partR2: Partitioning $R^2$ in generalized linear mixed models

Martin A. Stoffel$^{1,2}$, Shinichi Nakagawa$^3$, Holger Schielzeth$^1$

$^1$ Institute of Ecology and Evolution, Friedrich Schiller University Jena, Germany
$^2$ Institute of Evolutionary Biology, University of Edinburgh, Edinburgh, UK
$^3$ Evolution & Ecology Research Centre and School of Biological, Earth and Environmental Sciences, University of New South Wales, Sydney, Australia

ORCID: MAS 0000-0003-4030-3543, SN 0000-0002-7765-5182, HS 0000-0002-9124-2261

Corresponding authors
Martin A. Stoffel, martin.stoffel@ed.ac.uk
Holger Schielzeth, holger.schielzeth@uni-jena.de
Abstract

The coefficient of determination $R^2$ quantifies the amount of variance explained by regression coefficients in a linear model. It can be seen as the fixed-effects complement to the repeatability $R$ (intra-class correlation) for the variance explained by random effects and thus as a tool for variance decomposition. The $R^2$ of a model can be further partitioned into the variance explained by a particular predictor or a combination of predictors using semi-partial (part) $R^2$ and structure coefficients, but this is rarely done due to a lack of software implementing these statistics. Here, we introduce partR2, an R package that quantifies part $R^2$ for fixed effect predictors based on (generalized) linear mixed-effect model fits. The package iteratively removes predictors of interest from the model and monitors the change in the variance of the linear predictor. The difference to the full model gives a measure of the amount of variance explained uniquely by a particular predictor or a set of predictors. partR2 also estimates structure coefficients as the correlation between a predictor and fitted values, which provide an estimate of the total contribution of a fixed effect to the overall prediction, independent of other predictors. Structure coefficients can be converted to the total variance explained by a predictor, here called ‘inclusive’ $R^2$, as the square of the structure coefficients times total $R^2$. Furthermore, the package reports beta weights (standardized regression coefficients). Finally, partR2 implements parametric bootstrapping to quantify confidence intervals for each estimate. We illustrate the use of partR2 with real example datasets for Gaussian and binomial GLMMs and discuss interactions, which pose a specific challenge for partitioning the explained variance among predictors.
Introduction

Coefficients of determination $R^2$ are of interest in the study of ecology and evolution, because they quantify the amount of variation explained by a linear model (Edwards et al., 2008). By doing so, they go beyond significance testing in putting effects in perspective of the phenotypic variance. $R^2$ is expressed as a proportion of the total variance in the response, which represents a biologically relevant quantity if the total variation is representative for the total population (de Villemereuil et al., 2018). The total coefficient of determination in a generalised linear mixed model (GLMM) quantifies the variance explained by all fixed effects together (marginal $R^2$ sensu Nakagawa & Schielzeth, 2013, also known as the total correlation coefficient, Watanabe (1960)).

However, it is often of interest to attribute explained variation to individual predictors. Semi-partial coefficients of determination, also known as part $R^2$, decompose the variance of $R^2$ into components uniquely explained by individual predictors (Jaeger et al., 2017; Jaeger, Edwards & Gurka, 2019) or sets of predictors (Figure 1). The set of all predictors in the model yields the total proportion of variance explained by the fixed part of the model (total $R^2$). With correlations among predictors, it often happens that predictors in univariate regressions explain a large share of the variance, but do not show large part $R^2$ if other correlated predictors are included in the model. Note that part $R^2$ estimates the proportion of the variance in the response explained by a predictor while accounting for covariance between this predictor and the other predictors in the model, whereas the (arguably more familiar) partial $R^2$ estimates the proportion of the variance that is explained by a predictor of interest after accounting for the other predictors from the response as well as the predictor of interest. The difference is subtle, but important (see more below). Therefore, part $R^2$ represents ‘variance accounted for’ in relation to the total variance, but partial $R^2$ does not. Consequently, part $R^2$ will be conceptually easier to compare with (total) $R^2$.

