Oral Qualifying Exam

Megan Jones

Overview

This document is the written component of the oral qualifying examination for Megan Jones. The document is organized as follows. Section 1 provides an introduction to the use of effect size measures in the context of neuroimaging. Section 2 provides a modified version of Megan's first dissertation paper (Jones *et al.* 2023a), concerning an R package (**RESI**) for computation of effect sizes with confidence intervals for a broad range of models. Extra material from the paper can be found in the Appendix. Section 3 provides a literature review for effect size use in individual prediction in neuroimaging and underlying theory for semiparametric estimators that may be useful in this context. Section 4 describes the potential applications for the semiparametric estimators in current and future projects.

1. Introduction

1.1. Effect sizes vs. p values

Standardized effect sizes are unitless indices used to describe the magnitude of an association. While unstandardized effect sizes can be informative in a given scientific context, standardized measures have the benefit of allowing communication of associations for outcomes measured without an interpretable scale and facilitating comparison across settings where outcomes are measured using different instruments. Unlike p values, which are often used to evaluate statistical significance, effect sizes do not depend on sample size (Betensky 2019). A well known criticism of p values and significance testing is that for large sample sizes, very small effects will be found as significant, even though these effects may be negligible in real-world application, as noted in Principle 5 of the ASA Statement on Statistical Significance and p Values (Wasserstein and Lazar 2016). In contrast, effect sizes communicate the strength of the effect rather than the existence of an effect of arbitrary size, which may be more meaningful in practice (Sullivan and Feinn 2012). Although increased sample size helps improve the precision of the estimate of an effect size, the effect size is a parameter that is not dependent on sample size (Kang *et al.* 2023). Standardized effect sizes are also important statistical parameters for power analysis, as power is a function of the effect size, sample size, and degrees of freedom of the statistical test. Journals and statistical guidelines are increasingly encouraging authors to report effect sizes, either unstandardized or standardized, and their CIs alongside or in place of p values (Wasserstein and Lazar 2016; Wilkinson 1999; American Psychological Association 1994, 2001, 2010, 2020; Althouse *et al.* 2021). However, they are still not commonly reported (Fritz *et al.* 2012; Amaral and Line 2021) and when

reported, they often do not include confidence intervals (Fritz *et al.* 2012). This may be due to challenges that will be discussed below.

1.2. Neuroimaging context

As with many other fields, effect sizes are being increasingly emphasized in neuroimaging (Bowring *et al.* 2019; Nichols *et al.* 2017; Chen *et al.* 2017; Reddan *et al.* 2017; Soares *et al.* 2016). While neuroimaging, and specifically magnetic resonance imaging (MRI), encompasses many modalities such as structural MRI, diffusion MRI, and functional MRI (fMRI), we will focus our discussion on fMRI, as one of the most common neuroimaging modalities (Xue *et al.* 2010). Typically, fMRI data of the brain is measured as the blood-oxygen-level dependent (BOLD) signal, which changes in accordance with the underlying hemodynamic response function (HRF) (Xue *et al.* 2010). The data is collected as a 3-dimensional time series and typically undergoes a series of preprocessing steps to make images comparable across individuals. At each time point, or volume, the brain image is comprised of 3-dimensional voxels, with the usual image containing tens of thousands to over a million voxels, depending on the image resolution (Jollans *et al.* 2019). Subject-level (or first-level) models can be fit to the time series data to generate 3-dimensional maps or 2-dimensional summaries (such as in functional connectivity analysis) (Soares *et al.* 2016).

Group-level inference for fMRI data can proceed in a univariate or multivariate framework to attempt to make conclusions about differences between individuals (Soares *et al.* 2016). In the neuroimaging context, "univariate" approaches refer to those that analyze each voxel individually initially and then utilize various methods to perform valid inference in the presence of many tests. These analyses are typically group-level or associational (Soares *et al.* 2016). "Multivariate" approaches instead consider the whole brain image (or summary of the imaging data) as predictor data to generate predictions for individual features, such as age or diagnosis (Soares *et al.* 2016). As univariate approaches have historically been more common (Soares *et al.* 2016), we will first focus on these and how effect sizes can be utilized in this context, before discussing extensions to increasingly popular multivariate analyses.

One popular route of analysis for fMRI data analysis is to generate mass univariate statistical maps for a given sample. This involves fitting a model (e.g., a generalized linear model (GLM)) separately to each voxel in the image and generating a map of test statistics for a variable such as task activation (relative to baseline), age, or diagnosis (Soares *et al.* 2016). Statistical inference is then carried out using either voxel-based thresholding or cluster-extent thresholding, which use different methods to account for the high number of tests (Soares *et al.* 2016). Cluster-extent thresholding is more common (Vandekar and Stephens 2021; Woo *et al.* 2014; Soares *et al.* 2016). In this approach, the statistical map is first thresholded based on a cluster forming threshold (CFT), often set to an uncorrected p value of 0.01 or 0.001 (Woo *et al.* 2014). Next, p values for the resulting suprathreshold clusters are computed based on the estimated null distribution of cluster sizes, using methods such as Gaussian Random Field Theory (RFT) to control the family-wise error rate (FWER) (Woo *et al.* 2014).

These methods relying on statistical significance for interpretation of results have received criticism, many of which are similar to the general criticism of hypothesis testing, in recent years (Vandekar and Stephens 2021; Chen *et al.* 2017; Woo *et al.* 2014). For example, the method of cluster-extent thresholding described above is subject to the same pitfall of being highly dependent on sample size. As sample size increases, the ability to detect differences of

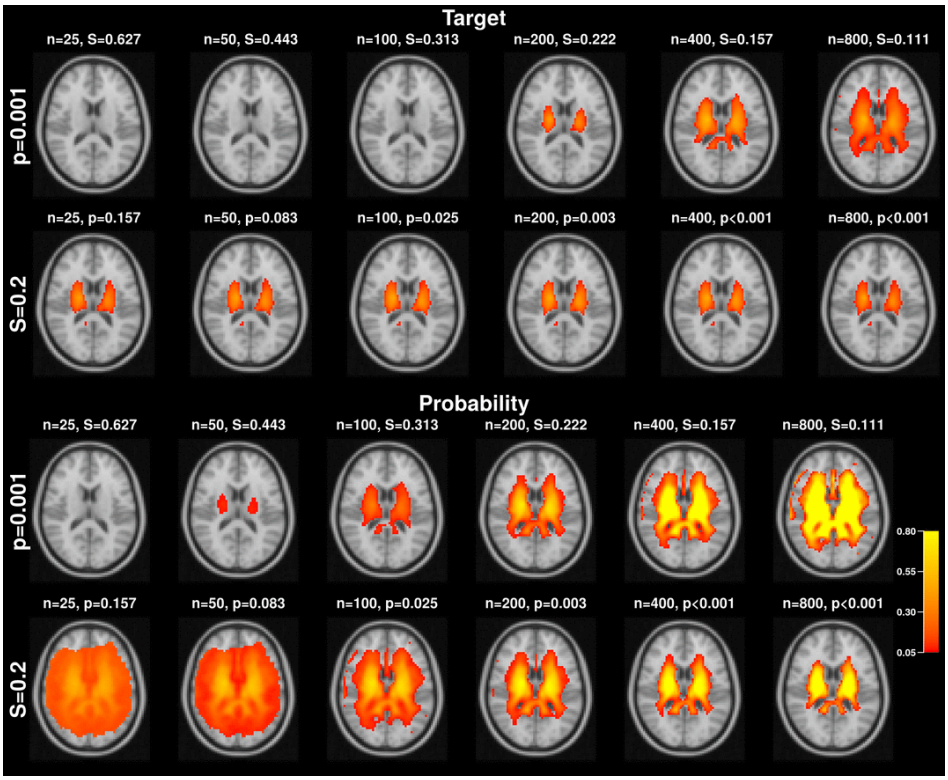

Figure 1: Illustration of the impact of sample size on p value thresholding vs. effect size thresholding. This is Figure 2 of (Vandekar and Stephens 2021), where S represents a standardized effect size (RESI). The top two rows show regions with a true expected suprathreshold value, and the bottom two rows show the probability of a voxel being found in the target set in random samples. Sample sizes increase left to right.

little clinical meaning increases, and can eventually result in brain maps where all voxels are deemed statistically significant, an illustration of the null hypothesis fallacy (Rozeboom 1960; Bowring *et al.* 2019). While the null hypothesis assumes that the noise has a mean of zero, this is not met in real neuroimaging data (Vandekar and Stephens 2021; Bowring *et al.* 2019; Gonzalez-Castillo *et al.* 2012). With the advent of large neuroimaging datasets such as the Adolescent Brain Cognitive Development Study (ABCD) and Human Connectome Project (HCP), this issue is becoming more evident in practice and complicates interpretation of results (Bowring *et al.* 2019).

Results for associational studies are often visualized with colored maps of the brain, with statistically significant voxels or clusters of voxels colored according to their test statistic or p value. Chen *et al.* (2017) argues that presenting the results in this manner is not ideal as there is no indication of the magnitude of the effect, or how meaningful it is in practice. Additionally, they argue neuroimaging studies are often underpowered, due low sample sizes, large variability between subjects, and a low signal-to-noise ratio. This leads to increased Type II errors, and may lead readers to interpret the lack of a significant effect in a region of the brain as proof of no effect. These issues can increase the type M error, or the probability of under- or over-estimating the magnitude of the effect, in neuroimaging studies (Chen *et al.* 2017).

Use of effect sizes, on the other hand, can alleviate some of these issues in neuroimaging. For example, cluster-based extent thresholding can replace p value thresholds with a relevant standardized effect size threshold (Vandekar and Stephens 2021). As the effect size is not dependent on sample size, it does not exhibit the same problem of increasingly large clusters with growing sample size (Vandekar and Stephens 2021) (see Figure 1). Additionally, the use of effect sizes directly conveys the magnitude of the effect, allowing readers to more easily discern the practical significance of results. Chen *et al.* (2017) argues that results should be displayed using colorized brain maps of effect size estimates. These can be unstandardized, such as the percent change in the BOLD signal, or can use a standardized effect size for greater comparability across settings.

While there is increasing encouragement for the use and reporting of effect sizes in neuroimaging studies, there are challenges both to widespread reporting of effect sizes across fields and specific challenges applicable to high-dimensional neuroimaging problems. The next section will describe broadly the challenges in effect size and confidence interval reporting and introduce a practical tool for computing standardized effect size estimates that can be applied in many modeling settings, including those commonly used in mass univariate neuroimaging analyses.

2. RESI: An R Package for Robust Effect Sizes

2.1. Background

As discussed above, the reporting of effect sizes is encouraged but often underutilized. There are four challenges to reporting effect sizes that limit their widespread use. First, there are many different effect size measures available (Cohen 1988; Hedges and Olkin 1985; Rosenthal 1994; Zhang and Schoeps 1997; Serdar *et al.* 2021), but they are typically defined in the context of a specific population parameter, which makes comparing effects across a wide range of models difficult (Vandekar *et al.* 2020). Second, many available effect size measures do not allow for nuisance parameters or covariates (Vandekar *et al.* 2020). Third, many effect size measures do not have accurate confidence interval procedures, which precludes quantification of the uncertainty around the effect size estimate (Kang *et al.* 2023). Finally, many default model summary functions available in statistical software automatically output p values, but few also report effect sizes with confidence intervals. The **RESI** R package was designed to address these challenges by implementing a recently proposed effect size measure.

There are several R packages available for effect size calculation. For example, packages such as **MOTE**, **MBESS**, **effsize**, **esvis**, **lsr**, **esc**, and **rcompanion** include functionality that allows the user to manually input data or the relevant test statistics for conversion to a variety of desired effect size measures (Buchanan *et al.* 2019; Kelley 2022; Torchiano 2020; Anderson 2020; Navarro 2015; Lüdecke 2019; Mangiafico 2023). **MOTE**, **effsize**, and **esc** compute confidence intervals for effect sizes using noncentral or central distributions. **rcompanion** utilizes bootstrapping for effect size confidence intervals, and **MBESS** implements both noncentral and bootstrapped confidence intervals. The **effectsize** package implements many effect size measures and conversions between some of them (Ben-Shachar *et al.* 2020). **effectsize** allows users to input test statistics, but also conveniently accepts fitted models directly to compute the desired effect size. This package computes confidence intervals in a variety of methods

depending on the effect size measure, but several functions use a noncentral distribution and a few use bootstrapping. The **emmeans** package also allows post-model-fitting effect size (Cohen's d) estimation for contrasts of estimated marginal means in `emmGrid` objects and computes parametric confidence intervals (Lenth *et al.* 2021). Another package, **bootES**, focuses specifically on providing bootstrapped confidence intervals for unstandardized effect sizes, Cohen's d, Pearson correlation, and Hedges's g (Gerlanc and Kirby 2023). Each of these tools can be helpful for computing effect sizes in a given data context; however, they do not fully address the general challenges to reporting and comparing effect sizes mentioned above. In particular, many confidence interval methods for effect sizes rely on Chi-square, F, or t statistic implementations, which have below nominal coverage rate (Kang *et al.* 2023). There is a need for an effect size index that can be broadly applied and compared across model types and user-friendly software tools that implement such a measure to promote easy reporting of effect sizes.

The recently proposed robust effect size index (RESI) (Vandekar *et al.* 2020; Kang *et al.* 2023) addresses many of these challenges because it is broadly applicable across all common model types, it accommodates nuisance parameters, and there is an effective confidence interval procedure available (Kang *et al.* 2023). The RESI can be estimated from Chi-square, F, Z, and t statistics. It is also possible to convert RESI estimates to and from other common effect size measures, such as Cohen's d, Cohen's f^2, and R^2 (Vandekar *et al.* 2020).

The **RESI** R package builds on existing infrastructure for robust standard error estimation (Zeileis 2006) and bootstrapping (Canty and Ripley 2022) allowing easy estimation, reporting, and visualization (Jones *et al.* 2023b). Similarly to the **effectsize** package, **RESI** is designed to work on model inputs, so that effect size estimates can be easily obtained in tandem with common model summaries. These model-based functions also allow for a large amount of customization in the estimation and reporting process. Directly inputting test statistics and the relevant degrees of freedom and sample size is an option as well, helpful for model types that have not yet been implemented via dedicated methods in the package. The package also aims to work with other effect size measures, providing functions to convert to and from a few common effect size indices. Plotting functions are provided to allow for quick visualization of the effect size estimates present in models. With these tools, we hope to make obtaining the highly generalizable RESI simple and accessible, in order to increase ease of reporting effect sizes in research. In the following sections, we outline the theory underlying the RESI, its estimators, and confidence interval procedure. We then discuss the **RESI** package, its structure, function arguments, and dependencies. Finally, we provide three in-depth examples using the **RESI** package to perform analysis of effect sizes, from model creation to post-estimation visualization.

2.2. Statistical methods

RESI definition

The RESI is defined from the noncentrality parameter of a test statistic in the context of M-estimation, so it is broadly applicable across statistical models and parameters. A full introduction to the RESI can be found in our previous work (Vandekar *et al.* 2020; Kang *et al.* 2023).

