



## credsubs: Multiplicity-Adjusted Subset Identification

Patrick M. Schnell  
The Ohio State University

Mark Fiecas  
University of Minnesota

Bradley P. Carlin  
Counterpoint Statistical  
Consulting, LLC

---

### Abstract

Subset identification methods are used to select the subset of a covariate space over which the conditional distribution of a response has certain properties – for example, identifying types of patients whose conditional treatment effect is positive. An often critical requirement of subset identification methods is multiplicity control, by which the family-wise Type I error rate is controlled, rather than the Type I error rate of each covariate-determined hypothesis separately. The credible subset (or credible subgroup) method provides a multiplicity-controlled estimate of the target subset in the form of two bounding subsets: one which entirely contains the target subset, and one which is entirely contained by it.

We introduce a new R package, **credsubs**, which constructs credible subset estimates using a sample from the joint posterior distribution of any regression model, a description of the covariate space, and a function mapping the parameters and covariates to the subset criterion. We demonstrate parametric and nonparametric applications of the package to a clinical trial dataset and a neuroimaging dataset, respectively.

*Keywords:* credible subgroups, multiple hypothesis testing, R, subset identification, subgroup analysis.

---

## 1. Introduction

Subset identification, the selection of the subset of an index set for which random variables have certain properties, can be applied to a wide variety of topics, but is especially useful in medicine and public health. In this context, the goal is usually to identify the types of patients, identified by their baseline covariate profiles, for whom a treatment has a positive effect. Given an index set  $X$  and an unknown, possibly parametric, real-valued function  $b(\mathbf{x}; \boldsymbol{\theta})$ , the large class of subset identification problems considered in this paper can be formulated as identifying

the subset  $B(\boldsymbol{\theta})$ , comprised of the  $\boldsymbol{x} \in X$  for which  $b(\boldsymbol{x}; \boldsymbol{\theta}) > \delta$ . In one example presented here, an analysis of clinical trials of Alzheimer’s disease treatments,  $b(\boldsymbol{x}; \boldsymbol{\theta})$  is a personalized (or conditional average) treatment effect, i.e., the difference in expected outcomes between subjects with identical baseline covariate profiles  $\boldsymbol{x}$ ,  $E[Y|\boldsymbol{x}, t = 1; \boldsymbol{\theta}] - E[Y|\boldsymbol{x}, t = 0; \boldsymbol{\theta}]$ . The target subset  $B(\boldsymbol{\theta})$  is then the set of covariate profiles that represent patients with positive personalized treatment effects, i.e., the benefiting subgroup. In a second example,  $\boldsymbol{x}$  represents coordinates of a voxel in a 2-dimensional slice of an fMRI scan of the brain,  $b(\boldsymbol{x}; \boldsymbol{\theta})$  is the blood oxygen level response to a stimulus, and  $B(\boldsymbol{\theta})$  is the set of voxels excited by that stimulus.

The subset identification problems considered here can be seen as collections of tests of the hypotheses  $H_{\boldsymbol{x}} = \{\boldsymbol{\theta} : b(\boldsymbol{x}; \boldsymbol{\theta}) \leq \delta\}$ . The subset  $B(\boldsymbol{\theta})$  is then the  $\boldsymbol{x}$  for which the  $H_{\boldsymbol{x}}$  are false, and a natural estimate  $\hat{B}$  is the set of  $\boldsymbol{x}$  for which the hypotheses are rejected by some test. However, in many cases the outcome of the family of tests as a whole is of interest. In particular, it may be desirable to control the probability that at least one member of  $\hat{B}$  is not truly in  $B$ , or that at least one member of  $B$  is not placed in  $\hat{B}$ . This article introduces the R (R Core Team 2020) package **credsubs** (Schnell and Carlin 2020), available from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/package=credsubs>, based on the “credible subgroups” methods developed by Schnell, Tang, Offen, and Carlin (2016), Schnell, Tang, Müller, and Carlin (2017), and Schnell, Müller, Tang, and Carlin (2018). Given a sample from the joint posterior distribution of  $b(\boldsymbol{x}; \boldsymbol{\theta})$  or the information necessary to construct one, the method constructs an estimator  $(D, S)$  with the property that  $P[D \subseteq B(\boldsymbol{\theta}) \subseteq S | \text{data}] \geq 1 - \alpha$ , thereby controlling multiplicity in both dimensions mentioned above. In most common cases, the estimator also has the frequentist property that  $P[D \subseteq B(\boldsymbol{\theta}_0) \subseteq S | \boldsymbol{\theta}_0] \geq 1 - \alpha$ .

The original credible subgroups methodology focused on identification of patient populations who benefit from treatment. This subgroup identification problem is one among many related problems under the umbrella of subgroup analysis, and credible subgroups is one among many possible approaches to addressing it. The broad problem of subset identification presented here is a generalization of subgroup identification as it appears in medicine. Broadly, subgroup analysis in medicine is concerned with treatment effects that may be heterogeneous among different subsets of the patient population, and includes problems such as detecting such heterogeneity, estimating treatment effects in subgroups, identifying subgroups that are anomalous in some way, identifying subgroups with positive average treatment effects, and identifying baseline covariate profiles that correspond to positive personalized treatment effects. It is especially important for our purposes to distinguish between the final two problems. Suppose that for the purpose of prescribing treatment for a particular condition, patients are characterized by their sex and a particular biomarker status. If biomarker-positive male patients benefit from treatment, and there is no effect for other patients, then the answer to the former problem is that three subgroups of the population have positive average treatment effects: male patients, biomarker-positive patients, and biomarker-positive male patients. However, the answer to the latter problem is that the only baseline covariate profile corresponding to a positive personalized treatment effect is that of biomarker-positive male patients.

As described in Schnell *et al.* (2016), there exist many approaches to benefiting subgroup identification. These include direct Bayesian inference from linear models with skeptical priors on interaction effects (Simon 2002), and tree-based or “flowchart-style” approaches such as classification and regression trees (Breiman, Friedman, Stone, and Olshen 1984; Chipman, George, and McCulloch 1998). A recent survey of variations on these approaches is given in

Zhang, Seibold, Vettore, Song, and François (2018), along with examples in R. In principle, the credible subsets approach may be applied on top of any modeling technique from which a sample from the joint posterior of  $b(\mathbf{x}; \boldsymbol{\theta})$  can be obtained.