Structure coefficients provide a valuable addition to part $R^2$ in the decomposition of the phenotypic variance (Nimon et al., 2008; Yeatts et al., 2017). Structure coefficients quantify the correlation between individual predictors and the linear predictor. Predictors that correlate well with a response, but are fitted with collinear predictors may show large structure coefficients as they are correlated to the predicted response, but low part $R^2$ as other predictors explain part of the same variance. Structure coefficients range from -1 to 1 with their absolute value expressing the correlation relative to a perfect correlation if a single predictor explains as much as the total fixed part of the model.
Structure coefficients are correlations and since the square of a correlation yields the variance explained, we can use structure coefficients to estimate the total variance explained by a predictor (Nimon et al., 2008). We call this the inclusive $R^2$ of a predictor and calculate it as the squared structure coefficient, i.e. its contribution to the linear predictor independent of other predictors (Nimon et al., 2008) times the proportion of variance explained by the linear predictor (which is the ‘total’ marginal $R^2$ of the model) (see also Nathans, Oswald & Nimon, 2012). As far as we are aware, inclusive $R^2$ has not been implemented before, but it provides valuable insights into the structure of the variance explained (Figure 1).

Here, we introduce partR2, a versatile package for estimating part $R^2$, inclusive $R^2$, structure coefficients and beta weights from mixed-effects models. Figure 1 gives an overview of how variances are calculated and how they relate to partial $R^2$ and to commonality analysis (Ray-Mukherjee et al., 2014; Seibold & McPhee, 1979; Zientek & Thompson, 2006). We illustrate how to use partR2 with real example datasets for Gaussian and binomial GLMMs, discuss how to estimate part $R^2$ in the presence of interactions and discuss some challenges and limitations.

---

**Figure 1: Conceptual framework for the estimation of proportions of variance components in a mixed model.** The large grey circle symbolizes the variance in a response $Y$, the dark grey area on the top indicates the share explained by random effects and the coloured ellipses symbolize variance in covariates with.
Mathematical representation

Part $R^2$

A Gaussian mixed-effects model can be written as:

$$y = X\beta + \sum \alpha_k + \varepsilon$$

(Eq. 1)

$$\alpha_k \sim N(0, \sigma^2_{\alpha_k})$$

$$\varepsilon \sim N(0, \sigma^2_{\varepsilon})$$

Where $y$ is a vector of response values (outcomes), $X$ is the design matrix of fixed effects, $\beta$ is a vector of regression coefficients, $\sum \alpha_k$ is the random part of the model that might contain multiple random effects and $\varepsilon$ is a vector of residual deviations. The linear predictor $\eta$ represents the vector of predicted values from the fixed part of the model as $\eta = X\beta$. Note that we dealing with estimates of regression coefficients and variance components throughout (hence all $\beta$ should be read as $\hat{\beta}$).

Since we are interested in the proportion of the phenotypic variance explained, we symbolize variance components by upper case $Y$ and index by the source of variance (als in Figure 1). While variances are frequently represented as $V$ with the source of variance as an index, this leads to ambiguity for $V_x$ which might represent variance in $y$ explained by $x$ or variance in $x$ itself, which is why we use this alternative notation. The total variance in the response is $Y_{total} = \text{var}(y)$ and is estimated from the raw data or from the model (see below). The variance of the residuals is estimated by the model as $Y_R = \text{var}(\varepsilon)$. The variance of the (sum of) random effects is estimated by the model as $Y_{RE} = \text{var}(\sum \alpha_k)$ and the variance explained by fixed effects can be estimated as the variance in the linear predictor $Y_X = \text{var}(X\beta)$.

The coefficient of determination $R^2$ estimates the proportion of variance in the response that is explained by fixed effects. The coefficient of determination $R^2$ for the total fixed part of the model is thus:

$$R^2 = \frac{Y_X}{Y_X + Y_{RE} + Y_R} = \frac{Y_X}{Y_{total}}$$

(Eq. 2)
Note that the sum of the components in the denominator might deviate numerically from the total outcome variance in the raw data. However, conceptually they are the same in that they represent the population-level outcome variance. The variance in the outcome is an estimate from the specific sample, while the sum of components of the mixed model represents a population-level estimate given the data and the model.

A reduced model with a (set of) fixed effect predictors $X^*$ removed but the same random effect structure can be fitted as (now using the tilde to highlight the differences from Eq. 1):

$$y = \tilde{X}\tilde{\beta} + \sum \tilde{\alpha}_k + \tilde{\epsilon}$$  \hspace{1cm} \text{(Eq. 3)}

$$\tilde{\alpha}_k \sim N(0, \sigma_{\tilde{\alpha}_k}^2)$$

$$\tilde{\epsilon} \sim N(0, \sigma_{\tilde{\epsilon}}^2)$$

with the variance in the linear predictor of the reduced model being $Y_\tilde{X} = \text{var}(\tilde{X}\tilde{\beta})$.