Briefly, consider a dataset of independent observations $W = \{W_1, \ldots, W_n\}$ with probability

distribution $\mathcal{P}$ and let $\theta = (\alpha, \beta) \in \mathbb{R}^m$ be a vector of parameters with $\alpha \in \mathbb{R}^{m_0}$ nuisance parameters and $\beta \in \mathbb{R}^{m_1}$ the target parameters of interest. The RESI is constructed using the test statistic for the null hypothesis $H_0 : \beta_0 \in \mathbb{R}^{m_1}$, where β_0 is a reference value, usually zero (Vandekar and Stephens 2021). Assuming known variance, the usual Wald-style test statistic, centered at the reference value, $T^2 = n(\hat{\beta} - \beta_0)^\top \Sigma_\beta^{-1}(\hat{\theta})(\hat{\beta} - \beta_0)$ follows a Chi-square distribution with m_1 degrees of freedom and noncentrality parameter $n(\beta - \beta_0)^\top \Sigma_\beta^{-1}(\beta - \beta_0)$. The RESI, S_β, is the square root of the component of the noncentrality parameter that does not depend on the sample size

$$S_\beta = \sqrt{(\beta - \beta_0)^\top \Sigma_\beta^{-1}(\beta - \beta_0)}.$$

RESI estimators

The RESI is very general, because its estimator can be computed for Chi-square, F, Z, and t statistics (Vandekar *et al.* 2020; Kang *et al.* 2023). In this section, we review these estimators and introduce new estimators for a modified RESI using Z and t statistics, which have the advantage that the proposed modification shows the direction of the effect for univariate parameters. We also describe the use of robust covariance in the estimation of the test statistics.

The original estimator for S_β was developed using an estimator for noncentrality parameters of Chi-square statistics (Vandekar *et al.* 2020)

$$\hat{S}_\beta = \left\{ \max\left[0, \frac{T^2 - m_1}{n}\right] \right\}^{\frac{1}{2}}. \tag{1}$$

Because S_β is nonnegative, the max operator ensures that the estimator is also nonnegative in finite samples.

Under normality, the finite sample distribution of the asymptotic Chi-square statistic divided by its degrees of freedom is an F distribution (Mantel 1963). When this is true, a better small sample estimator can be computed using method of moments with the F distribution

$$\hat{S}_\beta = \left\{ \max\left[0, \frac{F \times (n - m - 2) - m_1 \times (n - m)}{n \times (n - m)}\right] \right\}^{\frac{1}{2}}. \tag{2}$$

The RESI is called robust because its estimator is consistent under misspecification of the variance model when estimated with a robust test statistic (Vandekar *et al.* 2020; MacKinnon and White 1985), which uses a heteroskedastic consistent sandwich estimator for Σ_β (White 1980; MacKinnon and White 1985; Long and Ervin 2000). The RESI estimator is a consistent estimator of the true effect size in contexts where the robust variance estimator yields consistent results under aspects of model misspecification, such as unknown heteroskedasticity between measurements in general linear models or a misspecified correlation structure in GEE models. When the mean model is misspecified, the RESI is a consistent estimator of the best approximation of the true model within the class of models considered (Boos and Stefanski 2013).

With Equation 1 and Equation 2, we can compute RESI estimates for Chi-square and F statistics, which are easily obtained from many statistical models. However, these estimates

have the feature of being nonnegative, so they do not describe the direction of an effect. While this makes them generally applicable across univariate (can be negative and positive) and multivariate (can only be positive) parameters, for univariate parameters it is also useful to be able to obtain a signed effect size estimate, showing the directionality of the effect. With this in mind, we introduce RESI estimators for Z and t statistics. We use two approaches to develop these estimators, leading to two estimators with different properties and advantages.

The first approach is the same as the development of the RESI estimators for Chi-square and F statistic. We use the method of moments for the Z or t statistics to find estimators for S_β. Consider a Z statistic, whose expected value is $\mathsf{E}Z = \sqrt{n}\text{sgn}(\beta)S_\beta$, where sgn is the sign function, which leads to the signed RESI estimator

$$\hat{S}_\beta = \frac{Z}{\sqrt{n}} \tag{3}$$

For a t statistic with degrees of freedom $n-m$ and noncentrality parameter $\sqrt{n}S_\beta$, when $n-m>1$, the expected value is $\mathsf{E}t = \sqrt{\frac{n(n-m)}{2}}\frac{\Gamma((n-m-1)/2)}{\Gamma((n-m)/2)}S_\beta$. This gives the RESI estimator

$$\hat{S}_\beta = \frac{t\sqrt{2}\Gamma((n-m)/2)}{\sqrt{n(n-m)}\Gamma((n-m-1)/2)} \tag{4}$$

The advantage of estimators in Equation 3 and Equation 4 is that both are unbiased for S.

The second approach leverages the relationship between Z and Chi-square statistics and t and F statistics. Squaring a Z or t statistic gives a Chi-square or F statistic, respectively. We then use Equation 1 and Equation 2 as RESI estimators for Z and t statistics by multiplying them with the sign of the test statistic. For example,

$$\hat{S}_\beta = \text{sgn}(Z)\times\left\{\max\left[0, \frac{Z^2-1}{n}\right]\right\}^{\frac{1}{2}}, \tag{5}$$

and similarly for the t estimator. These estimators are biased, but consistent and have smaller mean squared error than the estimators in Equation 3 and Equation 4. These estimators are advantageous because their estimates are equal in absolute value to the unsigned RESI estimates, whereas the estimators in Equation 3 and Equation 4 are not.

Note that the RESI estimators are based on the Wald test statistics, which are dependent on the specific modeling decisions made when fitting the data. The RESI is linear on the scale of the linear predictor (e.g., the RESI for a logistic model is linear on the log-odds scale).

Bootstrapping procedure for confidence intervals

In recent work, we showed that Chi-square and F confidence intervals are not accurate for computing effect size confidence intervals in general. In particular, when the test statistic is estimated using a robust covariance estimator, or when the study design is observational, using a Chi-square or F distribution for the RESI estimate underestimates the variance and will therefore produce confidence intervals that provide less than nominal coverage level (Kang *et al.* 2023). These Chi-square and F confidence intervals with below nominal coverage are those that are implemented in most other software packages (SAS Institute Inc. 2016; Buchanan *et al.* 2019; Torchiano 2020; Lüdecke 2019; Ben-Shachar *et al.* 2020; Kelley 2022). As an alternative, we proposed a nonparametric bootstrap for the RESI confidence interval

Cohen's d	RESI	"Rule of Thumb" Interpretation
[0, 0.2]	[0, 0.1]	No effect - small
(0.2, 0.5]	(0.1, 0.25]	Small - medium
(0.5, 0.8]	(0.25, 0.4]	Medium - large
> 0.8	> 0.4	Large

Table 1: Guidelines for interpreting size of (absolute) RESI estimates based on analogous suggestions from (Cohen 1988). Note these ranges are a rule of thumb and effect sizes should always be interpreted within the scientific context.

because it produces confidence intervals with nominal coverage most consistently (Kang *et al.* 2023). This is the procedure implemented in the **RESI** package. For linear models and nonlinear least squares models, a Bayesian bootstrap is also available as an option (Rubin 1981).

Meaningful RESI ranges

When interpreting the RESI estimates, it is useful to have an idea of what constitutes a "large" or "small" effect. While a meaningful effect size (standardized or unstandardized) ultimately depends on the scientific context, ranges can be posited based on recommended effect size ranges for Cohen's d assuming equal sample proportions of the two groups and equal variance (Cohen 1988, Table 1). These are guidelines based on a difference in means in behavioral sciences and the size of a meaningful effect varies by field; effort should be made to interpret estimates within the given scientific context.

2.3. Package Details

RESI is available to the public via The Comprehensive R Archive Network (CRAN) under the GPL-3 license. To download, one can use the following code:

```
R> install.packages("RESI")
```

The development version is available on GitHub. This can be downloaded using the **devtools** package with the following command (Wickham *et al.* 2022b):

```
R> devtools::install_github("statimagcoll/RESI")
```

Operation

Users should have R version 2.10 or higher to use **RESI** (R Core Team 2023). The **RESI** package is designed to easily add RESI estimates and confidence intervals to common model outputs, such as coefficient summaries and ANOVA tables. The functions in the package are split into three categories: model-based functions, conversion functions, and additional methods to other functions (Figure 2). There are also two datasets provided.

Model-based functions

The main model-based RESI estimation functions of the **RESI** package are `resi_pe()`, to obtain point estimates, and `resi()` for point estimates with confidence or credible intervals.

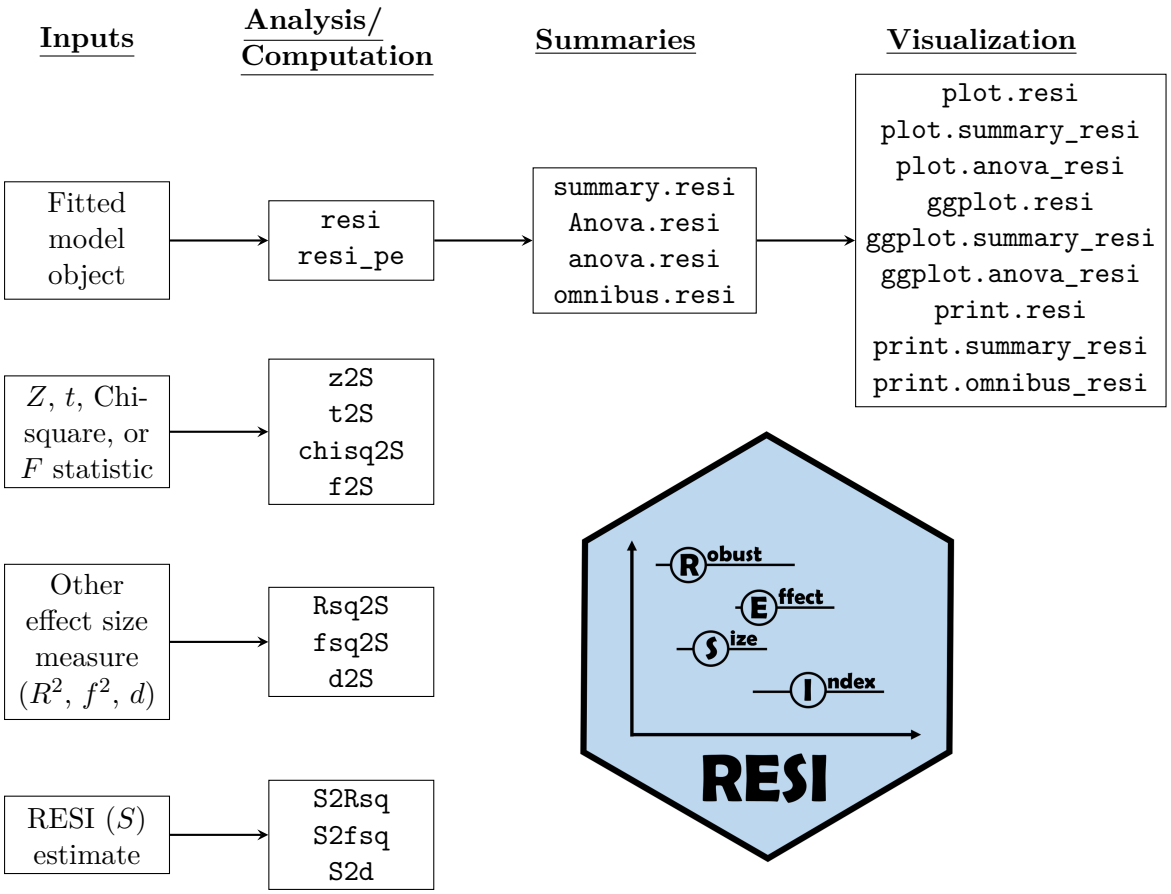

Figure 2: **RESI** Package Structure and Logo. Inputs to package functions can be models of supported types, test statistics with relevant degrees of freedom and sample size, or effect size measures. The analysis functions compute RESI estimates with or without confidence intervals, or convert to and from other effect size indices. Summary functions provide relevant information extracted from a '`resi`' object. Post-estimation visualization functions include plotting and printing.

`resi_pe()` uses standard summary and ANOVA outputs to compute the RESI point estimate. `resi()` uses `resi_pe()` and performs bootstrapping (via the **boot** package (Canty and Ripley 2022)) to produce confidence intervals for the RESI. These functions take supported fitted models as input and return a list that contains three main components: a coefficients summary table with a row for each non-reference level of each variable, an ANOVA table containing a row for each variable, and an overall RESI estimate. Details regarding functions used for table construction for the supported model types are given in Table 2. For model classes without dedicated methods, `resi()`/`resi_pe()` attempts to implement the default method and return informative messages in the case of failure.

While the user can simply run `resi()` on a supported model type and obtain a full output, there are several arguments that can be used to tailor the process. Details for all function arguments are available in the documentation, but we briefly cover important arguments here. `resi()` and `resi_pe()` both contain the following arguments. The `model.full` argument is the model to perform RESI estimation on. The `model.reduced` argument, `NULL` by default, specifies a reduced model which is used to compute an effect size estimate in comparison to the full model for a specific subset of variables that the user wishes to compare (See RESI on survival model). If left as `NULL`, `resi_pe()` will compute a reduced model of the same type as the full model, but including only the intercept term. `data` is a blank argument referring to the data used to generate the model. If left blank, `resi()` pulls the data from the model. For some model types ('`survreg`', '`coxph`', '`nls`'), the data is required as an input because these models objects do not store the original data frame used to fit the model. Additionally, when using some formula functions such as splines or factoring, the data needs to be input so that the spline arguments can be recomputed as they were in the original data.

The `vcovfunc` argument can be used to specify a different variance-covariance function and is important because it affects whether the effect size is robust to model misspecification (see RESI Estimators). By default, RESI will use a robust covariance estimator. Additional arguments to the given `vcovfunc` function can be specified in list form with the `vcov.args` argument. Similarly, additional arguments to the `Anova()` function (from **car** (Fox and Weisberg 2019)) can be specified with the `Anova.args` argument. The `unbiased` argument is logical (default `TRUE`) and corresponds to a choice of conversion formulas for the Z and t statistics (see RESI Estimators for details).

`resi()` contains additional arguments related to the bootstrap procedure. The confidence level (default 0.05) for the confidence or credible intervals can be specified with `alpha`. Multiple confidence levels can be specified using a numeric vector. For '`lm`' and '`nls`' models, there is a `boot.method` argument that can be specified as nonparametric (default) or Bayesian (see Bootstrapping). Bootstrapping is implemented via the `boot()` function from the **boot** package (Canty and Ripley 2022), so `resi()` accepts additional arguments corresponding to that function such as `parallel` and `ncpus` to allow for increased efficiency. Finally, the `store.boot` argument (default `FALSE`) determines whether to store the complete '`boot`' object as an element of the output. The `store.boot` option must be set to `TRUE` if the user wants to be able to obtain confidence intervals with different confidence levels without rerunning the bootstrap procedure.

The output of `resi()` is a list of class '`resi`' that contains the three main tables (coefficients, ANOVA, and overall) with confidence intervals and several other elements to track how the functions were called. `resi_pe()` produces a list with these tables (without confidence intervals) and other elements about the model. Each of these elements is optionally computed

and can be suppressed with an argument to `resi()`/`resi_pe()` (e.g., `anova = FALSE`).