The scope of this package includes only the construction of the estimator given properly formatted regression output. It does not include functionality for fitting any regression models. Packages that provide output in forms that facilitate use of this package include those which can directly produce a sample from the joint posterior of the regression surface, such as the **BayesTree** (Chipman and McCulloch 2016) package implementing Bayesian additive regression trees (BART; Chipman, George, and McCulloch 2010), and interfaces to hierarchical model samplers such as **R2OpenBUGS** (Sturtz, Ligges, and Gelman 2005), **rjags** (Plummer 2019), **nimble** (de Valpine *et al.* 2019; de Valpine, Turek, Paciorek, Anderson-Bergman, Temple Lang, and Bodik 2017), **INLA** (Martins, Simpson, Lindgren, and Rue 2013), and **rstan** (Carpenter *et al.* 2017; Stan Development Team 2020), which give samples from the joint posterior of the parameters.

The rest of this article proceeds as follows. Section 2 gives a brief overview of the credible subsets estimator’s definition and construction. Section 3 illustrates the basic usage of the package for a parametric example analysis of an Alzheimer’s disease clinical trial dataset. Section 4 presents a nonparametric example analysis of a neuroimaging dataset. Finally, Section 5 concludes with a discussion.

## 2. Credible subsets

### 2.1. Definition

Given a covariate space  $X$  consisting of points  $\mathbf{x}$ , and a real-valued function  $b(\mathbf{x}; \boldsymbol{\theta})$  possibly parameterized by  $\boldsymbol{\theta}$ , the credible subsets method is concerned with estimating the set  $B(\boldsymbol{\theta}) \equiv \{\mathbf{x} : b(\mathbf{x}; \boldsymbol{\theta}) > \delta\}$ , i.e., the subset of the covariate space for which  $b$  exceeds some threshold. The method constructs a credible subset pair  $(D, S)$  with the aim that  $D \subseteq B(\boldsymbol{\theta}) \subseteq S$ . The set  $D$  is termed the exclusive credible subset and is intended to contain only covariate points for which  $b(\mathbf{x}; \boldsymbol{\theta}) > \delta$ , and the set  $S$  is termed the inclusive credible subset and is intended to contain every covariate point for which  $b(\mathbf{x}; \boldsymbol{\theta}) > \delta$ .

The credible subset pair is constructed from a sample from the joint posterior distribution of  $b(\mathbf{x}; \boldsymbol{\theta})$ , written compactly as  $\mathbf{b}(X; \boldsymbol{\theta})$ . The pair may be interpreted in the Bayesian sense that  $P(D \subseteq B(\boldsymbol{\theta}) \subseteq S | \text{data}) \geq 1 - \alpha$ . However, under the construction of  $(D, S)$  from simultaneous confidence or credible bands detailed below, pairs constructed from a posterior sample often have asymptotic frequentist interpretations.

In most applications,  $b(\mathbf{x}; \boldsymbol{\theta})$  is estimated via regression of a response  $Y$  onto the covariate space. In the original credible subgroups usage,  $b(\mathbf{x}; \boldsymbol{\theta}) \equiv E[Y | \mathbf{x}, t = 1; \boldsymbol{\theta}] - E[Y | \mathbf{x}, t = 0; \boldsymbol{\theta}]$ , i.e., the conditional treatment effect ( $t = 1$  versus  $t = 0$ ). Thus, setting  $\delta = 0$ ,  $B(\boldsymbol{\theta})$  is the subgroup of patient types having positive conditional treatment effects.

### 2.2. Construction

In this section we present the construction of credible subsets in the Bayesian framework. We suppress dependence on both  $\boldsymbol{\theta}$  and  $Y$  in our notation, and unless otherwise stated, all distributions are posterior.

---

**Algorithm 1** Step-down procedure.

---

- 1 Let  $M = 1$ ,  $T_0 = X$ , and  $R_0 = \emptyset$  be the starting iteration, base test set, and base rejection set, respectively;
  - 2 **repeat**
  - 3   Let  $T_M = T_{M-1} \setminus R_{M-1} = X \setminus (\bigcup_{m < M} R_m)$  be the new test set;
  - 4   Construct a two-sided  $1 - \alpha$  restricted covariate space simultaneous confidence band (1) or (3) for  $b(\mathbf{x})$  over all  $\mathbf{x} \in T_M$ ;
  - 5   Let  $R_M$  be the set of  $\mathbf{x}$  for which the band does not contain  $\delta$ ;
  - 6   Increment  $M$ ;
  - 7 **until**  $R_M = \emptyset$ ;
  - 8 Reject  $H_{\mathbf{x}}$  for all  $\mathbf{x} \in \bigcup_{m < M} R_m$ .
- 

Credible subset pairs are constructed primarily through simultaneous credible bands, i.e, upper and lower bound surfaces  $\mathbf{u}(X)$  and  $\mathbf{l}(X)$  constructed so that the entirety of the regression surface  $\mathbf{b}(X)$  lies between them with posterior probability  $1 - \alpha$ . If the marginal distributions of the  $\mathbf{b}(\mathbf{x})$  are bell-shaped (e.g.,  $\mathbf{b}(X)$  is multivariate normal or multivariate  $t$ ), then a familiar form generalized from Uusipaikka (1983) may be used to construct the band:

$$[l(\mathbf{x}), u(\mathbf{x})] = \bar{b}(\mathbf{x}) \pm W_{\alpha, X}^* \sqrt{\text{VAR}[b(\mathbf{x})]}, \quad (1)$$

where  $\bar{b}(\mathbf{x})$  and  $\text{VAR}[b(\mathbf{x})]$  are the posterior mean and variance, respectively, of  $b(\mathbf{x})$  at each  $\mathbf{x}$ , and  $W_{\alpha, X}^*$  is the  $1 - \alpha$  quantile of

$$W_X = \sup_{\mathbf{x} \in X} \frac{|b(\mathbf{x}) - \bar{b}(\mathbf{x})|}{\sqrt{\text{VAR}[b(\mathbf{x})]}}. \quad (2)$$