The variance uniquely explained by $X^*$ is then the difference between the variance explained by fixed effects in the full and the reduced model $Y_{X^*} = Y_X - Y_\tilde{X}$. Part $R^2$ sets this variance in proportion to the total outcome variance:

$$R_{X^*}^2 = \frac{Y_X - Y_\tilde{X}}{Y_X + Y_{RE} + Y_R} = \frac{Y_X - Y_\tilde{X}}{Y_{Total}}$$  \hspace{1cm} \text{(Eq. 4)}

The process of fitting a reduced model, estimation of $Y_{X^*}$ and estimation of $R_{X^*}^2$ can be repeated for all predictors and combinations of predictors. At the limit for a model with all fixed effects removed, $R_{X^*}^2 = R_X^2$.

**Side-note on partial $R^2$**

For completeness we note that the partial $R^2$ could be calculated as:

$$R_{X^*}^2 = \frac{Y_X - Y_\tilde{X}}{Y_{Total} - Y_\tilde{X}}$$  \hspace{1cm} \text{(Eq. 5)}

However, this estimate does not put the explained variance in perspective of the total variance in the response. It has the major disadvantage that the denominator depends on $Y_\tilde{X}$. The same effect in terms of $Y_{X^*}$ thus appears larger if the reduced model explains more variance (larger $Y_\tilde{X}$). Even in the case of independent additive predictors, the contributions of the different fixed effects do not sum up to $R_X^2$, because of the change in the denominator that different $Y_{X^*}$ are compared to.
Finally, since we are interested in explaining phenotypic variation in some biological response (the phenomenon to be explained), we think that part $R^2$ is the more relevant quantity, as it represents the proportion of variance in the response uniquely explained by $X^\star$.

**Inclusive $R^2$**

Structure coefficients are the Pearson correlations between a particular predictor of interest $x^\star$ and the linear predictor $\eta$. Note that we now use a lower case $x^\star$ to indicate that we are dealing with a single predictor. Structure coefficients are quantified from the full model as:

$$SC_{x^\star} = \text{cor}(\eta, x^\star)$$  \hspace{1cm} (Eq. 6)

The squared correlation between two variables $a$ and $b$ gives the variance explained for these variables $\text{cor}(a, b)^2 = R^2_{ab}$. The squared structure correlations thus quantify the proportion of variance in the linear predictor $Y^\star_\chi$ that is explained by a the predictor of interest $x^\star$. Since the proportion of outcome variance explained by the linear predictor in the full model is $R^2_\chi$, the inclusive variance explained by predictor $x^\star$ is:

$$IR^2_{x^\star} = SC^2 \cdot R^2_{x^\star}$$  \hspace{1cm} (Eq. 7)

Inclusive $R^2$ as we define it here, complements part $R^2$ by giving additional insights. While part $R^2$ quantifies the variance uniquely explained by a predictor (or set of predictors), inclusive $R^2$ quantifies the total proportion of variance explained in the model, both uniquely and jointly with other predictors. In the special case of a single predictor in a model $SC_{x^\star} = \text{cor}(\eta, x^\star) = 1$, such that $IR^2_{x^\star} = R^2_{x^\star}$.

**Part $R^2$ in non-Gaussian models**

For Gaussian models there is a single residual error term $\varepsilon$ with variance $Y = \text{var}(\varepsilon)$. For non-Gaussian models, however, there is additional error that arises from the link function that translates latent-level predictions to observed outcomes. This variance can be approximated for a variety of link functions and error distributions (Nakagawa & Schielzeth, 2010; Nakagawa, Johnson & Schielzeth, 2017). Our R package currently implements distribution-specific variances for Poisson models with log and square root link functions and binomial models with logit and probit link functions. For Poisson models and non-binary binomial models (proportion models), partR2 also
fits an observational level random effect (if none is fitted already) to estimate variance due to
overdispersion (Harrison, 2014). Both the overdispersion variance, now denoted $Y_R$ and the
distribution-specific variance $Y_D$, are included in the denominator of the part $R^2$ calculation:

$$R^2_{X*} = \frac{Y_X - Y_X}{Y_X + Y_{RE} + Y_D + Y_R} \quad \text{(Eq. 8).}$$

Notably, there are other estimation methods for $R^2$ for non-Gaussian models or GLMM (Jaeger et
al., 2017; Piepho, 2019). Currently, partR2 only implements the method based on Nakagawa &
Schielzeth (2013) and Nakagawa, Johnson & Schielzeth (2017).