The `overall` element of the output is a table reporting a Wald test comparing the full model to the reduced model. The test statistic is typically converted from a Chi-square statistic to a RESI estimate internally using the `chisq2S()` function, which takes the number of observations from the data and degrees of freedom from the Wald test. In the case of a linear model, the RESI estimate is computed using the `f2S()` function.

The `coefficients` table is available for every model type supported by the package. This provides a RESI estimate for each model coefficient and appends it to a table resulting from one of the "Coefficients" functions in Table 2. The Z or t statistic from this function is converted to the signed RESI via `z2S()`/`t2S()`, with the logical `unbiased` argument determining if the unbiased estimator in Equation 3 and Equation 4 or the alternate version in Equation 5 is used.

The `anova` table is computed via `Anova()` from **car** (Fox and Weisberg 2019) where available (for '`geeglm`' models, `anova` is used). The `anova` argument (default = `TRUE`) in `resi()`/`resi_pe()` determines whether to compute this table. For '`lm`' models, an F-test is used. For the others, a Wald test is specified assuming Chi-square statistics. Other options can be passed to `Anova()` function via `Anova.args`. Note that the `test.statistic` argument is fixed in the `resi_pe()` function, so supplying a different value for this argument will result in an error. Additionally, if the user wishes to use a different `vcov.` argument in `Anova()` function, this should be done by providing the function to the `vcovfunc` argument in `resi()` (see RESI on linear model). Specifying this argument in `Anova.args` will result in an error. The resulting Chi-square or F statistics are converted to RESI estimates using `chisq2S()` or `f2S()`.

The RESI for longitudinal models is still in development. Currently, the package provides point estimate and confidence interval methods for '`gee`' (from **gee** package (Carey 2022)) and '`geeglm`' (from **geepack** (Halekoh *et al.* 2006)) models. For these models, both a longitudinal RESI and a per-measurement cross-sectional RESI estimate are computed for each factor in the `coefficients` table (for '`gee`' and '`geeglm`') and for each variable in the `anova` table (for '`geeglm`'). The longitudinal RESI is the estimated effect conditional on the sampling design, whereas the cross-sectional estimator is the effect if the data were collected cross-sectionally. This allows investigators to quantify the benefit conferred by considering a longitudinal design. For linear mixed effects models fit via `lme()` from **nlme** (Pinheiro *et al.* 2023) and `lmerMod()` from **lme4** (Bates *et al.* 2015), longitudinal RESI point estimation is available in both a `coefficients` and `anova` table. The confidence interval procedure is still being evaluated for these models, so running `resi()` on a model of this type will provide point estimates only with a corresponding message.

Other package elements

The package includes `print()`, `plot()`, `ggplot()` `summary()`, `anova()`, and `car::Anova()` methods for '`resi`' objects. The `summary()` and `anova()`/`Anova()` methods are intended to isolate the corresponding elements of the '`resi`' object and allow the user to specify a different confidence level without having to rerun the bootstrapping process, if the `store.boot` option was set to `TRUE` when running `resi()`. Running `summary()` on a '`resi`' object returns the `coefficients` table as an object of class '`summary_resi`', with its own `plot()`/`ggplot()` and `print()` methods. Running `anova()` or `car::Anova()` on a '`resi`' object returns the `anova`

Model	Package	Covariance	Coefficients	Anova	Overall
'lm'	**stats**	sandwich::vcovHC	coeftest	car::Anova	waldtest
'glm'	**stats**	sandwich::vcovHC	coeftest	car::Anova	waldtest
'nls'	**stats**	regtools::nlshc	coeftest	N/A	wald.test
'survreg'	**survival**	vcov	coeftest	car::Anova	waldtest
'coxph'	**survival**	vcov	coeftest	car::Anova	wald.test
'hurdle'	**pscl**	sandwich::sandwich	coeftest	N/A	waldtest
'zeroinfl'	**pscl**	sandwich::sandwich	coeftest	N/A	waldtest
'gee'	**gee**	summary	summary	N/A	N/A
'geeglm'	**geepack**	vcov	coeftest	anova	anova
'lme'	**nlme**	clubSandwich::vcovCR	summary	car::Anova	N/A
'lmerMod'	**lme4**	clubSandwich::vcovCR	summary	car::Anova	N/A

Table 2: Supported model types and related functions. `coeftest` and `waldtest` are from **lmtest** (Zeileis and Hothorn 2002). `wald.test` is from **aod** (Lesnoff *et al.* 2012).

table as an object of class 'anova_resi' and inherited classes from `anova()`/`car::Anova()`. There are also `plot()`/`ggplot()` (Wickham 2016) methods for 'anova_resi'.

The package also contains a few conversion functions from RESI to and from other common effect size measures. These are Cohen's d, Cohen's f^2, and R^2. Formulas for these conversions are found in (Vandekar *et al.* 2020).

Lastly, the **RESI** package contains two datasets. The `insurance` dataset is adapted from the open-source repository Kaggle (US Health Insurance Dataset) and the `depression` dataset is adapted from a data analysis textbook (Agresti 2002). Full details on the datasets are provided in the **RESI** package documentation.

Important dependencies

The **RESI** package currently has dedicated methods for 11 model types (Table 2). The software function used to compute the covariance matrix varies by model type. It is possible to pass additional arguments to these covariance functions in `resi` by using the `vcov.args` argument. Any other valid covariance function can be specified as well. Although robust covariance estimators are used as the default for most model types, the survival models ('survreg' and 'coxph') have the option for a robust covariance estimate in model setup and, when using the standard `vcov` from stats, they compute robust covariance matrices if the argument `robust=TRUE`. For 'geeglm' models, the robust covariance is taken from the model directly (Zeileis 2006).

Several other common analysis functions are used to obtain test statistics for RESI computation for the coefficients, ANOVA, and overall table. The functions used for different model types are found in Table 2.

Support

Users encountering problems with the package can reach out for help using the GitHub Issues page. The Discussions page can be used to seek additional support or suggest new features to be added to the package. Planned features are listed in the Coming Soon section of our **pkgdown** website (Wickham *et al.* 2022a). We ask those wishing to contribute to our software

to create a new branch on our GitHub and submit a pull request describing the contribution.

2.4. Illustrations

To demonstrate the flexibility of the **RESI** package, we analyze a few example datasets for several different model types using different covariance estimator functions and bootstrapping options. For space considerations, only the first illustration (linear model) is included here; please see the Appendix for additional illustrations for nonlinear least squares and survival models.

RESI on linear model

We first look at a linear model fit using `lm()`. After installing the package from CRAN or GitHub, we load the **RESI** library.

```
R> library("RESI")
```

We will use the `insurance` dataset in the package to fit our model. The dataset contains information on insurance charges, age, sex, BMI, number of children, smoking status, and geographical region for 1338 individuals in the United States. We fit a linear regression of charges against region, age, BMI, and sex, with an interaction term on region and age and return the standard coefficients table using the `summary()` function.

```
R> mod_lm <- lm(charges ~ region * age + sex + bmi, data = insurance)
R> summary(mod_lm)

Call:
lm(formula = charges ~ region * age + sex + bmi, data = insurance)

Residuals:
   Min     1Q Median     3Q    Max
-14871  -7062  -4885   6235  46347

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         -5359.44    2369.09  -2.262   0.0238 *
regionnorthwest     -2339.44    2647.85  -0.884   0.3771
regionsoutheast     -3230.85    2583.12  -1.251   0.2112
regionsouthwest      -232.48    2662.84  -0.087   0.9304
age                   220.33      45.08   4.888 1.14e-06 ***
sexmale              1328.02     622.07   2.135   0.0330 *
bmi                   323.77      53.72   6.027 2.17e-09 ***
regionnorthwest:age    34.90      63.55   0.549   0.5829
regionsoutheast:age    83.64      61.65   1.357   0.1751
regionsouthwest:age   -33.63      63.74  -0.528   0.5979
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 11360 on 1328 degrees of freedom
Multiple R-squared:  0.126,        Adjusted R-squared:  0.1201
F-statistic: 21.27 on 9 and 1328 DF,  p-value: < 2.2e-16
```

The p values in the standard model summary indicate that age, sex, and BMI are significantly associated with insurance charges. However, just by looking at the p values, it is hard to discern the strength of the the association. We would like to be able to see, in addition to significance, a measure of the effect size. To accomplish this, we can run `resi()` on the model object. We run it using all the default options first. This will use the `vcovHC()` function from the **sandwich** package (with default arguments) to compute robust standard error estimates (Zeileis *et al.* 2020). Since we are using the `resi()` function rather than the `resi_pe()` function, we will obtain bootstrapped confidence intervals in addition to RESI point estimates. We set the seed to ensure the results are reproducible. This function can take several seconds to run. Printing the full '`resi`' object will print several tables and notes, so to begin we just print the summary.

```
R> set.seed(0827)
R> resi_obj_lm <- resi(mod_lm)
R> summary(resi_obj_lm)

Analysis of effect sizes based on RESI:
Confidence level =  0.05
Call:  lm(formula = charges ~ region * age + sex + bmi, data = insurance)

Coefficient Table
                    Estimate Std. Error t value Pr(>|t|)   RESI   2.5%
(Intercept)         -5359.44    2175.94  -2.463    0.014 -0.067 -0.119
regionnorthwest     -2339.44    2395.15  -0.977    0.329 -0.027 -0.084
regionsoutheast     -3230.85    2643.11  -1.222    0.222 -0.033 -0.087
regionsouthwest      -232.48    2574.28  -0.090    0.928 -0.003 -0.060
age                   220.33      40.21   5.480    0.000  0.150  0.094
sexmale              1328.02     621.74   2.136    0.033  0.058  0.004
bmi                   323.77      58.08   5.574    0.000  0.152  0.105
regionnorthwest:age    34.90      57.24   0.610    0.542  0.017 -0.037
regionsoutheast:age    83.64      63.33   1.321    0.187  0.036 -0.016
regionsouthwest:age   -33.63      61.41  -0.548    0.584 -0.015 -0.070
                      97.5%
(Intercept)          -0.014
regionnorthwest       0.031
regionsoutheast       0.020
regionsouthwest       0.054
age                   0.210
sexmale               0.108
bmi                   0.199
regionnorthwest:age   0.072
regionsoutheast:age   0.090
```

```
regionsouthwest:age    0.037
```

This output shows the `coefficients` element of the '`resi`' object, as well as the model call and the confidence level (α, by default = 0.05). The coefficient table looks very similar to the standard model summary output. The estimates will remain unchanged, but the standard errors differ because `summary()` uses the model-based (naive) standard error, whereas `resi()` defaults to use a robust estimate. These standard error estimates will remain valid under heteroskedasticity. Accordingly, the t values and p values are different, but our qualitative conclusions about statistical significance are unchanged in this example. The three rightmost columns are new and represent the RESI estimates and $(1-\alpha)\%$ confidence intervals. Note that a RESI estimate further from 0 indicates a larger effect. The sign of the RESI estimate indicates the direction of the effect. From the table we can see that BMI is estimated to have a small to moderate effect (0.152 (CI: 0.105, 0.199)) based on the ranges given in Meaningful RESI Ranges. Sex is estimated to have a small effect (0.058 (CI: 0.004, 0.108)). The effect size estimates are conditional on the other terms in the model. Because our model includes an interaction on age and region, the RESI estimate for "age" in the coefficient table is interpreted as the estimated effect size of age for those in the northeast (the reference region). This is estimated to be 0.150 (CI: 0.094, 0.210), a small to moderate effect. For these results, if the p value is less than 0.05, then the CI for the RESI does not contain 0. This will not always be the case because the RESI CI is estimated for the distribution of the effect size estimator under the alternative.

Because region is a factor variable, the test for region and its interaction with age corresponds to multiple parameters in the model. To obtain an effect size estimate for multiple parameters that correspond to a single variable, we can report the ANOVA table. We can obtain this with either the standard `anova()` function or the `car::Anova()` function on the '`resi`' object.

```
R> anova(resi_obj_lm)

Analysis of Deviance Table (Type II tests)

Response: charges
           Df      F Pr(>F)   RESI  2.5% 97.5%
region      3   1.60  0.189 0.0365 0.000 0.120
age         1 117.70  0.000 0.2951 0.240 0.360
sex         1   4.56  0.033 0.0515 0.000 0.105
bmi         1  31.07  0.000 0.1498 0.101 0.197
region:age  3   1.12  0.341 0.0161 0.000 0.101
```

By default, `resi()` uses a Type II sum of squares, but this can be changed in the arguments (Papachristodoulou and Prajna 2005). This output is the same as running `car::Anova()` on the model using `sandwich::vcovHC` as the `.vcov` argument, but with the three rightmost columns added for the RESI estimates and confidence intervals. The interpretation of the RESI is the same as the coefficient table, but we note that in the ANOVA table, the RESI estimates are all nonnegative because they are estimated from F statistics. The estimates in the ANOVA table differ for two reasons: (1) Type II sum of squares first tests main effects

without their interactions in the model; for example, the "age" RESI estimate is interpreted as the effect of age compared to a model that does not include age or the interaction term for age and region. (2) For variables that are tested on 1 degree of freedom, the ANOVA table estimates the absolute effect size, whereas the coefficient table uses the unbiased signed effect size by default (see RESI Estimators). For example, with sex and BMI, we notice that the estimates are close in the ANOVA and coefficients tables, but not exactly equal in absolute value. This is due to using the default `unbiased = TRUE` argument, which uses the t to S estimator from Equation 4 rather than the one based on the F to S formula.

An overall Wald test is also reported in the model.

```
R> omnibus(resi_obj_lm)

Analysis of effect sizes based on RESI:
Confidence level =  0.05
Wald test

Model 1: charges ~ 1
Model 2: charges ~ region * age + sex + bmi
  Res.Df Df      F Pr(>F)  RESI  2.5% 97.5%
1   1337
2   1328  9 20.249      0 0.360 0.326 0.421
```

By default, this compares the model to a reduced model that has only the intercept. The RESI estimate represents the overall absolute effect size of the model. In this model, this is estimated to be 0.360 (CI: 0.326, 0.421). This is interpreted as a moderate to large effect.

We can also visualize the results using the `plot()` or `ggplot()` function (Figure 3). Running these functions on the '`resi`' object will plot the coefficient table. Margins will be automatically adjusted to accommodate long variable names, or this feature can be turned off with the argument `automar = FALSE`. Alternatively, the user can extract the estimates and CIs from the tables and plot using their preferred visualization tool.

```
R> library("ggplot2")
R> plot(resi_obj_lm)
R> ggplot(anova(resi_obj_lm))
```

If we want to see a plot of the ANOVA table, we can run `plot()`/`ggplot()` either directly on the `anova` element of the '`resi`' object or on `anova()` or `car::Anova()` on the '`resi`' object. These plots help us quickly visualize the RESI estimates and relative effect sizes of the variables.