In cases where the marginals of the  $b(\mathbf{x})$  are not so well-behaved (for example, if they are discrete or multimodal) a quantile-based band may be used instead. For a fixed  $\mathbf{x}$  temporarily suppressed in notation, let  $F_B(b) = \mathbb{P}[B \leq b]$  be the cumulative distribution function of  $b$ ,  $G_b(b) = \mathbb{P}[B < b]$  be its left-continuous counterpart, and  $F_B^{-1}(p) = \inf\{b : p \leq F_B(b)\}$ ,  $G_B^{-1}(p) = \sup\{b : p \geq G_B(b)\}$  be their inverses. We may then use the simultaneous credible band given by Schnell *et al.* (2018):

$$[l(\mathbf{x}), u(\mathbf{x})] = \left[ F_{b(\mathbf{x})}^{-1}\left(1 - W_{\alpha, X}^*\right), G_{b(\mathbf{x})}^{-1}\left(W_{\alpha, X}^*\right) \right], \quad (3)$$

where  $W_{\alpha, X}^*$  is the  $1 - \alpha$  quantile of

$$W_X = \sup_{\mathbf{x} \in X} \max \left\{ 1 - F_{b(\mathbf{x})}[b(\mathbf{x})], G_{b(\mathbf{x})}[b(\mathbf{x})] \right\}. \quad (4)$$

In either case, the distribution and quantile function of  $W_X$  may be estimated from the posterior sample of  $\mathbf{b}(X)$ . Credible subsets may be constructed as  $D = \{\mathbf{x} : l(\mathbf{x}) > \delta\}$  and  $S = \{\mathbf{x} : u(\mathbf{x}) \geq \delta\}$ . Under mild regularity conditions (Gelman, Carlin, Stern, Dunson, Vehtari,

and Rubin 2013; Carlin and Louis 2009), the posterior of  $\mathbf{b}(\mathbf{X})$  corresponds asymptotically to the sampling distribution of its maximum likelihood estimate, thus the Bayesian simultaneous credible bands and credible subset pairs are interpretable in the frequentist sense.

When interpreting credible subset pairs in the frequentist sense, the estimate may be improved by using a step-down testing procedure similar to the Holm-Bonferroni procedure (Holm 1979). For a given covariate vector  $\mathbf{x}$ , let  $H_{\mathbf{x}} = \{\boldsymbol{\theta} : b(\mathbf{x}) = \delta\}$  be the null hypothesis that  $b(\mathbf{x}) = \delta$ . Then Algorithm 1 controls the overall Type I error rate at  $\alpha$ . In words, the step-down procedure reconstructs the simultaneous credible band on the smaller subset of  $\mathbf{X}$  for which previous iterations did not reject  $H_{\mathbf{x}}$ . The proof of validity for this procedure relies on showing that it is a closed testing procedure (Marcus, Peritz, and Ruben 1976), in part by noting that  $W_{\alpha, V}^* \leq W_{\alpha, U}^*$  for  $V \subset U$ . If  $H_{\mathbf{x}}$  is rejected, we may place  $\mathbf{x}$  in  $D$  when the lower credible bound at  $\mathbf{x}$  was greater than  $\delta$ , and  $S^c$  (the complement of  $S$ ) when the upper bound was less than  $\delta$ . If  $H_{\mathbf{x}}$  is not rejected, we leave  $\mathbf{x}$  in  $S \setminus D$ .

### 3. Package

This section describes the use of the package’s three main functions:

- `sim.cred.band`, which constructs simultaneous credible bands according to (1) or (3);
- `credsubs`, which determines the membership of each covariate point in the exclusive and inclusive credible subsets using the step-down procedure described by Algorithm 1;
- `credsubs.level`, which determines the maximum credible level at which each covariate point is a member of either  $D$  or  $S^c$ .

It is important to note that this package provides functionality only for constructing credible subsets given a sample from the posterior of  $\mathbf{b}(\mathbf{X}; \boldsymbol{\theta})$ , or of the parameters necessary to construct them. The package provides no functionality for producing a regression fit in the first place, which is intended to be handled by software of the user’s choice, as long as it can provide the input necessary for the `credsubs` package. In this section, we analyze a dataset from four clinical trials of Alzheimer’s disease treatments, using `nimble` as the modeling package.

The clinical trials of Alzheimer’s disease treatments were carried out by AbbVie, Inc., and originally analyzed in the context of credible subsets in Schnell *et al.* (2018). The experimental treatment arms have been excluded, leaving the placebo ( $T = 0$ ) and standard of care ( $T = 1$ ) arms. Although the standard of care is palliative rather than disease-modifying, it is expected to alleviate symptoms, including worsening of the cognitive endpoint used in the trials. The primary endpoint  $Y$  is improvement in cognitive function from baseline, and the covariates making up  $\mathbf{x}$  are sex, carrier status of a genetic biomarker, baseline severity, and rate of cognitive decline since diagnosis. The quantity of interest for defining the benefiting population is the personalized treatment effect (PTE),  $b(\mathbf{x}; \boldsymbol{\theta}) \equiv \mathbb{E}[Y|\mathbf{x}, t = 1; \boldsymbol{\theta}] - \mathbb{E}[Y|\mathbf{x}, t = 0; \boldsymbol{\theta}]$ , not to be confused with the improvement of individual subjects assigned to the treatment arm.

#### 3.1. Formatting input

Each of the three main functions relies on the same three primary input objects:

- `params`, a matrix whose rows are draws from the posterior of  $\mathbf{b}(X; \boldsymbol{\theta})$ , or of  $\boldsymbol{\theta}$ ;
- `design`, a matrix whose rows represent points in the covariate space  $X$ ;
- `FUN`, a function which takes as arguments a row of `design` and the entirety of `params` and returns the corresponding sample from the posterior of  $b(\mathbf{x}; \boldsymbol{\theta})$ .

By default, `design` is set to `NULL`, in which case `params` is taken to be a sample from the posterior of  $\mathbf{b}(X; \boldsymbol{\theta})$  and `FUN` is ignored. If `design` is not `NULL`, `params` is taken to be a sample from the posterior of  $\boldsymbol{\theta}$ . Also by default, `FUN` is set to `function(x, params) { params %*% t(x) }`, so that a linear model is assumed whenever `design` is provided and `FUN` not otherwise specified. The outputs of each function are aligned with the rows of `design` if not `NULL`, and with the columns of `params` otherwise.

Note that `params` and `design` are matrices rather than data frames, and will be used in computations as such. Each function will attempt to coerce objects supplied to the `params` and `design` arguments to matrices via `data.matrix()`. In particular, vectors will be interpreted as matrices with a single column, and factors will be replaced by their numerical codes (so “dummy variable” or other coding schemes must be applied manually).