**Other implementations in R packages**

There are a few R packages that calculate part $R^2$ for linear models (lm), for example
rockchalk::getDeltaRsquare (Johnson & Grothendieck, 2019). Other packages calculate
partial $R^2$ (not part $R^2$) such as asbio::partial.R2 (Aho, 2020) and rr2::R2 (Ives & Li, 2018) for
lm and rsq::rsq.partial (Zhang, 2020) for linear models and generalized linear models (glm).

Note that partial $R^2$ is different from part (semi-partial) $R^2$ (partial $R^2 >$ part $R^2$), since it represents the
unique variance explained by a particular predictor but after removing (‘partialling out’) the variance
explained by the other predictors (Yeatts et al., 2017, Figure 1). The ppcor package calculates semi-
partial and partial correlations, but does not work on fitted GLM or GLMM models (Kim, 2015). The
package yhat features functions for commonality analyses in glms (Nimon, Oswald & Roberts,
2020). None of these packages estimates part $R^2$ for mixed-effects models that we focus on here.

Several packages estimate (marginal) $R^2$ as the variance explained by all fixed effects in linear mixed-
effects models. This includes performance::r2_nakagawa (Lüdecke et al., 2020),
MuMIn::r.squaredGLMM (Bartoń, 2019), and rptR::rpt (Stoffel, Nakagawa & Schielzeth, 2017).

These packages do not allow to estimate part $R^2$. The only versatile package to estimate part $R^2$ from
linear mixed-models is r2glmm (Jaeger, 2017). The function r2glmm::r2beta computes part $R^2$
from lmer, lme and glmmPQL model fits (also for linear models lm and glm) based on Wald statistics.
However, it does neither support lme4::glmer for generalized linear model fits nor does it allow
to estimate $R^2$ for combinations of predictors. Furthermore, it does not estimate structure
coefficients, inclusive $R^2$ or part $R^2$ for multilevel factors as a unit.
Features of \texttt{partR2}

\texttt{partR2} takes a fitted (generalized) linear mixed-model (GLMM), from the popular mixed model package \texttt{lme4} (Bates et al., 2015) and estimates part $R^2$ by iteratively removing fixed effects (Nimon et al., 2008). The specific fixed effects of interest are specified by the \texttt{partvars} and/or by the \texttt{partbatch} argument. The package estimates part $R^2$ for all predictors specified in \texttt{partvars} individually and in all possible combinations (the maximum level of combinations can be set by the \texttt{max_level} argument). A custom specification of fixed effects of interest saves computation time as compared to an all-subset specification and is therefore required in \texttt{partR2}.

The central function \texttt{partR2} will work for Gaussian, Poisson and binomial GLMMs. Since the model fit is done externally, there is no need to supply a family argument. For non-Gaussian GLMMs, the package estimates link-scale $R^2$ (sensu Nakagawa & Schielzeth, 2013). We implement parametric bootstrapping to quantify sampling variance and thus uncertainty in the estimates. Parametric bootstrapping works through repeated model fitting on simulated data based on fitted values (Faraway, 2015). The number of bootstrap iterations is controlled by the \texttt{nboot} argument. We recommend a low number of \texttt{nboot} for testing purposes and a large number (e.g. \texttt{nboot = 1000}) for the final analysis.

The package returns an object of class \texttt{partR2} that contains elements for part $R^2$, inclusive $R^2$, structure coefficients, beta weights (standardized regression slopes), bootstrapping iterations and some other information. An extended summary, that includes inclusive $R^2$, structure coefficients and beta weights can be viewed using the \texttt{summary} function. The \texttt{forestplot} function shows a graphical representation of the variance explained by individual predictors and sets of predictors along with their bootstrapping uncertainties. All computations can be parallelized across many cores based on the \texttt{future} and \texttt{furrr} packages (Vaughan & Dancho, 2018; Bengtsson, 2020). An extended vignette with details on the complete functionality accompanies the package.