If we want to use different arguments for the covariance estimator function or the ANOVA function, we can specify these using the `vcov.args` and `Anova.args` arguments, respectively, in the `resi()` function. For example, we can use the `sandwich::vcovHC()` function with the HC0 estimator instead of the default HC3 estimator (Long and Ervin 2000) and use Type III sum of squares instead of Type II as follows.

```
R> set.seed(0827)
R> resi_obj_lm2 <- resi(mod_lm, vcov.args = list(type = "HC0"),
```

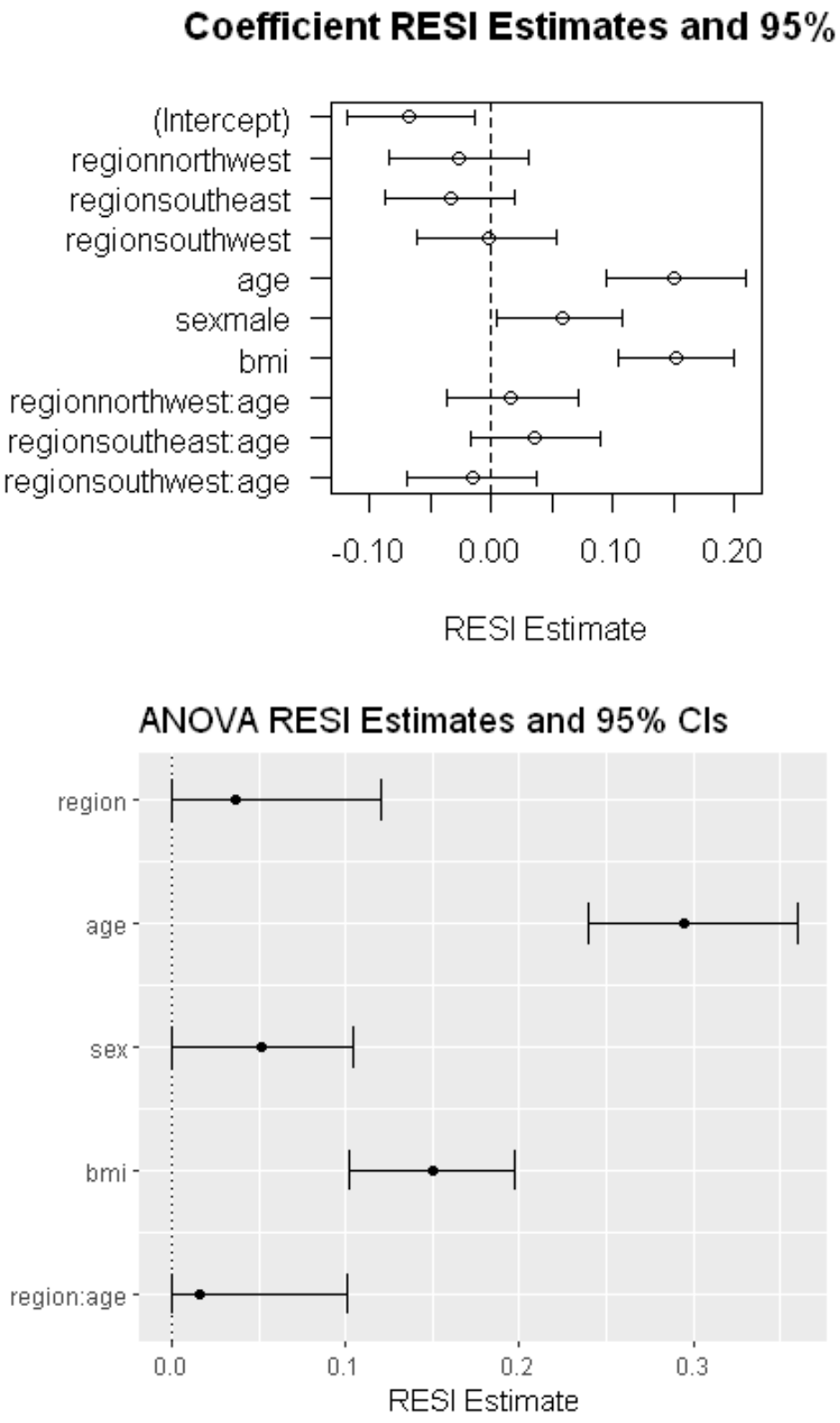

Figure 3: RESI estimates and confidence intervals from linear model coefficients and ANOVA tables.

```
R> Anova.args = list(type = 3))
R> resi_obj_lm2

Analysis of effect sizes based on RESI:
Confidence level =  0.05
Call:  lm(formula = charges ~ region * age + sex + bmi, data = insurance)

Coefficient Table
                    Estimate Std. Error t value Pr(>|t|)   RESI   2.5%
(Intercept)         -5359.44    2155.75  -2.486    0.013 -0.068 -0.120
regionnorthwest     -2339.44    2372.36  -0.986    0.324 -0.027 -0.085
regionsoutheast     -3230.85    2618.64  -1.234    0.218 -0.034 -0.088
regionsouthwest      -232.48    2549.50  -0.091    0.927 -0.003 -0.061
age                   220.33      39.82   5.534    0.000  0.151  0.095
sexmale              1328.02     617.05   2.152    0.032  0.059  0.004
bmi                   323.77      57.56   5.625    0.000  0.154  0.106
regionnorthwest:age    34.90      56.68   0.616    0.538  0.017 -0.037
regionsoutheast:age    83.64      62.72   1.333    0.183  0.036 -0.016
regionsouthwest:age   -33.63      60.78  -0.553    0.580 -0.015 -0.070
                       97.5%
(Intercept)           -0.014
regionnorthwest        0.031
regionsoutheast        0.020
regionsouthwest        0.055
age                    0.212
sexmale                0.109
bmi                    0.201
regionnorthwest:age    0.073
regionsoutheast:age    0.091
regionsouthwest:age    0.037


Analysis of Deviance Table (Type III tests)

Response: charges
            Df     F Pr(>F)  RESI  2.5% 97.5%
(Intercept)  1  6.18  0.013 0.062 0.000 0.117
region       3  0.73  0.537 0.000 0.000 0.096
age          1 30.62  0.000 0.149 0.091 0.210
sex          1  4.63  0.032 0.052 0.000 0.105
bmi          1 31.64  0.000 0.151 0.103 0.199
region:age   3  1.14  0.332 0.018 0.000 0.102

Overall RESI comparing model to intercept-only model:

  Res.Df Df      F Pr(>F)  RESI  2.5% 97.5%
1   1328  9 20.634      0 0.363 0.330 0.426
```

```
Notes:
1. The RESI was calculated using a robust covariance estimator.
2. Confidence intervals (CIs) constructed using 1000 non-parametric
bootstraps.
```

Here, we print the full output of the '`resi`' object. In addition to elements previously discussed, notes on the type of covariance estimator (robust or naive) and type and number of bootstraps are found at the bottom. As expected, we can see that the results differ slightly from our first '`resi`' object output.

2.5. RESI Summary

The **RESI** R package aims to provide estimates and confidence intervals for the recently introduced index in a way that intuitively complements common data analysis workflow in R. Similarly to running `summary()` after fitting a model, a user can simply run `resi()` on many models and obtain several useful model summaries that include original model estimates and p values as well as RESI estimates with confidence intervals. Reporting model parameter estimates with confidence intervals is useful for communicating estimated effects within a given context. Adding RESI estimates and their confidence intervals provides the extra benefit of communicating how strong or meaningful these effects may be in a way that can be compared across many model settings.

The package provides dedicated methods for several common model types currently, with more in process. Methods for both cross-sectional and longitudinal models are available, with longitudinal methods providing both a longitudinal and a per-measurement cross-sectional RESI estimate. For models that are not currently implemented, users can manually provide the relevant information to functions within **RESI** to obtain estimates directly. The package also makes it easy to visualize RESI estimates and convert to and from other effect size indices. The RESI is a widely applicable effect size index with several advantages, including the ability to accommodate nuisance parameters and incorporate robust covariance estimates. With increasing emphasis being placed on reporting of effect sizes in research, the **RESI** package is a user-friendly tool to easily report effect sizes and confidence intervals in publications.

2.6. Computational details

All examples were coded using R version 4.3.1 and **RESI** version 1.2.4 (R Core Team 2023; Jones *et al.* 2023b). The versions of relevant packages for the examples include **sandwich** 3.1-0 (Zeileis 2006), **sars** 1.3.6 (Matthews *et al.* 2019), **survival** 3.5-5 (Therneau 2023), **boot** 1.3-28.1 (Canty and Ripley 2022), and **ggplot2** 3.4.4 (Wickham 2016).

3. Effect Sizes and Individual Prediction in Neuroimaging

The RESI can be immediately applicable to the univariate neuroimaging setting. Consider t-statistic maps, which can be directly transformed into RESI maps. Thresholding can continue based on the magnitude of the RESI estimates instead of the p values associated with the t-statistic, as described in (Vandekar and Stephens 2021). Extensions to the multivariate setting may not be immediately apparent and require further discussion.

3.1. Multivariate Neuroimaging

In recent years, the neuroimaging literature has seen an increase in the use of multivariate methods to facilitate individual prediction (Sui *et al.* 2020; Scheinost *et al.* 2019; Jollans *et al.* 2019; Soares *et al.* 2016). In these studies, models are built to generate individual predictions of features such as cognition, diagnosis, or age. For example, some papers have demonstrated success predicting working memory using functional, network, and hub connectivity (Sui *et al.* 2020; Yamashita *et al.* 2018, 2015; Bertolero *et al.* 2018). The goal of this type of analysis is to build models that can yield reliable predictions for external, unseen data. This is in line with the trend in the broader medical world to move towards individualized or personalized medicine (Johnson *et al.* 2021). Patients and their providers may be less interested in a group-level average result than a numerical or categorical prediction based on their own data, which could include neuroimaging data and personal health and demographic features (Johnson *et al.* 2021; Bzdok and Ioannidis 2019). Multivariate methods are better suited to this goal than mass univariate analyses, which may not generalize as readily (Yeung *et al.* 2022; Bzdok and Ioannidis 2019).

While predictive models may be built using a small number of relevant features, it is often of interest to use as much of the imaging data as possible to generate predictions. Because of the high-dimensionality of imaging data, feature reduction techniques such as independent component analysis or considering regions of interest that summarize many voxels each are often employed, although modern flexible machine learning models make it feasible to use raw data from all voxels in an image as predictors (Soares *et al.* 2016). Methods that can effectively handle the high-dimensional data, where the number of features may far exceed the number of patients, include support vector machines and random forests (Jollans *et al.* 2019; Soares *et al.* 2016).

The process of a multivariate predictive analysis can be broken down into four steps: model building, internal validity, external validity, and generalizability and transposability (Bzdok and Ioannidis 2019). Each of these steps is important, but they are employed to varying degrees in neuroimaging studies. Model building includes deciding on which features to include and modeling approach to use for the data. Common choices for fMRI data include multiple linear regression, support vector machines (SVM), elastic net regression, and random forests (Yeung *et al.* 2022; Soares *et al.* 2016). Internal validity refers to assessing the model performance within the original data. This can be accomplished in several ways, including by splitting the data into a training and test set, employing K-fold or leave-one-out (LOO) cross-validation (CV), or cross-fitting. The key feature of internal validation is that it is using participants from the same dataset to compare predicted and actual outcomes. These individuals may be more similar in some ways than those coming from an external, independent dataset. As such, the predictive accuracy estimated using internal validation tends to be more optimistic than that of external validation (Bzdok and Ioannidis 2019). External validation can be more challenging to perform, as it requires the use of an independent dataset. This data could be collected from a separate study or at a later date from the data used to build and internally validate the model. Generalizability and transposability further expands on the external validity of a predictive model. Study data is often collected in a somewhat narrowly defined group of individuals, subject to exclusions based on age or other demographic or clinical factors. It is also important to consider how well a predictive model will generalize not just to an external dataset of individuals with similar characteristics to the initial dataset, but also to wider swaths of the population. Naturally, predictive accuracy in this context will

tend to be the worst of all the steps and the most challenging to capture (Bzdok and Ioannidis 2019). The increased availability of large neuroimaging datasets comprising many types of patients can make this more accessible. However, even external validation is often still not assessed in predictive neuroimaging studies (Yeung *et al.* 2022).

A recent review of 108 neuroimaging studies for individual trait prediction from 2007-2021 summarizes several interesting findings related to the steps discussed above (Yeung *et al.* 2022). They demonstrate that the number of studies in this field being published annually has seen a sharp increase, especially since 2017. However, use of an external dataset for validation was only present in 24% of the studies, and all of the studies using external validation were published in 2016 onward, without a significant trend in the proportion of studies using it in the years since. The authors found a negative correlation between sample size and prediction accuracy in internal validation, highlighting that smaller studies may result in overfit models with overly optimistic errors. However, they noted that this correlation was not observed when considering sample size and external validation accuracy, further arguing for the need of external validation. The pervasiveness of overly optimistic predictive models may contribute to the lack of replicability observed in neuroimaging (Marek *et al.* 2022).

Certainly, it is important to measure predictive accuracy, both internally and externally. However, there are many methods available for quantifying predictive performance, including correlation between predicted and actual values, mean square error, root mean square error, mean absolute error, and prediction R^2, for continuous target features (Scheinost *et al.* 2019). Yeung *et al.* (2022) found that Pearson correlation was the most commonly reported measure of predictive accuracy, in both internal (81/108, 75%) and external (16/26, 61.5%) validation. Both internal and external accuracy via Pearson correlation ranged widely across studies. Spearman rank correlation was the second most popular method.

The correlation between predicted and actual target features presents an interesting case for the discussion of effect sizes in neuroimaging. Note that correlation is an example of a standardized effect size. It is a unitless measure, ranging from -1 to 1, with its value signifying, in this context, the magnitude of predictive ability of a multivariate neuroimaging model. We typically take correlation in this setting to range from 0 to 1, as 0 indicates no trend between actual and predicted values, and we expect that, given our training data, our model will perform at least as well as completely random predictions. Values closer to 1 indicate better predictive ability. Because of its straightforward interpretation and its popular use in neuroimaging studies (Yeung *et al.* 2022), correlation is a natural starting point for discussion of effect sizes in multivariate neuroimaging.

We have discussed the desirability of computing confidence intervals around effect size estimates to quantify our uncertainty in these measures. In general, when computing the sample Pearson correlation coefficient from data, it is common to also compute a confidence interval using a Fisher Z-transformation (and possibly a corresponding p value). Unfortunately, the assumptions underlying the properties of typical sample Pearson correlation coefficient are often not met in the high-dimensional neuroimaging setting using flexible machine learning models, leading to inconsistent estimation, as will be discussed in Section 4. Thus, parametric or bootstrapped confidence intervals will not have the nominal coverage (Hines *et al.* 2022). In practice, it is not common to provide a confidence interval for Pearson correlation in multivariate neuroimaging studies (Yeung *et al.* 2022). It is still possible and common to compute a p value for the statistic using a permutation test. In the fMRI setting, this usually involves shuffling the target feature amongst the participants and repeating the ma-

chine learning pipeline (from data splitting to model building and evaluation) many times (e.g., 5000 or 10000) and reporting the p value as the proportion of permuted correlation estimates that were greater than or equal to the observed correlation. While this p value may be a useful piece of information, it is not as informative as a valid confidence interval would be. Therefore, it is worthwhile to investigate methods for constructing an estimator for correlation that would be consistent in the high-dimensional multivariate neuroimaging setting. The next subsection details the basics of some theory that may prove useful for this task. Section 4 will outline the initial application of this theory to our setting.