To begin our example, we load the data and reform it to be used by `nimble`. The `nimble` R package allows models to be specified in a manner similar to that of the BUGS language (Lunn, Spiegelhalter, Thomas, and Best 2009), can automatically configure an MCMC sampler, and compile to C. We take use of the BUGS model specification here for clarity, and in Section 4 take advantage of another feature: new datasets can be associated with the compiled model and sampler without recompilation.

```
R> library("nimble")
R> library("credsubs")
R> set.seed(1)
R> data("alzheimers", package = "credsubs")
R> head(alzheimers)
```

	Treatment	Severity	Decline	Sex	Carrier	Improvement
1	Placebo	22.0000	3.9834	F	CARRIER	-3.6667
2	Standard of Care	34.6667	2.4624	M	NON-CARRIER	14.0000
3	Standard of Care	27.6667	1.0639	F	NON-CARRIER	12.3333
4	Placebo	34.0000	4.8407	M	CARRIER	5.0000
5	Placebo	34.3333	4.2348	F	NON-CARRIER	5.6667
6	Standard of Care	28.6667	2.9314	M	NON-CARRIER	-1.0000

```
R> alzheimers$Treatment <- as.numeric(alzheimers$Treatment) - 1
R> alzheimers$Sex <- as.numeric(alzheimers$Sex) - 1
R> alzheimers$Carrier <- as.numeric(alzheimers$Carrier) - 1
R> alzheimers$Severity <- scale(alzheimers$Severity)
R> alzheimers$Decline <- scale(alzheimers$Decline)
```

We then specify the Bayesian normal linear model:

```

R> nimble.code <- nimbleCode({
+   for (i in 1:N) {
+     mean[i] <- beta[1] +
+       beta[2] * Severity[i] + beta[3] * Decline[i] +
+       beta[4] * Sex[i] + beta[5] * Carrier[i] +
+       beta[6] * Sex[i] * Carrier[i] +
+       gamma[1] * Treatment[i] +
+       gamma[2] * Severity[i] * Treatment[i] +
+       gamma[3] * Decline[i] * Treatment[i] +
+       gamma[4] * Sex[i] * Treatment[i] +
+       gamma[5] * Carrier[i] * Treatment[i] +
+       gamma[6] * Sex[i] * Carrier[i] * Treatment[i]
+
+     Improvement[i] ~ dnorm(mean[i], tau)
+   }
+
+   for (j in 1:6) {
+     beta[j] ~ dnorm(0, 10E-4)
+     gamma[j] ~ dnorm(0, ifelse(j == 1, 10E-4, 1))
+   }
+
+   tau ~ dgamma(10E-4, 10E-4)
+ })

```

Here `beta` and `gamma` are the regression parameters, of which we will be primarily concerned with `gamma` in making inferences about the PTEs. The parameter `tau` is the inverse of the usual error variance  $\sigma^2$ . The following code compiles the model and runs a MCMC sampler for 11,000 iterations.

```

R> nimble.constants <- list(N = nrow(alzheimers))
R> nimble.data <- list(Treatment = as.numeric(alzheimers$Treatment),
+   Severity = as.vector(alzheimers$Severity),
+   Decline = as.vector(alzheimers$Decline),
+   Sex = as.numeric(alzheimers$Sex),
+   Carrier = as.numeric(alzheimers$Carrier),
+   Improvement = as.vector(alzheimers$Improvement))
R> nimble.inits <- list(beta = rep(0, 6), gamma = rep(0, 6), tau = 1)
R> nimble.model <- nimbleModel(code = nimble.code, name = "example",
+   constants = nimble.constants, data = nimble.data, inits = nimble.inits)
R> nimble.spec <- configureMCMC(nimble.model)
R> nimble.mcmc <- buildMCMC(nimble.spec)
R> C.model <- compileNimble(nimble.model)
R> C.mcmc <- compileNimble(nimble.mcmc, project = nimble.model)
R> C.mcmc$run(11000)
R> mcmc.trace <- as.matrix(C.mcmc$mvSamples)[1001:11000, ]

```

Only some of the parameters for which we have a sample from the joint posterior are needed to compute PTEs. We extract these columns to form the `params` object for later use in



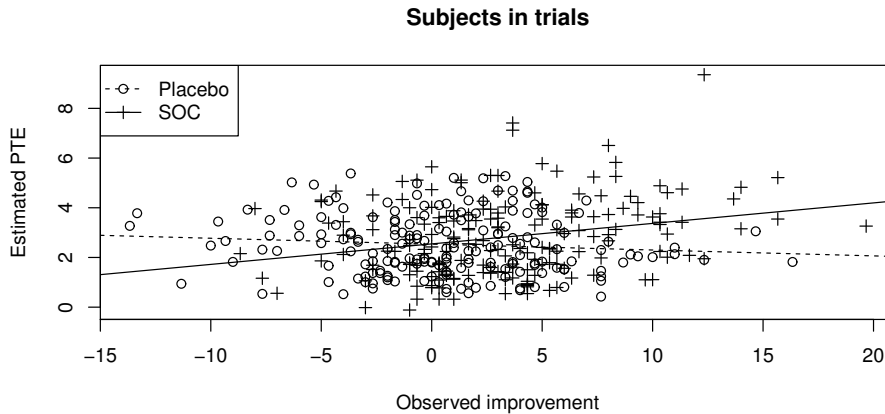


Figure 1: Estimated (posterior mean) personalized treatment effects and observed improvements in cognitive function for subjects in trials.

constructing credible subsets. We can compute a posterior sample of the PTEs for subjects included in the dataset by applying the appropriate regression parameters to a design matrix with all treatment indicators set to 1.

```
R> pte.params <- mcmc.trace[, c("gamma[1]", "gamma[2]", "gamma[3]",
+ "gamma[4]", "gamma[5]", "gamma[6]")]
R> subject.mat <- alzheimers[setdiff(colnames(alzheimers), "Improvement")]
R> subject.mat$Treatment <- 1
R> subject.mat$Sex.by.Carrier <- subject.mat$Sex * subject.mat$Carrier
R> subject.pte <- as.matrix(subject.mat) %>% t(pte.params)
R> subject.pte.est <- rowMeans(subject.pte)
R> plot(subject.pte.est ~ alzheimers$Improvement,
+ pch = ifelse(alzheimers$Treatment, 3, 1), main = "Subjects in trials",
+ xlab = "Observed improvement", ylab = "Estimated PTE")
R> abline(lm(subject.pte.est[alzheimers$Treatment == 1] ~
+ alzheimers$Improvement[alzheimers$Treatment == 1]), lty = 1)
R> abline(lm(subject.pte.est[alzheimers$Treatment == 0] ~
+ alzheimers$Improvement[alzheimers$Treatment == 0]), lty = 2)
R> legend("topleft", legend = c("Placebo", "SOC"), lty = c(2, 1),
+ pch = c(1, 3))
```

Figure 1 displays the estimated PTEs versus the observed improvements, stratified by treatment arm. Note that while the observed improvements visually appear related to treatment assignment (observed improvements in the control group tend to be farther left than those in the treatment group), the estimated PTEs do not. Least squares regression lines are also included, indicating a positive correlation between improvement and PTE in the treatment arm, but not in the control arm. Standard methods of hypothesis testing for these relationships are not applicable, as the estimated PTEs are not independent, and are omitted.