Example with Gaussian data

We use an example dataset with hormone data collected from a population of captive guinea pigs to illustrate the features of \texttt{partR2}. The dataset contains testosterone measurements of 31 male guinea pigs, each measured at 5 time points (age between 120 and 240 days at 30-day intervals). We analyze log-transformed testosterone titers and fit male identity as a random effect. As covariates the dataset contains the time point of measurement and a rank index derived from behavioral observations around the time of measurement (Mutwill et al., in prep.).
Rank and Time are correlated in the dataset ($r = 0.40$), since young individuals are typically low rank, while older individuals tend to hold a high rank. Time might be fitted as a continuous predictor or as a factor with five levels. Here we present the version of a factorial predictor to illustrate the estimation of part $R^2$ for interactions terms. Hence, an interaction between time and rank will also be fitted.

First, the package needs to be loaded (after successful installation) in an R session (R Core Team, 2019). The package comes with the guinea pig dataset that also needs to be loaded using the data function.

```r
library(partR2)
data(GuineaPigs)
```

A single record contains missing values for testosterone measurements. Missing records can be problematic to handle in partR2 and are better removed prior to the analysis. We also log-transform the response and convert Time to a factor and filter for the first three time points to simplify the output.

```r
GuineaPigs <- subset(GuineaPigs, !is.na(Testo) & !is.na(Rank) & (Time %in% c(1, 3, 5)))
GuineaPigs$TestoTrans <- log(GuineaPigs$Testo)
GuineaPigs$Time <- factor(GuineaPigs$Time)
```

We then fit a linear mixed effects model using `lmer` from the lme4 package (Bates et al., 2015). Further exploration of the data and model checks are omitted here for simplicity, but are advisable in real data analysis.

```r
library(lme4)
mod <- lmer(TestoTrans ~ Rank * Time + (1|MaleID), data=GuineaPigs)
```

The `partR2` analysis takes the `lmer` model fit (an `merMod` object) and a character vector `partvars` indicating the fixed effects to be evaluated. Interactions are specified with the colon syntax (see the package’s vignette for further details).
The function returns a `partR2` object. The `print` function reports the part coefficients of determination and a more extensive summary can be viewed with the `summary` function which also shows inclusive $R^2$, structure coefficients and beta weights (standardized slopes) (Figure 2).

```r
res <- partR2(mod, partvars = c("Rank", "Time", "Rank:Time"), nboot=100)

print(res)

summary(res, round_to = 2)
```

### R2 (marginal) and 95% CI for the full model:

<table>
<thead>
<tr>
<th></th>
<th>R2</th>
<th>CI_lower</th>
<th>CI_upper</th>
<th>ndf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>0.17</td>
<td>0.09</td>
<td>0.36</td>
<td>6</td>
</tr>
</tbody>
</table>

### Part (semi-partial) R2:

<table>
<thead>
<tr>
<th>Predictor(s)</th>
<th>R2</th>
<th>CI_lower</th>
<th>CI_upper</th>
<th>ndf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>0.17</td>
<td>0.09</td>
<td>0.36</td>
<td>6</td>
</tr>
<tr>
<td>Rank</td>
<td>0.00</td>
<td>0.00</td>
<td>0.18</td>
<td>6</td>
</tr>
<tr>
<td>Time</td>
<td>0.02</td>
<td>0.00</td>
<td>0.20</td>
<td>4</td>
</tr>
<tr>
<td>Rank:Time</td>
<td>0.04</td>
<td>0.00</td>
<td>0.21</td>
<td>4</td>
</tr>
<tr>
<td>Rank+Time</td>
<td>0.02</td>
<td>0.00</td>
<td>0.20</td>
<td>4</td>
</tr>
<tr>
<td>Rank+Rank:Time</td>
<td>0.16</td>
<td>0.08</td>
<td>0.34</td>
<td>3</td>
</tr>
<tr>
<td>Time+Rank:Time</td>
<td>0.04</td>
<td>0.00</td>
<td>0.22</td>
<td>2</td>
</tr>
<tr>
<td>Rank+Time+Rank:Time</td>
<td>0.17</td>
<td>0.09</td>
<td>0.36</td>
<td>1</td>
</tr>
</tbody>
</table>