3.2. Applicable Semiparametric Theory

In order to construct a better estimator for correlation with a method for obtaining valid confidence intervals, we will use the theory of efficient influence functions and one-step bias-corrected estimators. This theory can be developed in the nonparametric setting, with typically straightforward applications in the semiparametric setting (Kennedy 2023). We summarize the theory as presented in Kennedy (2023) and Fisher and Kennedy (2019).

Notation

Consider a sample of independent and identically distributed observations $\{\mathbf{Z}_1, \mathbf{Z}_2, ...\mathbf{Z}_n\}$, where $\mathbf{Z}_i = (Y_i, X_{i1}, X_{i2}, ..., X_{im})$ and the true distribution of $\mathbf{Z}$ is denoted $\mathbb{P}$, with probability density function p. $\mathbb{P}$ lies in the model $\mathcal{P}$. Here, $\mathcal{P}$ can be considered as the nonparametric model which contains all probability distributions of the sample space.

We are interested in estimating some mapping of the model to a real value, rather than estimating the full distribution. The quantity we are interested in is referred to as a "functional" (or "target parameter" or "estimand") and is denoted as $\psi : \mathcal{P} \mapsto \mathbb{R}$. $\psi(\mathbb{P})$ is the value of the functional evaluated in the true model.

Submodels and Influence Function

Within the model $\mathcal{P}$, we can consider smooth parametric submodels denoted $\mathcal{P}_\epsilon = \{P_\epsilon : \epsilon \in \mathbb{R}\}$, where $P_0 = \mathbb{P}$, the true distribution.

Consider an arbitrary parametric distribution within $\mathcal{P}$, denoted $\tilde{P}$, with probability density function $\tilde{p}$. Let $\psi(\tilde{P})$ denote an estimate of $\psi(\mathbb{P})$ based on $\tilde{P}$.

We will define the distributions of the parametric submodel $\mathcal{P}_\epsilon$ according to the following probability density function:

$$p_\epsilon(z) = (1 - \epsilon)p(z) + \epsilon\tilde{p}(z), \epsilon \in [0, 1] \tag{6}$$

Therefore, we can consider the submodel $\mathcal{P}_\epsilon$ as a "path" connecting the parametric distribution $\tilde{P}$ ($P_{\epsilon=1}$) to the true distribution $\mathbb{P}$ ($P_{\epsilon=0}$). For each $\epsilon \in [0, 1]$, we can conceptualize a value for the functional $\psi(P_\epsilon)$, even though we will only compute an estimate at $\epsilon = 1$.

One way we can deconstruct our functional (under certain smoothness conditions) is with a von Mises expansion, which is essentially a distributional extension of a Taylor expansion. The von Mises expansion of $\psi(\mathbb{P})$ is

$$\psi(\mathbb{P}) = \psi(P_{\epsilon=1}) + \frac{d}{d\epsilon}\psi(P_\epsilon)|_{\epsilon=1}(0 - 1) - R_2(P_{\epsilon=1}, \mathbb{P}) \tag{7}$$

or equivalently,

$$\psi(\mathbb{P}) = \psi(P_{\epsilon=1}) - \int \varphi(z; P_{\epsilon=1}) d(P_{\epsilon=1} - \mathbb{P})(z) - R_2(P_{\epsilon=1}, \mathbb{P}) \tag{8}$$

where $\varphi(z; P)$ denotes the influence function of the target parameter for distribution P and R_2 denotes a second-order remainder, which depends on products or squares of differences between $P_{\epsilon=1}$ and $\mathbb{P}$. The influence function has a mean of zero and finite variance.

One-Step Estimator

From this expansion, one can see a route for constructing an estimator for the target parameter. If we use a plug-in estimator of the form $\psi(P_{\epsilon=1})$, we will have a first order bias of the form $-\int \varphi(z; P_{\epsilon=1}) d(P_{\epsilon=1} - \mathbb{P})(z)$. Note that this simplifies to $-\int \varphi(z; P_{\epsilon=1}) d\mathbb{P}(z)$ because as the influence function is a zero-mean function, $\int \varphi(z; P_{\epsilon=1}) dP_{\epsilon=1}(z) = 0$. This suggests constructing a modified estimator, commonly called a "one-step estimator," where we add in a bias-correction term as the sample average of the influence function. The one-step estimator takes the form

$$\hat{\psi} = \psi(P_{\epsilon=1}) + \mathbb{P}_n\{\varphi(Z; P_{\epsilon=1})\} \tag{9}$$

where $\mathbb{P}_n\{f\}$ denotes the sample average of a function. To assess the asymptotic behavior of this estimator, we can first decompose the difference between the estimator and the target parameter as

$$\begin{aligned} \hat{\psi} - \psi &= \psi(P_{\epsilon=1}) + \mathbb{P}_n\{\varphi(Z; P_{\epsilon=1})\} - \psi(\mathbb{P}) \\ &= (\mathbb{P}_n - \mathbb{P})\{\varphi(Z; \mathbb{P})\} + (\mathbb{P}_n - \mathbb{P})\{\varphi(Z; P_{\epsilon=1}) - \varphi(Z; \mathbb{P})\} + R_2(P_{\epsilon=1}, \mathbb{P}) \\ &\equiv S^* + T_1 + T_2 \end{aligned} \tag{10}$$

S^* is asymptotically normally distributed by the central limit theorem with variance $\frac{\text{var}(\varphi)}{n}$. For T_1 to converge at the rate of $\frac{1}{\sqrt{n}}$, we can rely on certain complexity conditions, or employ cross-fitting and assume consistency of the influence function. Cross-fitting involves randomly splitting the data into K folds and estimating ψ within each fold, taking the final estimator as the average of the estimates across the folds. The within-fold estimator is

$$\hat{\psi}_k = \psi(\hat{P}_{\epsilon=1,-k}) + \mathbb{P}_n^k\{\varphi(Z; \hat{P}_{\epsilon=1,-k})\} \tag{11}$$

where $\hat{P}_{\epsilon=1,-k}$ is the estimated distribution using the data from all but the k^{th} fold. $\psi(\hat{P}_{\epsilon=1,-k})$ is the plug-in estimate for the k^{th} fold using the predicted distribution from the other folds, and $\mathbb{P}_n^k\{\varphi(Z; \hat{P}_{\epsilon=1,-k})\}$ is the average of the influence function in the k^{th} fold using the predicted distribution from the other folds. We refer to Proposition 1 in (Kennedy 2023) to show that under cross-fitting with finite K and a consistent influence function, T_1 is consistent at the $\frac{1}{\sqrt{n}}$ rate.

Regarding the remaining term, T_2, which is equivalent to the second-order remainder $R_2(\hat{P}_{\epsilon=1}, \mathbb{P})$ in the influence-function-based estimator setting, the term depends on second-order products of differences between the estimated parametric distribution and the true distribution. To

achieve $\sqrt{n}$-convergence, each of the differences needs to converge at the rate of $\frac{1}{n^{\frac{1}{4}}}$ so the product is $\frac{1}{\sqrt{n}}$. This allows us to use more flexible models for $P_{\epsilon=1}$.

Under these conditions, the cross-fit influence-function-based estimator $\hat{\psi} = \sum_{k=1}^{K} (\frac{N_k}{n})\hat{\psi}_k$, where N_k is the number of observations in the k^{th} fold, is $\sqrt{n}$-consistent and asymptotically normal, with variance equal to $\text{var}\{\varphi(Z;\mathbb{P})\}$ (see Proposition 2, Kennedy (2023)). This variance is minimax optimal (Theorem 2, Kennedy (2023)) if φ is the efficient influence function. The efficient influence function of a parameter is the influence function that satisfies the von Mises expansion and is a valid score. For nonparametric models, there is only one influence function that satisfies the von Mises expansion.

This theory lays the groundwork for constructing bias-corrected estimators that are asymptotically normal while accommodating flexible modeling strategies, such as machine learning models. With this theory, we can construct estimators and valid 95% confidence intervals of the form $\hat{\psi} \pm 1.96\sqrt{\frac{\widehat{\text{var}}\{\hat{\varphi}(Z;\hat{P}_{\epsilon=1})\}}{n}}$. In the next subsection, we will detail some practical tools for finding the efficient influence function of a target parameter.

Finding the Influence Function

While the influence function can be most generally derived using the defintions of pathwise differentiability and explicitly solving for the influence function, there are methods to make the process more straightforward. In both of the following strategies, we first act as if the data are discrete. The resulting influence function can then be checked against the formal definitions to ensure its validity, but typically will hold. The process of deriving an influence function can be conceptualized using the Gateaux derivative. The Gateaux derivative of the target parameter ψ is written as

$$\frac{\partial}{\partial \epsilon}\psi\{(1-\epsilon)d\mathbb{P}(z) + \epsilon\delta_{z'}\}|_{\epsilon=0} \tag{12}$$

where $\delta_{z'}$ is an indicator function $\mathbb{I}(Z = z')$. Conceptually, this represents a slight contamination of the true distribution in the direction of z'. Computing the Gateaux derivative can proceed with use of typical derivative rules. This strategy is helpful for quickly computing the influence function of simple functionals, such as the marginal or conditional expectation of a random variable.

An extension of this strategy simplifies the process even further, but is based on the same concepts. In this strategy, we can view influence functions like derivatives and directly apply derivative rules to construct influence functions for more complicated parameters. This method usually leads to decomposing the influence function for a parameter into components where we can directly plug in the influence functions for simpler functional components.

For illustration, let $\mathbb{IF}(\psi)$ denote an operator for influence function of ψ. We can apply the chain rule as $\mathbb{IF}(f(\psi)) = f'(\psi)\mathbb{IF}(\psi)$ and the product rule as $\mathbb{IF}(\psi_1\psi_2) = \mathbb{IF}(\psi_1)\psi_2 + \psi_1\mathbb{IF}(\psi_2)$, for example.

The next section will provide an example of using this strategy to derive an influence function for a target parameter that can be useful in the multivariate neuroimaging context.

4. Current and Future Directions

As we shift our focus from the univariate setting to the multivariate neuroimaging setting, we plan to use the theory outlined above to improve effect size estimators in the predictive neuroimaging context. Our current project centers on deriving and evaluating an improved estimator for the correlation between predicted and actual phenotypes for flexible machine learning models using neuroimaging data as input.

4.1. Improved Correlation Estimator

For this discussion, we will consider a target feature, Y, which is a continuous variable, such as age or intelligence. $\mathbf{X}$ represents a set of predictive features, such as fMRI data and clinical or demographic covariates. We are interested in predicting Y from $\mathbf{X}$, where the number of features in $\mathbf{X}$ can be much greater than the number of individuals observed, n. We build a model for $\mu(\mathbf{X}) = \mathrm{E}[Y|\mathbf{X}]$ and use this model to generate predictions for Y for data that is independent of the data used to build the model.

As discussed above, the Pearson correlation coefficient is a common metric for measuring predictive accuracy for continuous target features (Yeung *et al.* 2022). The Pearson correlation between the actual target feature Y and the predicted feature $\mu(\mathbf{X})$ is defined as

$$\rho = \frac{\mathrm{cov}(\mu(\mathbf{X}), Y)}{\sqrt{\mathrm{var}(\mu(\mathbf{X}))\mathrm{var}(Y)}} \tag{13}$$

The typical estimator for the Pearson correlation coefficient for a sample of n observations is given as

$$\hat{\rho}_p = \frac{\sum_{i=1}^{n}(\hat{\mu}(\mathbf{X}_i) - \bar{\hat{\mu}}(\mathbf{X}))(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(\hat{\mu}(\mathbf{X}_i) - \bar{\hat{\mu}}(\mathbf{X}))^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{14}$$

where $\hat{\mu}(\mathbf{X})$ is the predicted value from the model and $\bar{\hat{\mu}}(\mathbf{X})$ is the sample average of the model predictions. While in many analysis settings, it is common practice to compute a confidence interval for Pearson correlation using a Fisher transformation (the default in the commonly used `cor.test()` function in R), this practice relies on the assumption that the variables follow a bivariate normal distribution. This type of confidence interval performs well for parametric predictive models, such a linear regression model with a low number of predictors. However, when using a more flexible model such as a random forest, the assumption is not met, and the Fisher confidence interval fails to provide nominal coverage.

In the absence of being able to use the Fisher confidence interval, we may think to turn to methods such as nonparametric bootstrapping. However, the consistency of the estimator $\hat{\rho}_p$ depends on the consistency of $\hat{\mu}(\mathbf{X})$. In general, for flexible nonparametric machine learning methods, we cannot rely on $\sqrt{n}$-consistency for $\hat{\mu}(\mathbf{X})$ (Fisher and Kennedy 2019). This means bootstrapping will not provide valid confidence intervals (Hines *et al.* 2022). We illustrate these ideas in Figure 4 using data from the Autism Brain Imaging Data Initiative (ABIDE) Preprocessed dataset (Craddock *et al.* 2013).

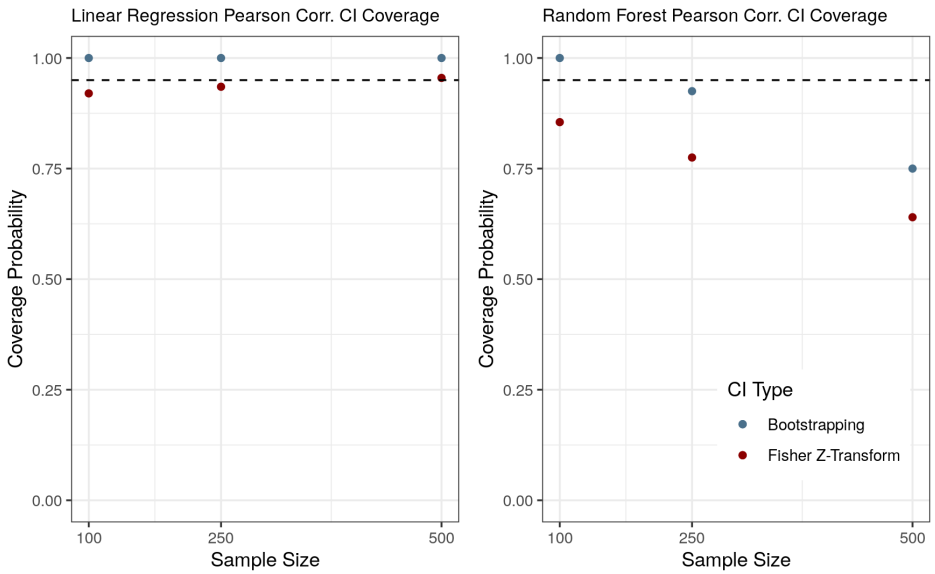

Figure 4: Comparison of Fisher Z-transformed and nonparametric bootstrapped 95% confidence interval coverage probability for Pearson correlation coefficient. fMRI data (fALFF) from the ABIDE Preprocessed dataset was used to predict age with either a linear regression model (left) with the mean voxel value and standard deviation voxel value as predictors or a random forest model (right) with the entire image as predictors. 200 simulations were run for each sample size, with 1000 bootstraps for the bootstrapped CIs.

