261.x`.

Nunnaly JC (1967). *Psychometric Theory.* McGraw Hill, New York.

Nuzzo R (2014). "Scientific Method: Statistical Errors." *Nature*, **506**, 150–152. `doi:10.1038/506150a`.

Overall JE, Spiegel DK, Cohen J (1975). "Equivalence of Orthogonal and Nonorthogonal Analysis of Variance." *Psychological Bulletin*, **82**(2), 182–186. `doi:10.1037/h0076381`.

Plummer M (2003). "**JAGS**: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling." In K Hornik, F Leisch, A Zeileis (eds.), *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*. Technische Universität Wien, Vienna, Austria. URL `https://www.R-project.org/conferences/DSC-2003/Proceedings/Plummer.pdf`.

Plummer M (2016a). *JAGS: Just Another Gibbs Sampler*. Version 4.2.0, URL `http://mcmc-jags.sourceforge.net/`.

Plummer M (2016b). *rjags: Bayesian Graphical Models Using MCMC*. R package version 4-6, URL `https://CRAN.R-project.org/package=rjags`.

Plummer M, Best N, Cowles K, Vines K (2006). "**coda**: Convergence Diagnosis and Output Analysis for MCMC." *R News*, **6**(1), 7–11. URL `https://CRAN.R-project.org/doc/Rnews/`.

Press SJ (2003). *Subjective and Objective Bayesian Statistics: Principles, Models, and Applications*. John Wiley & Sons.

Raudenbusch SW (1988). "Eductional Applications of Hierarchical Linear Models: A Review." *Journal of Educational Statistics*, **13**, 85–116. `doi:10.2307/1164748`.

Raudenbush SW (1999). *Hierarchical Models*. John Wiley & Sons.

Rossi P (2017). *bayesm: Bayesian Inference for Marketing/Micro-Econometrics*. R package version 3.1-0.1, URL `https://CRAN.R-project.org/package=bayesm`.

Rouder JN, Speckman P, Sun D, Morey R, Iverson G (2009). "Bayesian t Tests for Accepting and Rejecting the Null Hypothesis." *Psychoneurosis Bulletin and Review*, **16**, 225–237. `doi:10.3758/pbr.16.2.225`.

Savage LJ (1954). *The Foundations of Statistics*. John Wiley & Sons.

Tabachnick BG, Fidell LS (1989). *Using Multivariate Statistics*. 2 edition. Harper and Row, New York.

Tan WY (1982). "Sampling Distributions and Robustness of t, F and Variance-Ratio in Two Samples and ANOVA Models with Respect to Departure from Normality." *Communications in Statistics – Theory and Methods*, **11**, 2485–2511.

Thorndike RL (1971). *Educational Measurement*. American Council on Education, Washington, DC.

Tiku ML (1971). "Power Function of the F-Test under Non-Normal Situations." *Journal of the American Statistical Association*, **66**, 913–915. `doi:10.2307/2284254`.

Wasserstein RL, Lazar NA (2016). "The ASA's Statement on p-Values: Context, Process, and Purpose." *The American Statistician*. `doi:10.1080/00031305.2016.1154108`.

Wedel M, Pieters R (2015). "The Buffer Effect: The Role of Color When Advertising Exposures Are Brief and Blurred." *Marketing Science*, **34**(1), 134–143. doi:10.1287/mksc.2014.0882.

Affiliation:

Chen Dong
Department of Mathematics
Mathematics Building
University of Maryland
College Park, MD 20742-4015, United States of America
E-mail: cdong@math.umd.edu

Michel Wedel
Department of Marketing
Robert H. Smith School of Business
3303 Van Munching Hall
University of Maryland
College Park, MD 20742-1815, United States of America
E-mail: mwedel@rhsmith.umd.edu
URL: http://www.rhsmith.umd.edu/directory/michel-wedel