### Inclusive R2 (SC^2 * R2):

<table>
<thead>
<tr>
<th>Predictor</th>
<th>IR2</th>
<th>CI_lower</th>
<th>CI_upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rank</td>
<td>0.13</td>
<td>0.03</td>
<td>0.26</td>
</tr>
<tr>
<td>Time3</td>
<td>0.00</td>
<td>0.00</td>
<td>0.04</td>
</tr>
<tr>
<td>Time5</td>
<td>0.00</td>
<td>0.00</td>
<td>0.04</td>
</tr>
<tr>
<td>Rank:Time3</td>
<td>0.05</td>
<td>0.01</td>
<td>0.13</td>
</tr>
<tr>
<td>Rank:Time5</td>
<td>0.01</td>
<td>0.00</td>
<td>0.07</td>
</tr>
</tbody>
</table>

### Structure coefficients $r(Yhat,x)$:

<table>
<thead>
<tr>
<th>Predictor</th>
<th>SC</th>
<th>CI_lower</th>
<th>CI_upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rank</td>
<td>0.87</td>
<td>-0.56</td>
<td>0.94</td>
</tr>
<tr>
<td>Time3</td>
<td>0.14</td>
<td>-0.18</td>
<td>0.43</td>
</tr>
<tr>
<td>Time5</td>
<td>0.16</td>
<td>-0.26</td>
<td>0.48</td>
</tr>
<tr>
<td>Rank:Time3</td>
<td>0.56</td>
<td>0.22</td>
<td>0.75</td>
</tr>
<tr>
<td>Rank:Time5</td>
<td>0.20</td>
<td>-0.14</td>
<td>0.57</td>
</tr>
</tbody>
</table>

### Beta weights (standardised estimates)

<table>
<thead>
<tr>
<th>Predictor</th>
<th>BW</th>
<th>CI_lower</th>
<th>CI_upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rank</td>
<td>0.50</td>
<td>-0.08</td>
<td>0.94</td>
</tr>
<tr>
<td>Time3</td>
<td>-0.19</td>
<td>-0.53</td>
<td>0.14</td>
</tr>
<tr>
<td>Time5</td>
<td>0.17</td>
<td>-0.20</td>
<td>0.55</td>
</tr>
<tr>
<td>Rank:Time3</td>
<td>0.17</td>
<td>-0.36</td>
<td>0.83</td>
</tr>
<tr>
<td>Rank:Time5</td>
<td>-0.36</td>
<td>-0.95</td>
<td>0.38</td>
</tr>
</tbody>
</table>

Figure 2: Summary output for example data analysis with Gaussian data (guinea pig analysis).
The variances appear largely additive, since combinations of predictors explain about the sum of the variance explained by individual predictors. The main components of the `partR2` object can be accessed for further processing as `res$R2` for part $R^2$ (with point estimates and confidence intervals), `res$SC` for structure coefficients, `res$IR2` for inclusive $R^2$ and `res$BW` for beta weights.

### Dealing with interactions

Models with interactions are problematic, because the variance explained by a main factor can be estimated in multiple ways (Figure 3) and because of the internal parametrization of the model matrix.

![Conceptual framework for dealing with interactions](image)

**Figure 3:** Conceptual framework for dealing with interactions. An interaction is the product of two main effects and thus often correlated with each of the main effects. The figure shows three options for estimating the part $R^2$ for main effects that are involved in an interaction.
The model output above shows the number of parameters fitted in each model (Figure 2, each row in the R2 part refers to a reduced model). In the print and summary output this is visible as a column labelled 'ndf'. A close inspection shows that the removal of rank did not change the number of parameters (6 for the full model, 6 for the model excluding rank). This is because the model matrix is reparametrized in the reduced model and lmer will fit three terms for the interaction (here Time1:Rank, Time3:Rank, Time5:Rank) rather than just two for the interaction in the full model. Dummy coding of the factor can be usefully combined with centering of dummy coded variables (Schielzeth, 2010) and gives more control over this re-parametrisation. It allows for example to estimate the part R2 for the average effect of Rank by constraining the average Rank effect to zero, so that only the two contrasts are fitted (here Time3:Rank, Time5:Rank):

```r
GuineaPigs <- cbind(GuineaPigs, model.matrix(~ 0 + Time,
                                      data=GuineaPigs))
GuineaPigs$Time3 <- GuineaPigs$Time3 - mean(GuineaPigs$Time3)
GuineaPigs$Time5 <- GuineaPigs$Time5 - mean(GuineaPigs$Time5)
```