This situation presents a good opportunity to apply the one-step estimation theory discussed above. We can conceptualize the Pearson correlation coefficient as our functional/target parameter and apply the simple derivative tricks to compute the corresponding influence function, allowing us to construct a bias-corrected estimator that can be employed for a wider range of flexible predictive models. We derive this estimator below.

The first step in our derivation is to simplify the target parameter based on the relationship between $\mu(\mathbf{X})$ and Y. We define the target parameter as $\psi = \rho$.

$$\begin{aligned}
\psi &= \frac{\text{cov}(\mu(\mathbf{X}), Y)}{\sqrt{\text{var}(\mu(\mathbf{X}))\text{var}(Y)}} \\
&= \frac{\mathsf{E}[\mu(\mathbf{X})Y] - \mathsf{E}[\mu(\mathbf{X})]\mathsf{E}[Y]}{\sqrt{\text{var}(\mu(\mathbf{X}))\text{var}(Y)}} \\
&= \frac{\mathsf{E}\{\mathsf{E}[\mu(\mathbf{X})Y|\mathbf{X}]\} - \mathsf{E}\{\mathsf{E}[Y|\mathbf{X}]\}\mathsf{E}[\mu(\mathbf{X})]}{\sqrt{\text{var}(\mu(\mathbf{X}))\text{var}(Y)}} \\
&= \frac{\mathsf{E}[\mu(\mathbf{X})^2] - \mathsf{E}[\mu(\mathbf{X})]^2}{\sqrt{\text{var}(\mu(\mathbf{X}))\text{var}(Y)}} \\
&= \frac{\sqrt{\text{var}(\mu(\mathbf{X}))}}{\sqrt{\text{var}(Y)}}
\end{aligned} \tag{15}$$

where the third equality is by iterated expectation and the fourth equality is by noting $\mu(\mathbf{X}) = \mathsf{E}[Y|\mathbf{X}]$. This simplification will help in the derivation of the influence function, and

also explicitly conveys that the target parameter is nonnegative. This formulation of the parameter suggests the initial plug in estimator of

$$\hat{\rho}_{pi} = \frac{\sqrt{\widehat{\text{var}}(\hat{\mu}(\mathbf{X}))}}{\sqrt{\widehat{\text{var}}(Y)}} \tag{16}$$

We also note a couple of common parameters and their influence functions that can be used as building blocks. The influence function for $\mathsf{E}[X]$ is $X - \mathsf{E}[X]$ and the influence function for $\text{var}(X)$ is $(X - \mathsf{E}[X])^2 - \text{var}(X)$ (Tsiatis 2007). To derive the influence function for ψ, we use the $\mathbb{IF}$ operator and apply derivative rules.

$$\begin{aligned}
\mathbb{IF}(\psi) &= \mathbb{IF}\Big(\Big(\frac{\text{var}(\mu(\mathbf{X}))}{\text{var}(Y)}\Big)^{\frac{1}{2}}\Big) \\
&= \frac{1}{2}\Big(\frac{\text{var}(\mu(\mathbf{X}))}{\text{var}(Y)}\Big)^{-\frac{1}{2}} \mathbb{IF}\Big(\frac{\text{var}(\mu(\mathbf{X}))}{\text{var}(Y)}\Big) \\
&= \frac{1}{2}\Big(\frac{\text{var}(\mu(\mathbf{X}))}{\text{var}(Y)}\Big)^{-\frac{1}{2}} \frac{\mathbb{IF}(\text{var}(\mu(\mathbf{X})))\text{var}(Y) - \mathbb{IF}(\text{var}(Y))\text{var}(\mu(\mathbf{X}))}{\text{var}(Y)^2} \\
&= \frac{1}{2}\Big(\frac{\text{var}(\mu(\mathbf{X}))}{\text{var}(Y)}\Big)^{-\frac{1}{2}} \frac{((\mu(\mathbf{X}) - \mathsf{E}[\mu(\mathbf{X})])^2 - \text{var}(\mu(\mathbf{X})))\text{var}(Y) - ((Y - \mathsf{E}[Y])^2 - \text{var}(Y))\text{var}(\mu(\mathbf{X}))}{\text{var}(Y)^2} \\
&= \frac{1}{2\sqrt{\text{var}(\mu(\mathbf{X}))}\text{var}(Y)^{\frac{3}{2}}}((\mu(\mathbf{X}) - \mathsf{E}[\mu(\mathbf{X})])^2\text{var}(Y) - \text{var}(\mu(\mathbf{X}))\text{var}(Y) \\
&\quad - (Y - \mathsf{E}[Y])^2\text{var}(\mu(\mathbf{X})) + \text{var}(Y)\text{var}(\mu(\mathbf{X})) \\
&= \frac{1}{2\sqrt{\text{var}(\mu(\mathbf{X}))}\text{var}(Y)^{\frac{3}{2}}}((\mu(\mathbf{X}) - \mathsf{E}[Y])^2\text{var}(Y) - (Y - \mathsf{E}[Y])^2\text{var}(\mu(\mathbf{X}))) \\
&= \frac{(\mu(\mathbf{X}) - \mathsf{E}[Y])^2}{2\sqrt{\text{var}(\mu(\mathbf{X}))\text{var}(Y)}} - \frac{(Y - \mathsf{E}[Y])^2\sqrt{\text{var}(\mu(\mathbf{X}))}}{2\text{var}(Y)^{\frac{3}{2}}}
\end{aligned} \tag{17}$$

To construct the corresponding one-step estimator, we add the sample average of the influence function above to the plug-in estimator to arrive at

$$\begin{aligned}
\hat{\psi} &= \frac{\sqrt{\widehat{\text{var}}(\hat{\mu}(\mathbf{X}))}}{\sqrt{\widehat{\text{var}}(Y)}} + \frac{1}{2n}\sum_{i=1}^{n}\Big\{\frac{(\hat{\mu}(\mathbf{X}_i) - \hat{\mathsf{E}}[Y])^2}{\sqrt{\widehat{\text{var}}(\hat{\mu}(\mathbf{X}))\widehat{\text{var}}(Y)}} - \frac{(Y_i - \hat{\mathsf{E}}[Y])^2\sqrt{\widehat{\text{var}}(\hat{\mu}(\mathbf{X}))}}{\widehat{\text{var}}(Y)^{\frac{3}{2}}}\Big\} \\
&= \frac{\sqrt{\widehat{\text{var}}(\hat{\mu}(\mathbf{X}))}}{\sqrt{\widehat{\text{var}}(Y)}} + \frac{\frac{1}{2n}\sum_{i=1}^{n}(\hat{\mu}(\mathbf{X}_i) - \hat{\mathsf{E}}[Y])^2}{\sqrt{\widehat{\text{var}}(\hat{\mu}(\mathbf{X}))\widehat{\text{var}}(Y)}} - \frac{\frac{1}{2n}\sum_{i=1}^{n}(Y_i - \hat{\mathsf{E}}[Y])^2\sqrt{\widehat{\text{var}}(\hat{\mu}(\mathbf{X}))}}{\widehat{\text{var}}(Y)^{\frac{3}{2}}} \\
&= \frac{\sqrt{\widehat{\text{var}}(\hat{\mu}(\mathbf{X}))}}{\sqrt{\widehat{\text{var}}(Y)}} + \frac{\frac{1}{2n}\sum_{i=1}^{n}(\hat{\mu}(\mathbf{X}_i) - \hat{\mathsf{E}}[Y])^2}{\sqrt{\widehat{\text{var}}(\hat{\mu}(\mathbf{X}))\widehat{\text{var}}(Y)}} - \frac{\sqrt{\widehat{\text{var}}(\hat{\mu}(\mathbf{X}))}}{2\sqrt{\widehat{\text{var}}(Y)}} \\
&= \frac{\sqrt{\widehat{\text{var}}(\hat{\mu}(\mathbf{X}))}}{2\sqrt{\widehat{\text{var}}(Y)}} + \frac{\frac{1}{2n}\sum_{i=1}^{n}(\hat{\mu}(\mathbf{X}_i) - \hat{\mathsf{E}}[Y])^2}{\sqrt{\widehat{\text{var}}(\hat{\mu}(\mathbf{X}))\widehat{\text{var}}(Y)}}
\end{aligned} \tag{18}$$

If we apply this estimator using cross-fitting, the theory suggests that we should be able to achieve asymptotic normality as long as $\hat{\mu}(\mathbf{X})$ converges in quadratic mean at the rate of $\frac{1}{n^{\frac{1}{4}}}$ and the variance of Y is finite. This will allow us to construct 95% confidence intervals as $\hat{\psi} \pm 1.96\sqrt{\frac{\widehat{\text{var}}(\hat{\mathbb{IF}}(\psi))}{n}}$. We are currently working on simulations to evaluate the performance of this estimator in the ABIDE Preprocessed dataset.

4.2. Other Extensions

One potential drawback of the confidence interval procedure mentioned above is the possibility of estimating confidence interval limits that lie outside of [0,1]. The use of the Fisher transformation for the typical Pearson correlation estimator circumvents this issue, returning estimates in the [-1,1] in situations in which the estimation and confidence interval procedure is valid. We plan to investigate the development of a similar transformation that could be applied to the one-step estimator and guarantee confidence intervals in the correct parameter space. Another approach we could explore is to use targeted maximum likelihood estimation (TMLE), which provides estimators that are asymptotically equivalent to the one-step estimator but can exhibit better finite-sample performance in the setting where we are dealing with a bounded parameter (Kennedy 2023).

Although we focused this section on the estimation of Pearson correlation as the effect size measure for measuring predictive performance, this process could be extended to other effect size measures. For example, we could use this asymptotically normal one-step estimator to construct a Z-statistic that could be converted to a RESI estimate. One-step estimators (and corresponding RESI estimates) could also be derived for other measures of predictive performance. Using RESI estimates in both the univariate and multivariate neuroimaging settings could facilitate greater use and communication about effect sizes across the neuroimaging literature. We can also use the theory of influence functions to work on other pressing problems in neuroimaging analysis, such as missing data imputation. Improvement in this area could in turn provide more precise effect size estimates.

5. Conclusion

Much like many areas of biostatistics, the neuroimaging field is calling for an increase in the reporting of standardized effect sizes (Bowring *et al.* 2019; Nichols *et al.* 2017; Chen *et al.* 2017; Reddan *et al.* 2017; Soares *et al.* 2016; Wasserstein and Lazar 2016). While there are challenges to widespread reporting of effect sizes in general, there are additional challenges applicable in the high-dimensional imaging setting. One major challenge has been the availability of software that allows users to easily obtain estimates and confidence intervals for a standardized effect size index (Chen *et al.* 2017). Our **RESI** R package and corresponding software paper helps to address this challenge and can be useful for many univariate neuroimaging problems. Computational challenges also arise in the multivariate setting, especially if bootstrapping is needed to obtain confidence intervals, as this may be too computationally expensive depending on the number of samples, features, and computational needs of the machine learning models employed. We are working to provide estimators for standardized effect sizes that can be employed in the multivariate setting without requiring infeasible computation times and that rely on solid semiparametric theory. We hope that our work can contribute meaningfully to this growing area of interest in the neuroimaging community.

6. Appendix

6.1. Additional RESI package examples

RESI on nonlinear least squares

In this example, we use `resi()` on a nonlinear least squares model using `nls()`, demonstrating a helpful workaround to deal with model convergence issues in '`nls`' models when bootstrapping. For this analysis, we use the `niering` dataset in the **sars** package, available on CRAN (Matthews *et al.* 2019). This dataset provides the area (in km^2) and number of plant species for 32 islands in the Kapingamarangi Atoll (Matthews *et al.* 2019).

```
R> data("niering", package = "sars")
R> head(niering)
```

```
        a  s
1 0.00012  5
2 0.00160  7
3 0.00240  8
4 0.00280 10
5 0.00360  9
6 0.00360 11
```

The species-to-area relationship is commonly modeled using a power curve, where $Species = cArea^z$ (Preston 1962). We can fit this model using `nls()` to estimate the c and z parameters. It is well known that '`nls`' models can be sensitive to the choice of starting values. For example, the following naive guesses for the starting values produce an error due to failed convergence.

```
R> mod_nls <- nls(s ~ c*a^z, data = niering, start = list(c = 2, z = 0.5))
```

```
Error in nls(s ~ c * a^z, data = niering, start = list(c = 2, z = 0.5)) :
singular gradient
```

If we use good starting values the model converges successfully.

```
R> mod_nls <- nls(s ~ c*a^z, data = niering, start = list(c = 3, z = 0.25))
R> summary(mod_nls)
```

```
Formula: s ~ c * a^z

Parameters:
  Estimate Std. Error t value Pr(>|t|)
c 89.30789   10.11148   8.832 7.59e-10 ***
z  0.40206    0.03677  10.935 5.49e-12 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.819 on 30 degrees of freedom

Number of iterations to convergence: 12
Achieved convergence tolerance: 2.792e-06
```

With our '`nls`' model, we can run `resi()`, making sure to provide the data argument. For this example, we will demonstrate the Bayesian bootstrap option.

```
R> set.seed(0827)
R> resi_obj_nls <- resi(mod_nls, data = niering, boot.method = "bayes")
R> resi_obj_nls


Analysis of Effect sizes (ANOES) based on RESI:
Confidence level =  0.05
Call:  nls(formula = s ~ c * a^z, data = niering, start = list(c = 3,
    z = 0.25), algorithm = "default", control = list(maxiter = 50,
    tol = 1e-05, minFactor = 0.0009765625, printEval = FALSE,
    warnOnly = FALSE, scaleOffset = 0, nDcentral = FALSE), trace = FALSE)

Coefficient Table
  Estimate Std. Error t value Pr(>|t|)   RESI   2.5%  97.5%
c  89.3079    20.5866  4.3382    1e-04 0.7475 0.6241 1.5651
z   0.4021     0.0597  6.7325    0e+00 1.1601 0.9599 2.5043

Overall RESI comparing model to intercept-only model:

              chi2 df P   RESI   2.5%   97.5%
Wald Test 142.1953  2 0 2.0931 1.2195  3.6873

Notes:
1. The RESI was calculated using a robust covariance estimator.
2. Credible intervals constructed using 1000 Bayesian bootstraps.
3. The bootstrap was successful in 744 out of 1000 attempts.
```

The `resi()` function runs without error, and we obtain a coefficients table and an overall Wald test for the model with RESI estimates and 95% credible intervals. Although the original model was able to be fit by `nls()` without issue, using this model for `resi()` does not have optimal performance. We can see from Note 3 that the bootstrap was only successful in 744 of the replicates.