When identifying the benefiting subgroup, we consider not only the covariate profiles of subjects enrolled in the trials, but those of all potential patients. Thus, we construct our `design` object as a design matrix representing the covariate space over which we will make



inferences. We must place the `design` matrix on the same (here, standardized) scale as the data.

```
R> design <- expand.grid(Treatment = 1, Severity = 9:49,
+   Decline = 0:8, Sex = c(0, 1), Carrier = c(0, 1))
R> design$Severity <- scale(design$Severity,
+   center = attr(alzheimers$Severity, "scaled:center"),
+   scale = attr(alzheimers$Severity, "scaled:scale"))
R> design$Decline <- scale(design$Decline,
+   center = attr(alzheimers$Decline, "scaled:center"),
+   scale = attr(alzheimers$Decline, "scaled:scale"))
R> design$Sex.by.Carrier <- design$Sex * design$Carrier
R> head(design)
```

	Treatment	Severity	Decline	Sex	Carrier	Sex.by.Carrier
1	1	-1.613593	-1.283438	0	0	0
2	1	-1.525112	-1.283438	0	0	0
3	1	-1.436632	-1.283438	0	0	0
4	1	-1.348151	-1.283438	0	0	0
5	1	-1.259670	-1.283438	0	0	0
6	1	-1.171190	-1.283438	0	0	0

### 3.2. Inference

Four arguments determine the behavior of the primary functions in ways that may affect their primary outputs:

- `cred.level`, the credible level  $1 - \alpha$  (default 0.95) at which simultaneous credible bands or credible subset pairs are constructed;
- `threshold`, the threshold  $\delta$  (default 0) of  $b(\mathbf{x}; \boldsymbol{\theta})$  for membership in  $B(\boldsymbol{\theta})$ ;
- `method`, either "asymptotic" (default, using (1)) or "quantile" (using (3)), specifying which simultaneous credible band construction method is used;
- `step.down`, whether or not the step-down procedure should be used (default TRUE).

Table 1 displays the applicability of each of the above arguments to the primary functions. Additionally, the argument `sides` (default "both") may be used to specify one-sided constructions by passing the values "lower" or "upper" for `sim.cred.band()`, and "exclusive" or "inclusive" for `credsubs()` and `credsubs.level()`.

The output of `sim.cred.band()` is a list of class 'sim.cred.band' with the primary elements `upper` and `lower`. These are numeric vectors of length equal to `nrow(design)`, if provided, and to `ncol(params)` otherwise. Their entries are the upper and lower bounds, respectively, of the simultaneous credible band at the corresponding covariate point.

```
R> scb <- sim.cred.band(params = pte.params, design = design)
R> scb
```

	sim.cred.band	credsubs	credsubs.level
cred.level	✓	✓	
threshold		✓	✓
method	✓	✓	✓
step.down		✓	✓

Table 1: Applicability of behavior arguments to primary functions.

## Simultaneous credible band

Credible level: 0.95  
Method: asymptotic  
Critical W: 3.082141

Upper bounds: 3.333657 3.358145 3.386394 3.418632 3.45509 3.495999 ...  
Lower bounds: -3.391408 -3.231213 -3.074779 -2.922334 -2.77411 -2.630336 ...

The output of `credsubs()` is a list of class ‘`credsubs`’ with the primary elements `exclusive` and `inclusive`, which are logical vectors of the same lengths as the primary elements of the output of `sim.cred.band()`. An element of `exclusive` equal to `TRUE` indicates that the corresponding covariate point is a member of the exclusive credible subset `D`; a similar relationship holds for `inclusive` and the inclusive credible subset `S`.

```
R> cs <- credsubs(params = pte.params, design = design)
R> cs
```

## Credible subsets

Credible level: 0.95  
Threshold: 0  
Method: asymptotic  
Step-down: TRUE  
Critical W: 2.98266

Exclusive credible subset: FALSE FALSE FALSE FALSE FALSE FALSE ...  
Inclusive credible subset: TRUE TRUE TRUE TRUE TRUE TRUE ...

The output of `credsubs.level()` is a list of class ‘`credsubs.level`’ and is somewhat more complicated. The primary elements are numerical vectors `level`, which gives the maximum credible level at which each covariate point is not a member of  $S \setminus D$ , and `sign`, which is equal to 1 if the covariate point is in `D` for credible levels less than or equal to that specified by `level` and  $-1$  if it is in  $S^c$  for the same credible levels. If a covariate point is always in  $S \setminus D$  (i.e., the credible band is centered at  $\delta$  there), its entries in `level` and `sign` are both 0.

```
R> cs1 <- credsubs.level(params = pte.params, design = design)
R> cs1
```

Maximum credible levels

Threshold: 0

Method: asymptotic

Step-down: TRUE

Maximum levels: 0.0053 0.0053 0.0054 0.01 0.0195 0.0195 ...

Signs: -1 1 1 1 1 1 ...

### 3.3. Diagnostics

The package offers several facilities for diagnostics of simultaneous credible band and credible subsets outputs. This section describes them and presents illustrations.

#### *Target function distributions*

The material input to the credible subsets methods is the sample from the joint posterior of the  $b(\mathbf{x}; \boldsymbol{\theta})$ , and thus it is an important target of diagnostics. In particular, the realized distribution of each of the  $b(\mathbf{x}; \boldsymbol{\theta})$  should be examined to determine whether the asymptotic simultaneous credible band (1) is appropriate, or if the quantile-based version (3) should be used instead.