The model can then be fitted with dummy predictors. Since the usual specification in partR2 via partvars would fit all possible combinations, including combinations of the different Time terms, such a run can take a long time. However we are mostly interested in fitting and removing all dummy predictors at a time. The package therefore features an additional argument partbatch to specify a list of character vectors containing the sets of predictors that should always be kept together. In the example, the list has two elements, a character vector for the dummy-coded main effects and a character vector for the interaction terms. The analysis yields part R2 for two batches of predictors as well as Rank and their combinations.

```r
mod <- lmer(TestoTrans ~ (Time3 + Time5) * Rank + (1|MaleID),
             data=GuineaPigs)
batch <- c("Time3", "Time5")
partR2(mod, partvars=c("Rank"), partbatch=list(Time=batch,
                                        `Time:Rank` = paste0(batch, ":Rank")), nboot=100)
```
This, however, is only one way of dealing with interactions (Option A in Figure 3). It represents the variance uniquely explained by main effects even in the presence of an interaction. Since interactions are the products of main effects, interaction terms are typically correlated with main effects and the part $R^2$ calculated above might not represent a biologically relevant quantity. There are two alternative ways of how to deal with interactions. Both are possible in partR2, but since requirements differ between applications, we do not implement one with priority.

One way to think about variance explained by main effects and their interactions is to pool the variance explained by a main effect with the variance explained by interactions that the term is involved in (Option B in Figure 3). In the guinea pig example, for instance, Rank might be considered important either as a main effect or in interaction with time and we might want to estimate the total effect of rank. This can be done for the guinea pig dataset by using partbatch:

```r
mod <- lmer(Testo ~ Time * Rank + (1|MaleID), data=GuineaPigs) partR2(mod, partbatch = list(Time=c("Time", "Time:Rank"), Rank=c("Rank", "Time:Rank")), nboot=100)
```

A third, which we think usually preferable option is to prioritize main effects by assigning the proportion of variance that is explained by a main effect together with the variance jointly explained with its interaction to the main effect (Option C in Figure 3). This implies that part $R^2$ for a main effect is estimated when its own interaction is excluded from the model (mod1 and part1 below). The variance explained by the interaction is then estimated in a separate model (mod2 and part2 below).

We have implemented a helper function mergeR2 that allows to merge two partR2 runs.

```r
mod1 <- lmer(Testo ~ Time * Rank + (1|MaleID), data=GuineaPigs) part1 <- partR2(mod1, partvars = c("Time:Rank"), nboot=100) mod2 <- lmer(Testo ~ Time + Rank + (1|MaleID), data=GuineaPigs) part2 <- partR2(mod2, partvars = c("Time", "Rank"), nboot=100) mergeR2(part1, part2)
```

All these results can be viewed by print, summary and plotted by forestplot. It is important to bear in mind the differences in the interpretation as illustrated in Figure 3.
An example with proportion data

As an example for proportion data, we analyze a dataset on spatial variation in color morph ratios in a color-polymorphic species of grasshopper. Individuals of this species occur either in a green or a brown color variant and the dataset contains counts of brown and green individuals (separated for females and males) from 42 sites sampled in the field (Dieker et al., 2018). Site identity will be fitted as a random effect. As covariates the dataset contains a range of Bioclim variable that describe various aspects of ecologically relevant climatic conditions (Karger et al., 2017). The aim is to identify the climatic conditions that favour one or the other colour variant.

We first load the grasshopper dataset. We standardize all Bioclim variables using the `scale` function and add an observation-level counter that will be used as an observation-level random effect (OLRE) to account for overdispersion (Harrison, 2014).

```r
data(Grasshoppers)
for (i in which(substr(colnames(Grasshoppers),1,3)=="Bio")){
  Grasshoppers[,i] <- scale(Grasshoppers[,i])
}
Grasshoppers$OLRE <- 1:nrow(Grasshoppers)
```

We first fit a GLMM with binomial error structure and logit link using the `glmer` function from the `lme4` package (Bates et al., 2015). A previous analysis has shown that the first principle component of the Bioclim data explains a small, but significant part of variation in morph ratios (Dieker et al., 2018). For illustration, we use the four Bioclim variables that show a loading of more than 0.30 on the first principle component.

```r
mod <- glmer(cbind(nGreen, nBrown) ~ Bio7 + Bio14 + Bio17 + Bio19 + (1|SiteID) + (1|OLRE), data=Grasshoppers, family="binomial")
res <- partR2(mod, partvars=c("Bio7", "Bio14", "Bio17", "Bio19"),
              max_level = 1, nboot=100)
```