The unsuccessful replicates failed to converge when attempting to update the '`nls`' model with bootstrap data. We can improve the performance of `resi()` for this model by refitting the '`nls`' model with different start values before running `resi()`. We use the estimated coefficients from the original model as the new start values.

```
R> mod_nls2 <- nls(s ~ c*a^z, data = niering,
+                  start = list(c = coef(mod_nlsn)[1],
+                  z = coef(mod_nlsn)[2]))
R> set.seed(0827)
R> resi(mod_nls2, data = niering, boot.method = "bayes")

Analysis of effect sizes based on RESI:
Confidence level =  0.05
Call:  nls(formula = s ~ c * a^z, data = niering,
    start = list(c = coef(mod_nls)[1], z = coef(mod_nls)[2]),
    algorithm = "default", control = list(maxiter = 50,
    tol = 1e-05, minFactor = 0.0009765625, printEval = FALSE,
    warnOnly = FALSE, scaleOffset = 0, nDcentral = FALSE), trace = FALSE)

Coefficient Table
    Estimate Std. Error t value Pr(>|t|)  RESI  2.5% 97.5%
c.c   89.308      20.59   4.338        0 0.748 0.619 1.941
z.z    0.402       0.06   6.732        0 1.160 0.950 2.506

Overall RESI comparing model to intercept-only model:

             chi2 df P  RESI  2.5%  97.5%
overall.tab 142.2  2 0 2.093 1.120  3.557

Notes:
1. The RESI was calculated using a robust covariance estimator.
2. Credible intervals constructed using 1000 Bayesian bootstraps.
3. The bootstrap was successful in 1000 out of 1000 attempts.
```

We see that running `resi()` on this model gives us the same RESI estimates and similar credible intervals, but the performance of the bootstrap is much better. In this case all 1000 bootstrap replicates are successful, and we obtain credible intervals based on the desired number of bootstrap replicates. When using `resi()` on an '`nls`' model, consider using this strategy if the model fails to converge in many of the bootstrap samples.

RESI on survival model

As a final example, we consider a parametric survival model using the survival package. Following an example in the survival package documentation, we fit a Weibull model using the `lung` dataset in the survival package (Therneau 2023). The outcome is survival time (in days). The regressors are age, sex, and Karnofsky score.

It is important to note that for survival models (using `coxph()` or `survreg()`), the option to use a robust covariance is included in the model fitting function. The `resi()` function ignores the `vcovfunc` argument for these model types and assumes the user has specified the desired covariance method when fitting the model.

In this example we also demonstrate how the user can obtain confidence intervals for different levels of α both during and after running the `resi()` function. The `alpha` arguments allows

the user to specify a vector of α levels, and the results corresponding to these levels will be output with the 'resi' object. In the case that the user wants to produce different level confidence intervals after running the `resi()` function without rerunning the bootstrapping process the user can set `store.boot = TRUE`. This will store a 'boot' object in the 'resi' object called `boot.results` that includes all of the RESI estimates for each bootstrap replicate. Confidence intervals of a specific α level can then be obtained via the **boot** package, manually, or by using the `summary()` or `anova()`/`car::Anova()` functions.

For this example we will use the `unbiased = FALSE` option to demonstrate the alternate Z to S estimator described in Equation 5. We also specify a reduced model to compute a RESI for a subset of the model parameters, rather than using an intercept-only model. Our reduced model uses Karnofsky score as the only predictor and we use 1500 bootstrap replicates to construct CIs.

```
R> library("survival")
R> set.seed(0827)
R> mod_surv <- survreg(Surv(time, status) ~ age + sex + ph.karno,
+                       data = survival::lung, dist="weibull",
+                       robust = TRUE)
R> mod_surv_reduced <- survreg(Surv(time, status) ~ ph.karno,
+                      data = survival::lung, dist="weibull",
+                      robust = TRUE)
R> resi_obj_surv <- resi(mod_surv, mod_surv_reduced, data = survival::lung,
+                    unbiased = FALSE, store.boot = TRUE,
+                    alpha = c(0.05, 0.1), nboot = 1500)
R> resi_obj_surv

Analysis of effect sizes based on RESI:
Confidence level =  0.05 0.1
Full Model:survreg(formula = Surv(time, status) ~ age + sex + ph.karno,
    data = survival::lung, dist = "weibull", robust = TRUE)
Reduced Model:survreg(formula = Surv(time, status) ~ ph.karno,
     data = survival::lung, dist = "weibull", robust = TRUE)


Coefficient Table
            Estimate Std. Error z value Pr(>|z|)   RESI   2.5%     5%    95%
(Intercept)    5.326      0.685   7.771    0.000  0.512  0.338  0.360  0.669
age           -0.009      0.007  -1.217    0.223 -0.046 -0.205 -0.184  0.000
sex            0.370      0.123   3.022    0.003  0.189  0.032  0.071  0.299
ph.karno       0.009      0.006   1.587    0.112  0.082  0.000  0.000  0.268
Log(scale)    -0.281      0.067  -4.164    0.000 -0.268 -0.443 -0.422 -0.171
              97.5%
(Intercept)    0.700
age            0.000
sex            0.318
ph.karno       0.306
Log(scale)    -0.152
```

```
Analysis of Deviance Table (Type II tests)

Response: Surv(time, status)
         Df Chisq Pr(>Chisq)  RESI  2.5%    5%   95% 97.5%
age       1  1.48      0.224 0.046 0.000 0.000 0.184 0.205
sex       1  9.13      0.003 0.189 0.032 0.071 0.299 0.318
ph.karno  1  2.52      0.112 0.082 0.000 0.000 0.268 0.306

Overall RESI comparing full model to reduced model:

  Res.Df Df Chisq Pr(>Chisq)  RESI  2.5%    5%   95% 97.5%
1    222  2 10.23      0.006 0.190 0.039 0.078 0.315 0.339

Notes:
1. The RESI was calculated using a robust covariance estimator.
2. Confidence intervals (CIs) constructed using 1500 non-parametric
bootstraps.
```

The printed output reflects the modifications we made to the `resi()` arguments. The reduced model formula is displayed, which is relevant only for the overall RESI estimate. For comparison, we can look at the `overall` element of running `resi()` with an intercept-only reduced model.

```
R> set.seed(0827)
R> omnibus(resi(mod_surv, data = survival::lung,
+    unbiased = FALSE, alpha = c(0.05, 0.1), nboot = 1500))

Analysis of effect sizes based on RESI:
Confidence level =  0.05 0.1
Wald test

Model 1: Surv(time, status) ~ 1
Model 2: Surv(time, status) ~ age + sex + ph.karno
  Res.Df Df Chisq Pr(>Chisq)  RESI  2.5%    5%   95% 97.5%
1    225
2    222  3  11.5      0.009 0.194 0.062 0.098 0.381 0.408
```

The overall RESI estimate is slightly higher when comparing to an intercept-only model than the model that adjusts for Karnofsky score.

The coefficient table and ANOVA table are computed only for the full model. Because we chose the unbiased option, the RESI estimates are equal in absolute value for the coefficient and ANOVA tables. The RESI estimates for age and Karnofsky (-0.046 (95% CI: -0.205, 0) and 0.082 (95% CI: 0, 0.306) respectively) are interpreted as small effects, while the RESI estimate for sex (0.189 (95% CI: 0.032, 0.318)) is interpreted as a small to moderate effect.

We see from the output that there are now four columns for the RESI confidence intervals – a lower and upper bound for each of the α levels specified. If we now want to obtain an interval with a different confidence level, we can run `summary()` and `anova()` using the `alpha` argument and specify a vector of values.

```
R> summary(resi_obj_surv, alpha = c(0.001, 0.01))

Analysis of effect sizes based on RESI:
Confidence level =  0.001 0.01
Call:  survreg(formula = Surv(time, status) ~ age + sex + ph.karno,
    data = survival::lung, dist = "weibull", robust = TRUE)

Coefficient Table
            Estimate Std. Error z value Pr(>|z|)   RESI  0.05%   0.5%  99.5%
(Intercept)    5.326      0.685   7.771    0.000  0.512  0.231  0.293  0.758
age           -0.009      0.007  -1.217    0.223 -0.046 -0.293 -0.247  0.055
sex            0.370      0.123   3.022    0.003  0.189  0.000  0.000  0.361
ph.karno       0.009      0.006   1.587    0.112  0.082 -0.072  0.000  0.383
Log(scale)    -0.281      0.067  -4.164    0.000 -0.268 -0.561 -0.517 -0.087
              99.95%
(Intercept)    0.807
age            0.105
sex            0.425
ph.karno       0.445
Log(scale)     0.000

R> anova(resi_obj_surv, alpha = c(0.001, 0.01))

          Df Chisq Pr(>Chisq)   RESI 0.05%   0.5% 97.5% 99.95%
age        1  1.48     0.2235 0.0461     0      0 0.247  0.293
sex        1  9.13     0.0025 0.1892     0      0 0.361  0.425
ph.karno   1  2.52     0.1124 0.0818     0      0 0.383  0.445
```

Note that if we try to specify different α levels with these functions on a '`resi`' object that did not use the `store.boot = TRUE` option, an error will occur with a message informing the user that this option was not used. A larger number of bootstrap samples are necessary to obtain adequate precision for smaller `alpha` levels.

References

Agresti A (2002). *Categorical Data Analysis*. Wiley Series in Probability and Statistics. John Wiley & Sons, Hoboken, NJ, USA. ISBN 978-0-471-36093-3 978-0-471-24968-9. doi:10.1002/0471249688. URL http://doi.wiley.com/10.1002/0471249688.

Althouse AD, Below JE, Claggett BL, Cox NJ, de Lemos JA, Deo RC, Duval S, Hachamovitch R, Kaul S, Keith SW, Secemsky E, Teixeira-Pinto A, Roger VL (2021). "Recommendations

for Statistical Reporting in Cardiovascular Medicine: A Special Report From the American Heart Association." *Circulation*, **144**(4), e70–e91. doi:10.1161/CIRCULATIONAHA.121.055393. URL https://www.ahajournals.org/doi/10.1161/CIRCULATIONAHA.121.055393.

Amaral EdOS, Line SRP (2021). "Current Use of Effect Size or Confidence Interval Analyses in Clinical and Biomedical Research." *Scientometrics*, **126**(11), 9133–9145. ISSN 0138-9130. doi:10.1007/s11192-021-04150-3.

American Psychological Association (1994). *Publication Manual of the American Psychological Association*. 4th ed. edition. American Psychological Association, Washington, DC. ISBN 1-55798-243-0 (Hardcover); 1-55798-241-4 (Paperback).

American Psychological Association (2001). *Publication Manual of the American Psychological Association*. 5th ed. edition. American Psychological Association, Washington, DC. ISBN 978-1-55798-791-4.

American Psychological Association (2010). *Publication Manual of the American Psychological Association*. 6th ed. edition. American Psychological Association, Washington, DC. ISBN 978-1-4338-0561-5.

American Psychological Association (2020). *Publication Manual of the American Psychological Association*. 7th ed. edition. American Psychological Association, Washington, DC. ISBN 978-1-4338-3215-4.

Anderson D (2020). ***esvis**: Visualization and Estimation of Effect Sizes*. R package version 0.3.1, URL https://CRAN.R-project.org/package=esvis.

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

Ben-Shachar MS, Lüdecke D, Makowski D (2020). "**effectsize**: Estimation of Effect Size Indices and Standardized Parameters." *Journal of Open Source Software*, **5**(56), 2815. doi:10.21105/joss.02815. URL https://doi.org/10.21105/joss.02815.

Bertolero MA, Yeo BT, Bassett DS, D'Esposito M (2018). "A Mechanistic Model of Connector Hubs, Modularity and Cognition." *Nature human behaviour*, **2**(10), 765–777. ISSN 2397-3374. doi:10.1038/s41562-018-0420-6. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6322416/.

Betensky RA (2019). "The p-Value Requires Context, Not a Threshold." *The American Statistician*, **73**(sup1), 115–117. ISSN 0003-1305. doi:10.1080/00031305.2018.1529624. URL https://doi.org/10.1080/00031305.2018.1529624.

Boos DD, Stefanski LA (2013). *Essential Statistical Inference: Theory and Methods*. Springer Texts in Statistics. Springer-Verlag, New York. ISBN 978-1-4614-4817-4. URL //www.springer.com/us/book/9781461448174.

Bowring A, Telschow F, Schwartzman A, Nichols TE (2019). "Spatial confidence sets for raw effect size images." *NeuroImage*, p. 116187. ISSN 1053-8119. doi:10.1016/j.neuroimage.2019.116187. URL http://www.sciencedirect.com/science/article/pii/S1053811919307785.

Buchanan EM, Gillenwaters A, Scofield JE, Valentine K (2019). ***MOTE**: Measure of the Effect: Package to Assist in Effect Size Calculations and Their Confidence Intervals*. R package version 1.0.2, URL http://github.com/doomlab/MOTE.

Bzdok D, Ioannidis JPA (2019). "Exploration, Inference, and Prediction in Neuroscience and Biomedicine." *Trends in Neurosciences*, **42**(4), 251–262. ISSN 0166-2236. doi:10.1016/j.tins.2019.02.001. URL https://www.sciencedirect.com/science/article/pii/S0166223619300074.

Canty A, Ripley BD (2022). ***boot**: Bootstrap R (S-PLUS) Functions*. R package version 1.3-28.1, URL https://cran.r-project.org/package=boot.

Carey VJ (2022). ***gee**: Generalized Estimation Equation Solver*. R package version 4.13-25, URL https://CRAN.R-project.org/package=gee.

Chen G, Taylor PA, Cox RW (2017). "Is the Statistic Value All We Should Care About in Neuroimaging?" *NeuroImage*, **147**, 952–959. ISSN 1095-9572. doi:10.1016/j.neuroimage.2016.09.066.

Cohen J (1988). *Statistical Power Analysis for the Behavioral Sciences*. Erlbaum Associates, Hillsdale, NJ. ISBN 978-0-203-77158-7. URL https://doi.org/10.4324/9780203771587.

Craddock C, Benhajali Y, Chu C, Chouinard F, Evans A, Jakab A, Khundrakpam BS, Lewis JD, Li Q, Milham M (2013). "The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives." *Neuroinformatics*, **41**.

Fisher A, Kennedy EH (2019). "Visually Communicating and Teaching Intuition for Influence Functions." doi:10.48550/arXiv.1810.03260. ArXiv:1810.03260 [math, stat], URL http://arxiv.org/abs/1810.03260.

Fox J, Weisberg S (2019). *An R Companion to Applied Regression*. Third edition. Sage, Thousand Oaks CA. URL https://socialsciences.mcmaster.ca/jfox/Books/Companion/.

Fritz CO, Morris PE, Richler JJ (2012). "Effect Size Estimates: Current Use, Calculations, and Interpretation." *Journal of Experimental Psychology: General*, **141**(1), 2–18. ISSN 1939-2222(Electronic),0096-3445(Print). doi:10.1037/a0024338.

Gerlanc D, Kirby K (2023). ***bootES**: Bootstrap Confidence Intervals on Effect Sizes*. R package version 1.3.0, URL https://CRAN.R-project.org/package=bootES.

Gonzalez-Castillo J, Saad ZS, Handwerker DA, Inati SJ, Brenowitz N, Bandettini PA (2012). "Whole-brain, time-locked activation with simple tasks revealed using massive averaging and model-free analysis." *Proceedings of the National Academy of Sciences of the United States of America*, **109**(14), 5487–5492. ISSN 0027-8424. doi:10.1073/pnas.1121049109. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3325687/.

Halekoh U, Højsgaard S, Yan J (2006). "The R Package **geepack** for Generalized Estimating Equations." *Journal of Statistical Software*, **15/2**, 1–11. URL https://cran.r-project.org/package=geepack.

Hedges LV, Olkin I (1985). *Statistical Methods for Meta-Analysis*. Elsevier, London, UK. ISBN 978-0-08-057065-5. doi:10.1016/C2009-0-03396-0. URL https://linkinghub.elsevier.com/retrieve/pii/C20090033960.