For parametric models, a sample from the posterior of  $\boldsymbol{\theta}$  and the matrix representing  $\mathbf{X}$  are passed to the package's functions and the  $b(\mathbf{x}; \boldsymbol{\theta})$  computed internally. In this case the user may request the samples of the  $b(\mathbf{x}; \boldsymbol{\theta})$  from a subset (or all) of the  $\mathbf{x}$  using the `track` argument to any of the primary functions. The argument (default `numeric(0)`) is a vector containing the indices of the  $\mathbf{x}$  in  $\mathbf{X}$  for which the sample of the  $b(\mathbf{x}; \boldsymbol{\theta})$  should be returned. For example, setting `track = c(1, 10, 100)` would result in the output of `sim.cred.band()` containing a 3-column matrix `trace` whose columns are samples of  $b(\mathbf{x}; \boldsymbol{\theta})$  where  $\mathbf{x}$  is the first, tenth, and hundredth row of `design`. The value of `track` is retained as the column names of `trace`.

```
R> scb <- sim.cred.band(params = pte.params, design = design,
+   track = c(1, 10, 100))
R> hist(scb$trace[, "1"], xlab = "Value",
+   main = expression("Distribution of b(x"[1]*")"))
R> hist(scb$trace[, "10"], xlab = "Value",
+   main = expression("Distribution of b(x"[10]*")"))
```

The output is shown in Figure 2. The approximate normality of  $b(\mathbf{x}_1; \boldsymbol{\theta})$  and  $b(\mathbf{x}_{10}; \boldsymbol{\theta})$  indicates that the asymptotic simultaneous credible bands may be used. The same process may be repeated at several “sentinel” points throughout the covariate space.

#### *Band and credible level plots*

The ‘`sim.cred.band`’ and ‘`credsubs.level`’ objects are equipped with plotting methods that produce visualizations ignoring the spatial relationships between covariate points. Discarding the covariate information allows generic plots to be produced regardless of the quantity, type, or range of the covariates. In the case of the ‘`sim.cred.band`’ object, the covariate points

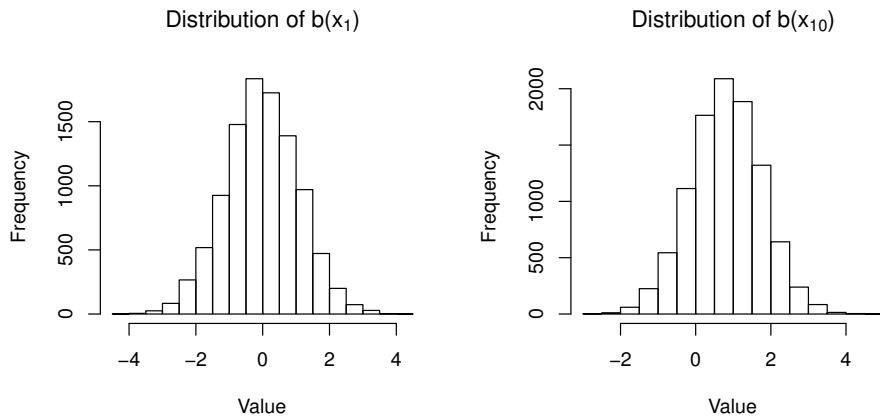
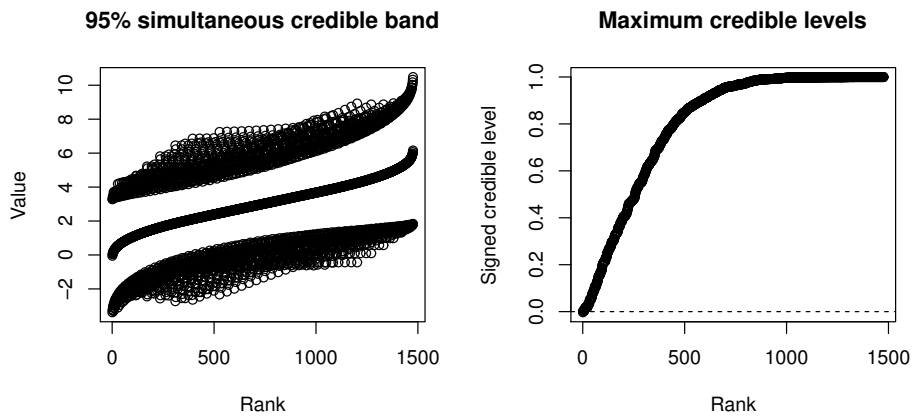
Figure 2: Histograms of the computed samples of  $b(\mathbf{x}_1; \boldsymbol{\theta})$  and  $b(\mathbf{x}_{10}; \boldsymbol{\theta})$ .

Figure 3: Plots of the simultaneous credible band and maximum credible levels.

are placed in ascending order according to the estimates of  $b(\mathbf{x}; \boldsymbol{\theta})$ , by default their posterior means (`method = "asymptotic"`) or medians (`method = "quantile"`). The estimates are then plotted along with the corresponding upper and lower bounds. The `plot` method for `'credsubs.level'` objects plots the signed maximum credible levels `credsubs.level$sign * credsubs.level$level` in ascending order.

```
R> plot(scb, main = "95% simultaneous credible band")
R> plot(csl, main = "Maximum credible levels")
```

The output is shown in Figure 3. The plot of the simultaneous credible band may be used to examine the range of widths of the band and compare its bounds to arbitrary thresholds. In the presented case the majority of the estimates, along with many lower bounds, are above zero, and no upper bounds are below zero. That of the maximum credible levels may be examined to determine the proportion of covariate points falling outside of  $S \setminus D$  at various credible levels. Here, roughly two-thirds of the points have maximum credible levels above 80%.

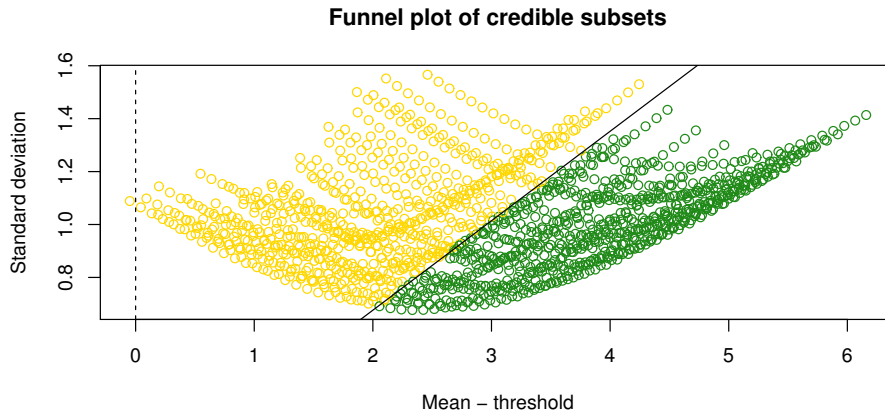


Figure 4: Posterior means and standard deviations of target function at each covariate point. Points below the sloped line (green) are in the exclusive credible subset, while the rest (gold) constitute the remainder of the inclusive credible subset.