The summary output informs us (at the bottom) that there have been warnings in the bootstrapping processes. This is not unusual since bootstrapping frequently generates data, for which one of the
parameters is estimated at the boundary (in particular if one of the variance components is very small). The results can be visualised using the `forestplot` function (Figure 4). Plotting is based on `ggplot2` (Wickham, 2016), and multiple forest plots can easily be assembled using the `patchwork` package (Pedersen, 2020). Forest plots show the effect sizes graphically and can be set to either show part $R^2$ when `type = "R2"` (the default), inclusive $R^2$ when `type = "IR2"`, structure coefficients when `type = "SC"`, and beta weights (standardized model estimates) with `type = "BW"`.

```r
p1 <- forestplot(res, type = "R2")
p2 <- forestplot(res, type = "IR2")
p3 <- forestplot(res, type = "SC")
p4 <- forestplot(res, type = "BW")

library(patchwork)
(p1 + p2) / (p3 + p4) + plot_annotation(tag_levels = "A",
  tag_prefix = "("), tag_suffix = ")")
```

**Figure 4:** Comparison of part $R^2$ for individual predictors (A), inclusive $R^2$ (B), structure coefficients (C) and beta weights (D) for an example dataset with proportion data from grasshoppers.
A comparison of part $R^2$, inclusive $R^2$, structure coefficients beta weights shows the different insights that can be gained from these different summaries of the model fit (Figure 3). In this case, three of the Bioclim variables (Bio14, Bio17, Bio19) are highly positively correlated ($r \geq 0.93$), while a fourth one (Bio7) is moderately negatively correlated to all three of them ($r \leq -0.63$). Part $R^2$ are thus low, because none of the parameters uniquely explains a large share of the variance. Bio17 seems to be the best predictor of morph ratios, with the largest (negative) beta weight, largest part $R^2$, largest structure coefficients and largest inclusive $R^2$. Beta weights for the two positively correlated (but slightly weaker) predictors, Bio14 and Bio19, switch sign as is not unusual for collinear predictors. This means that after accounting for the effect of Bio17, they contribute positively to prediction. However, structure coefficients show that both variables load negatively on the linear predictor, as does Bio17.

**Challenges**

Using transformation or functions in the formula argument can lead to issues with matching the terms of the model with the `partvars` argument of `partR2`. It is therefore important that the names in `partvars` match exactly the terms in the `merMod` object. However, any complications are easily circumvented by implementing the transformations before fitting the model and storing them in the data frame used in the analysis. It is also worth to be aware that unusual names may cause complications and renaming can offer an easy solution.

We have repeatedly seen model outputs where the point estimate does not fall within the confidence interval. This might seem like an issue in the package, but in our experience usually indicates issues with the data and/or the model. In fact, parametric bootstrapping can be seen as a limited form of posterior predictive model checks (Gelman & Hill, 2006). If generating new data from the fitted model (as done with parametric bootstrapping) results in data that are dissimilar to the original data, then the model is probably not a good fit to the data. Bootstrap iterations can sometimes yield slightly negative estimates of part $R^2$, in particular if the variance explained by a predictor is low. These negative estimates happen in mixed-effects models, because estimates of random-effect variance might change when a predictor is removed and this can lead to a slight decrease in the residual variance, and hence a proportional increase in $R^2$ (see also Rights & Sterba, 2019). By default, `partR2` sets negative $R^2$ values to 0, but this can be changed by setting `allow_neg_r2` to `TRUE`. It also happens that inclusive $R^2$ is estimated slightly lower than part $R^2$ when the contribution of a particular predictor is very large. We consider both cases as
sampling error that should serve as a reminder that variance components are estimated with relatively large uncertainty and minor differences should not be over-interpreted. A warning needs to be added for the estimation of $R^2$ (and, in fact, also repeatability $R$) from small datasets. In particular if the number of levels of random effect is low, variance components might be slightly overestimated (Xu, 2003). This issue applies similarly to the variance explained by fixed effects, in particular if the number of predictors is large relative to the number of data points.

### Code and data availability

The current stable version of `partR2` can be downloaded from CRAN (https://cran.r-project.org/web/packages/partR2/index.html) and the development version can be obtained from GitHub (https://github.com/mastoffel/partR2). The data used in the examples is part of the package.

### Acknowledgements

We thank Alexandra Mutwill for providing the guinea pig data.

### References


Piepho H-P. 2019. A coefficient of determination (R2) for generalized linear mixed models. 