Hines O, Dukes O, Diaz-Ordaz K, Vansteelandt S (2022). "Demystifying Statistical Learning Based on Efficient Influence Functions." *The American Statistician*, **76**(3), 292–304. ISSN 0003-1305, 1537-2731. doi:10.1080/00031305.2021.2021984. ArXiv:2107.00681 [math, stat], URL http://arxiv.org/abs/2107.00681.

Jackman S (2020). ***pscl**: Classes and Methods for R Developed in the Political Science Computational Laboratory*. United States Studies Centre, University of Sydney, Sydney, New South Wales, Australia. R package version 1.5.5.1, URL https://github.com/atahk/pscl/.

Johnson KB, Wei W, Weeraratne D, Frisse ME, Misulis K, Rhee K, Zhao J, Snowdon JL (2021). "Precision Medicine, AI, and the Future of Personalized Health Care." *Clinical and Translational Science*, **14**(1), 86–93. ISSN 1752-8054. doi:10.1111/cts.12884. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7877825/.

Jollans L, Boyle R, Artiges E, Banaschewski T, Desrivières S, Grigis A, Martinot JL, Paus T, Smolka MN, Walter H, Schumann G, Garavan H, Whelan R (2019). "Quantifying Performance of Machine Learning Methods for Neuroimaging Data." *NeuroImage*, **199**, 351–365. ISSN 1095-9572. doi:10.1016/j.neuroimage.2019.05.082.

Jones M, Kang K, Vandekar S (2023a). "**RESI**: An R Package for Robust Effect Sizes." doi:10.48550/arXiv.2302.12345. ArXiv:2302.12345 [stat], URL http://arxiv.org/abs/2302.12345.

Jones M, Kang K, Vandekar S (2023b). ***RESI**: Robust Effect Size Index (RESI) Estimation*. URL https://CRAN.R-project.org/package=RESI.

Kang K, Jones MT, Armstrong K, Avery S, McHugo M, Heckers S, Vandekar S (2023). "Accurate Confidence and Bayesian Interval Estimation for Non-centrality Parameters and Effect Size Indices." *Psychometrika*. ISSN 1860-0980. doi:10.1007/s11336-022-09899-x.

Kelley K (2022). "**MBESS**: The **MBESS** R Package." URL https://CRAN.R-project.org/package=MBESS.

Kennedy EH (2023). "Semiparametric Doubly Robust Targeted Double Machine Learning: A Review." doi:10.48550/arXiv.2203.06469. ArXiv:2203.06469 [stat], URL http://arxiv.org/abs/2203.06469.

Lenth RV, Buerkner P, Herve M, Love J, Riebl H, Singmann H (2021). "**emmeans**: Estimated Marginal Means, aka Least-Squares Means." URL https://CRAN.R-project.org/package=emmeans.

Lesnoff, M, Lancelot, R (2012). ***aod**: Analysis of Overdispersed Data*. R package version 1.3.2, URL https://cran.r-project.org/package=aod.

Long JS, Ervin LH (2000). "Using Heteroscedasticity Consistent Standard Errors in the Linear Regression Model." *The American Statistician*, **54**(3), 217–224. doi:https://doi.org/10.2307/2685594.

Lüdecke D (2019). ***esc**: Effect Size Computation for Meta Analysis (Version 0.5.1)*. doi:10.5281/zenodo.1249218. URL https://CRAN.R-project.org/package=esc.

MacKinnon JG, White H (1985). "Some Heteroskedasticity-Consistent Covariance Matrix Estimators with Improved Finite Sample Properties." *Journal of econometrics*, **29**(3), 305–325. doi:https://doi.org/10.1016/0304-4076(85)90158-7.

Mangiafico SS (2023). ***rcompanion**: Functions to Support Extension Education Program Evaluation*. Rutgers Cooperative Extension, New Brunswick, New Jersey. Version 2.4.30, URL https://CRAN.R-project.org/package=rcompanion/.

Mantel N (1963). "Chi-Square Tests with One Degree of Freedom; Extensions of the Mantel-Haenszel Procedure." *Journal of the American Statistical Association*, **58**(303), 690–700. ISSN 0162-1459. doi:10.2307/2282717. URL https://www.jstor.org/stable/2282717.

Marek S, Tervo-Clemmens B, Calabro FJ, Montez DF, Kay BP, Hatoum AS, Donohue MR, Foran W, Miller RL, Hendrickson TJ, Malone SM, Kandala S, Feczko E, Miranda-Dominguez O, Graham AM, Earl EA, Perrone AJ, Cordova M, Doyle O, Moore LA, Conan GM, Uriarte J, Snider K, Lynch BJ, Wilgenbusch JC, Pengo T, Tam A, Chen J, Newbold DJ, Zheng A, Seider NA, Van AN, Metoki A, Chauvin RJ, Laumann TO, Greene DJ, Petersen SE, Garavan H, Thompson WK, Nichols TE, Yeo BTT, Barch DM, Luna B, Fair DA, Dosenbach NUF (2022). "Reproducible Brain-Wide Association Studies Require Thousands of Individuals." *Nature*, **603**(7902), 654–660. ISSN 1476-4687. doi:10.1038/s41586-022-04492-9. Publisher: Nature Publishing Group, URL https://www.nature.com/articles/s41586-022-04492-9.

Matloff N, Yancey R (2022). ***regtools**: Regression and Classification Tools*. R package version 1.7.0, URL https://CRAN.R-project.org/package=regtools.

Matthews T, Triantis K, Whittaker R, Guilhaumon F (2019). "**sars**: An R Package for Fitting, Evaluating and Comparing Species—Area Relationship Models." *Ecography*, **42**, 1446–1455. doi:10.1109/ACC.2005.1470374. URL https://cran.r-project.org/package=sars.

Navarro D (2015). *Learning Statistics with R: A Tutorial for Psychology Students and Other Beginners. (Version 0.6)*. University of New South Wales, Sydney, Australia. R package version 0.5.1, URL https://learningstatisticswithr.com.

Nichols TE, Das S, Eickhoff SB, Evans AC, Glatard T, Hanke M, Kriegeskorte N, Milham MP, Poldrack RA, Poline JB, Proal E, Thirion B, Van Essen DC, White T, Yeo BTT (2017). "Best Practices in Data Analysis and Sharing in Neuroimaging Using Mri." *Nature Neuroscience*, **20**(3), 299–303. ISSN 1546-1726. doi:10.1038/nn.4500.

Papachristodoulou A, Prajna S (2005). "A tutorial on sum of squares techniques for systems analysis." In *Proceedings of the 2005, American Control Conference, 2005.*, pp. 2686–2700. IEEE.

Pinheiro J, Bates D, R Core Team (2023). ***nlme**: Linear and Nonlinear Mixed Effects Models*. R package version 3.1-162, URL https://CRAN.R-project.org/package=nlme.

Preston FW (1962). "The Canonical Distribution of Commonness and Rarity: Part I." *Ecology*, **43**(2), 185. ISSN 00129658. doi:10.2307/1931976. URL http://www.jstor.org/stable/1931976?origin=crossref.

R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Pustejovsky J (2022). ***clubSandwich**: Cluster-Robust (Sandwich) Variance Estimators with Small-Sample Corrections*. R package version 0.5.8, URL https://CRAN.R-project.org/package=clubSandwich.

Reddan MC, Lindquist MA, Wager TD (2017). "Effect Size Estimation in Neuroimaging." *JAMA Psychiatry*, **74**(3), 207–208. ISSN 2168-622X. doi:10.1001/jamapsychiatry.2016.3356. URL https://doi.org/10.1001/jamapsychiatry.2016.3356.

Rosenthal R (1994). "Parametric Measures of Effect Size." *The Handbook of Research Synthesis*, **621**, 231–244.

Rozeboom WW (1960). "The Fallacy of the Null-Hypothesis Significance Test." *Psychological Bulletin*, **57**(5), 416–428. ISSN 1939-1455. doi:10.1037/h0042040. Place: US Publisher: American Psychological Association.

Rubin DB (1981). "The Bayesian Bootstrap." *The Annals of Statistics*, **9**(1), 130–134. ISSN 0090-5364, 2168-8966. doi:10.1214/aos/1176345338. URL https://projecteuclid.org/journals/annals-of-statistics/volume-9/issue-1/The-Bayesian-Bootstrap/10.1214/aos/1176345338.full.

SAS Institute Inc (2016). *SAS/STAT 14.2 User's Guide: High-Performance Procedures*. SAS Institute Inc., Cary, NC.

Scheinost D, Noble S, Horien C, Greene AS, Lake EM, Salehi M, Gao S, Shen X, O'Connor D, Barron DS, Yip SW, Rosenberg MD, Constable RT (2019). "Ten Simple Rules for Predictive Modeling of Individual Differences in Neuroimaging." *NeuroImage*, **193**, 35–45. ISSN 1095-9572. doi:10.1016/j.neuroimage.2019.02.057.

Serdar CC, Cihan M, Yücel D, Serdar MA (2021). "Sample Size, Power and Effect Size Revisited: Simplified and Practical Approaches in Pre-Clinical, Clinical and Laboratory Studies." *Biochemia Medica*, **31**(1), 010502. ISSN 1330-0962. doi:10.11613/BM.2021.010502. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7745163/.

Soares JM, Magalhães R, Moreira PS, Sousa A, Ganz E, Sampaio A, Alves V, Marques P, Sousa N (2016). "A Hitchhiker's Guide to Functional Magnetic Resonance Imaging." *Frontiers in Neuroscience*, **10**. ISSN 1662-453X. doi:10.3389/fnins.2016.00515. Publisher: Frontiers, URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2016.00515/full.

Sui J, Jiang R, Bustillo J, Calhoun V (2020). "Neuroimaging-Based Individualized Prediction of Cognition and Behavior for Mental Disorders and Health: Methods and Promises." *Biological psychiatry*, **88**(11), 818–828. ISSN 0006-3223. doi:10.1016/j.biopsych.2020.02.016. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7483317/.

Sullivan GM, Feinn R (2012). "Using Effect Size - or Why the P Value Is Not Enough." *Journal of Graduate Medical Education*, **4**(3), 279–282. ISSN 1949-8349. doi:10.4300/JGME-D-12-00156.1. URL https://doi.org/10.4300/JGME-D-12-00156.1.

Therneau TM (2023). *A Package for Survival Analysis in R*. R package version 3.5-5, URL https://CRAN.R-project.org/package=survival.

Torchiano M (2020). ***effsize**: Efficient Effect Size Computation*. doi:10.5281/zenodo.1480624. R package version 0.8.1, URL https://CRAN.R-project.org/package=effsize.

Tsiatis A (2007). *Semiparametric Theory and Missing Data*. Springer Science & Business Media. ISBN 978-0-387-37345-4. Google-Books-ID: xqZFi2EMB40C.

Vandekar S, Tao R, Blume J (2020). "A Robust Effect Size Index." *Psychometrika*, **85**(1), 232. doi:https://doi.org/10.1007/s11336-020-09698-2.

Vandekar SN, Stephens J (2021). "Improving the Replicability of Neuroimaging Findings by Thresholding Effect Sizes Instead of P-Values." *Human brain mapping*, **42**(8), 2393–2398. ISSN 1065-9471. doi:10.1002/hbm.25374. Place: Hoboken, USA Publisher: John Wiley & Sons, Inc.

Wasserstein RL, Lazar NA (2016). "The ASA's Statement on p-Values: Context, Process, and Purpose." *The American Statistician*, **70**(2), 129–133. doi:https://doi.org/10.1080/00031305.2016.1154108.

White H (1980). "A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity." *Econometrica: Journal of the Econometric Society*, pp. 817–838. doi:https://doi.org/10.2307/1912934.

Wickham H (2016). ***ggplot2**: Elegant Graphics for Data Analysis*. Springer-Verlag. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org.

Wickham H, Hesselberth J, Salmon M (2022a). "**pkgdown**: Make Static HTML Documentation for a Package." R package version 2.0.7, URL https://CRAN.R-project.org/package=pkgdown.

Wickham H, Hester J, Chang W, Bryan J (2022b). ***devtools**: Tools to Make Developing R Packages Easier*. R package version 2.4.5, URL https://CRAN.R-project.org/package=devtools.

Wilkinson L (1999). "Statistical Methods in Psychology Journals: Guidelines and Explanations." *American Psychologist*, **54**, 594–604. ISSN 1935-990X(Electronic),0003-066X(Print). doi:10.1037/0003-066X.54.8.594.

Woo CW, Krishnan A, Wager TD (2014). "Cluster-Extent Based Thresholding in Fmri Analyses: Pitfalls and Recommendations." *NeuroImage*, **91**, 412. doi:10.1016/j.neuroimage.2013.12.058. Publisher: NIH Public Access, URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4214144/.

Xue G, Chen C, Lu ZL, Dong Q (2010). "Brain Imaging Techniques and Their Applications in Decision-Making Research." *Xin li xue bao. Acta psychologica Sinica*, **42**(1), 120–137. ISSN 0439-755X. doi:10.3724/SP.J.1041.2010.00120. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2849100/.

Yamashita M, Kawato M, Imamizu H (2015). "Predicting Learning Plateau of Working Memory from Whole-Brain Intrinsic Network Connectivity Patterns." *Scientific Reports*, **5**, 7622. ISSN 2045-2322. doi:10.1038/srep07622. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5154600/.

Yamashita M, Yoshihara Y, Hashimoto R, Yahata N, Ichikawa N, Sakai Y, Yamada T, Matsukawa N, Okada G, Tanaka SC, Kasai K, Kato N, Okamoto Y, Seymour B, Takahashi H, Kawato M, Imamizu H (2018). "A Prediction Model of Working Memory Across Health and Psychiatric Disease Using Whole-Brain Functional Connectivity." *eLife*, **7**, e38844. ISSN 2050-084X. doi:10.7554/eLife.38844. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6324880/.

Yeung AWK, More S, Wu J, Eickhoff SB (2022). "Reporting Details of Neuroimaging Studies on Individual Traits Prediction: A Literature Survey." *NeuroImage*, **256**, 119275. ISSN 1095-9572. doi:10.1016/j.neuroimage.2022.119275.

Zeileis A (2006). "Object-Oriented Computation of Sandwich Estimators." *Journal of Statistical Software*, **16**, 1–16. ISSN 1548-7660. doi:10.18637/jss.v016.i09. URL https://doi.org/10.18637/jss.v016.i09.

Zeileis A, Hothorn T (2002). "Diagnostic Checking in Regression Relationships." *R News*, **2**(3), 7–10. URL https://CRAN.R-project.org/doc/Rnews/.

Zeileis A, Köll S, Graham N (2020). "Various Versatile Variances: An Object-Oriented Implementation of Clustered Covariances in R." *Journal of Statistical Software*, **95**(1), 1–36. doi:10.18637/jss.v095.i01.

Zhang Z, Schoeps N (1997). "On Robust Estimation of Effect Size Under Semiparametric Models." *Psychometrika*, **62**(2), 201–214. ISSN 1860-0980. doi:10.1007/BF02295275. URL https://doi.org/10.1007/BF02295275.