### Funnel plots

The ‘`credsubs`’ objects also have a `plot` method inspired by the funnel plots often used to detect publication bias in meta-analysis (Richard and Pillemer 1984). Funnel plots relate estimates – study treatment effects in meta-analyses, and  $\bar{b}(\mathbf{x}; \boldsymbol{\theta}) - \delta$  in credible subsets analyses – to their standard errors. Since constructing a funnel plot in our context requires a posterior mean and standard error, the method is only available for objects created with the `method = "asymptotic"` argument. If visible, the vertical and horizontal lines through the origin, as well as those through the origin with slopes  $\pm 1/W_{\alpha, X}^*$ , are added. Points falling below the positively sloped line are members of the exclusive credible subset  $D$ , while those falling below the negatively sloped line are not members of the inclusive credible subset  $S$ . Points falling above both lines fall in  $S \setminus D$ . When `step.down = TRUE`, the value of  $W_{\alpha, X}^*$  from the final iteration is used.

```
R> plot(cs, main = "Funnel plot of credible subsets")
```

Figure 4 displays the result. The funnel plot may be used to understand how the distribution of points among  $D$ ,  $S \setminus D$ , and  $S^c$  change as the threshold varies (points are translated horizontally) and as the credible level decreases (sloped lines approach vertical) or increases (approach horizontal).

### 3.4. Reporting

The package contains functions for building and running a **shiny** (Chang, Cheng, Allaire, Xie, and McPherson 2020) application that allows users to choose a covariate point by selecting levels from a list for each covariate, and obtain the maximum credible level and interpretation. The application is built by passing a ‘`credsubs.level`’ object along with a data frame `cov.space` representing the covariate space to `build.shiny.calc()`. The `cov.space` object should be formatted to be human-readable, e.g., covariate values on the original scales, and well-formatted column names and factor levels.

## Calculator

Select a covariate point.

### Baseline Severity

21

### Rate of Decline

5

### Sex

Female

### Carrier Status

Non-Carrier

## Result

The covariate point selected is **in the exclusive credible subset** at a maximum credible level of **91.85%**. At higher credible levels, no conclusion may be drawn.

Close

Figure 5: Example of calculator for reporting the Alzheimer's disease dataset credible subset results.

```
R> cov.space <- expand.grid("Baseline Severity" = 9:49,
+   "Rate of Decline" = 0:8, "Sex" = c("Female", "Male"),
+   "Carrier Status" = c("Non-Carrier", "Carrier"))
R> build.shiny.calc(csl, cov.space, name = "alzheimers")
```

```
Building calculator in ./alzheimers/
Execute run.shiny.calc('./alzheimers') to run.
```

```
R> run.shiny.calc("alzheimers")
```

An image of the example calculator is displayed in Figure 5. The `build.shiny.calc()` function creates a directory containing the **shiny** application, which can be moved to any location or deployed online. The argument to `run.shiny.calc()` need only point to the intended directory.

The credible subsets represent bounds on the benefiting subgroup with a clinically useful interpretation: With high credibility (or confidence, depending on the desired paradigm), researchers can make the claim that all types of patients in the exclusive credible subset have

a positive PTE, and all types of patients with positive PTEs are contained in the inclusive credible subset. A primary utility of such a global inference is regulatory: Investigators control the probability of erroneously declaring benefit for one or more types of patients that do not in fact benefit. This family-wise Type I error control is critical when submitting results to a regulatory agency such as the United States Food and Drug Administration. Another use is in designing enrichment studies, which target a particular population thought to be more likely to benefit or have a stronger treatment effect. In this case, the calculator may be used to define eligibility criteria, e.g., in order to enroll, patients must be part of the exclusive credible subgroup at a maximum credible level of 90% or higher. The multiplicity control provided by this approach then protects against proceeding with an enrichment trial on a target population that is either overoptimistically large or entirely spurious.

#### 4. A nonparametric example

The example usage presented in Section 3 is parametric both in the modeling and credible subsets senses: A subset of the regression parameters in the model are collected in  $\theta$  and passed to the package's functions along with a design matrix. In this section we present a usage which is nonparametric in the sense that a posterior draw from  $\mathbf{b}(C)$  is passed directly. Additionally, the following analysis is an example outside of the clinical trials scope for which the credible subsets method was originally developed.

We analyze a neuroimaging dataset derived from that described in the **SPM12** manual ([The FIL Methods Group 2014](#)), Chapter 30, and provided in the supplementary materials. The data are from an fMRI experiment in which subjects were given alternating blocks of auditory stimulus and no stimulus. We preprocessed the data using **FSL** ([Woolrich \*et al.\* 2009](#)). The preprocessing steps were as follows. We applied motion correction using `mcfliirt` ([Jenkinson, Bannister, Brady, and Smith 2002](#); rigid body transform; cost function normalized correlation; reference volume the middle volume), and then we smoothed the images using a Gaussian kernel with `FWHM = 5mm`. We consider data from one patient, which consist of 96 blood oxygen level scans of  $64 \times 64 \times 64$  voxels. We restrict our attention to the 33rd horizontal slice, and at the advice of the **SPM12** Manual, we ignore the first 12 scans to avoid start-up artifacts. Each scan lasts for 7 seconds, and the scans are grouped into blocks of 6. The blocks alternate between no stimulus and an auditory stimulus, beginning with a no-stimulus block. The blood oxygen level does not respond instantaneously to stimulus; instead, the canonical hemodynamic response function (HRF; [Friston, Fletcher, Josephs, Holmes, Rugg, and Turner 1998](#)) is convolved with the stimulus indicator to produce the design coefficient  $x(t)$  pattern shown in Figure 6.

```
R> library("credsubs")
R> library("neuRosim")
R> library("nimble")
R> load("slice32.RData")
R> slice.arr <- array(as.vector(t(slice[-(1:3), ])), dim = c(64, 64, 96))
R> experiment <- rep(rep(c(0, 1), each = 6), 8)
R> chrf <- canonicalHRF(1:96 * 7)
R> design <- convolve(experiment, rev(chrf), type = "open")[1:96]
R> plot(experiment, main = "Experiment and design coefficients",
+       ylim = range(experiment, design), ylab = "Coefficient", xlab = "Scan")
```



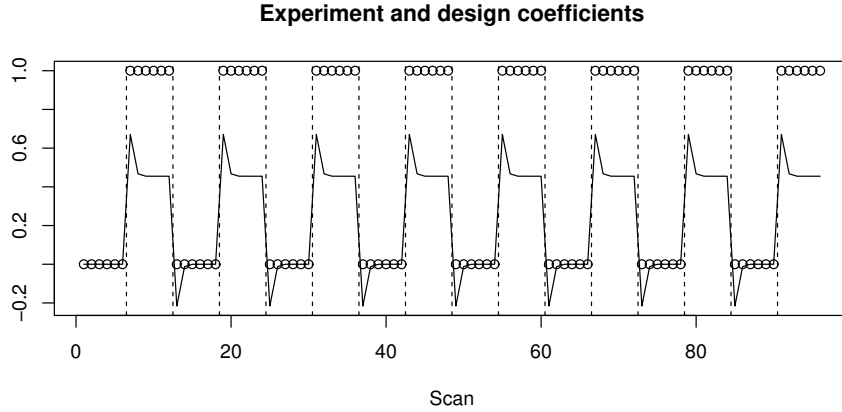


Figure 6: Raw experimental design (o) and HRF-convolved design (—) coefficients.

```
R> lines(design)
R> abline(v = seq(6, 90, by = 6) + 1 / 2, lty = 2)
```

For the sake of simple illustration, we model each voxel individually. The blood oxygen level measurement for the voxel  $v$  at time  $t$  is denoted  $Y_v(t)$ , and modeled as

$$Y_v(t) = \alpha_v + \beta_v x(t) + (t - 1)s_v + \epsilon_v(t), \quad (5)$$

where  $\epsilon_v(t)$  is a normal lag-1 autoregressive (AR(1)) process and the  $s_v$  coefficient models scanner drift. For the prior specification,  $\alpha_v, \beta_v \stackrel{\text{iid}}{\sim} \text{Normal}[0, 10^{-6}]$ , the autoregressive coefficient  $\rho_v \stackrel{\text{iid}}{\sim} \text{Uniform}[-1, 1]$ , and the prior precision of  $\epsilon_v(t)|\epsilon_v(t - 1)$  is  $\tau_v \stackrel{\text{iid}}{\sim} \text{Gamma}[10^{-4}, 10^{-4}]$ .

```
R> M <- 1000
R> trace <- array(NA, dim = c(64, 64, M))
R> nimble.code <- nimbleCode({
+   r[1] <- 0
+   bold[1] ~ dnorm(alpha + design[1] * beta + scan * (1) + r[1] * rho,
+     tau)
+   for (t in 2:N.t) {
+     r[t] <- bold[t - 1] - (alpha + beta * design[t - 1] + scan * (t - 1))
+     bold[t] ~ dnorm(alpha + design[t] * beta + scan * (t) + r[t] * rho,
+     tau)
+   }
+
+   alpha ~ dnorm(0, 1E-6)
+   beta ~ dnorm(0, 1E-6)
+   rho ~ dunif(-1, 1)
+   tau ~ dgamma(1E-4, 1E-4)
+   scan ~ dnorm(0, 1E-6)
+ })
R> nimble.constants <- list(N.t = 84)
R> nimble.data <- list(bold = slice.arr[1, 1, 13:96], design = design[13:96])
R> nimble.inits <- list(alpha = 0, beta = 0, rho = 0, tau = 1, scan = 0)
```

```

R> nimble.model <- nimbleModel(code = nimble.code, name = "example",
+   constants = nimble.constants, data = nimble.data, inits = nimble.inits)
R> nimble.spec <- configureMCMC(nimble.model, monitors = "beta")
R> nimble.mcmc <- buildMCMC(nimble.spec, progressBar = FALSE)
R> C.model <- compileNimble(nimble.model)
R> C.mcmc <- compileNimble(nimble.mcmc, project = nimble.model)
R> set.seed(1)
R> for (x in 1:64) {
+   cat(x, "\n")
+   for (y in 1:64) {
+     nimble.data <- list(bold = slice.arr[x, y, 13:96])
+     C.model$setData(nimble.data)
+     C.mcmc$run(niter = M * 1.1, nburnin = M * 0.1, progressBar = FALSE)
+     trace[x, y, ] <- as.matrix(C.mcmc$mvSamples, "beta")
+   }
+ }

```

A large portion of the scan slice is outside the brain. As a crude method of restricting further inference to the brain alone, we select the voxels for which the observed mean blood oxygen level measurement is at least 1,000.

```
R> brain <- rowMeans(slice.arr[, , 13:96], dims = 2) >= 1000
```

The activation coefficient  $\beta_v$  is our primary inferential target. We identify the voxels at which the ratio of the posterior mean to posterior standard deviation corresponds to the 10th and 90th quantiles of the standard normal distribution, and also compute the maximum credible levels for each voxel.

```

R> est <- rowMeans(trace, dims = 2)
R> sd <- apply(trace, c(1, 2), sd)
R> z.score <- matrix(NA, nrow = 64, ncol = 64)
R> z.score[brain] <- est[brain] / sd[brain]
R> p <- (pnorm(abs(z.score)) > 0.90) * sign(z.score)
R> params <- trace
R> dim(params) <- c(64 * 64, M)
R> params <- t(params)
R> csl <- credsubs.level(params[, brain], threshold = 0, sides = "both")
R> thresh.mat <- matrix(NA, nrow = 64, ncol = 64)
R> thresh.mat[brain] <- (csl$level >= 0.80) * csl$sign
R> plot(csl)
R> colors <- c("firebrick", "gold", "forestgreen")
R> par(mai = c(0.5, 0.5, 0.5, 0.5), oma = c(0, 0, 0, 0), mfrow = c(1, 2))
R> image(p, col = colors, ylim = c(1.025, -0.025), xaxt = "n", yaxt = "n",
+   main = "Uncorrected 80% Tests", useRaster = TRUE)
R> image(thresh.mat, col = colors, ylim = c(1.025, -0.025), xaxt = "n",
+   yaxt = "n", main = "80% Credible Subsets", useRaster = TRUE)

```

Figure 7 shows the signed maximum credible levels for the  $\beta_v$ . The shape shown, in which most voxels have maximum credible levels very close to 0, and the remaining few have much

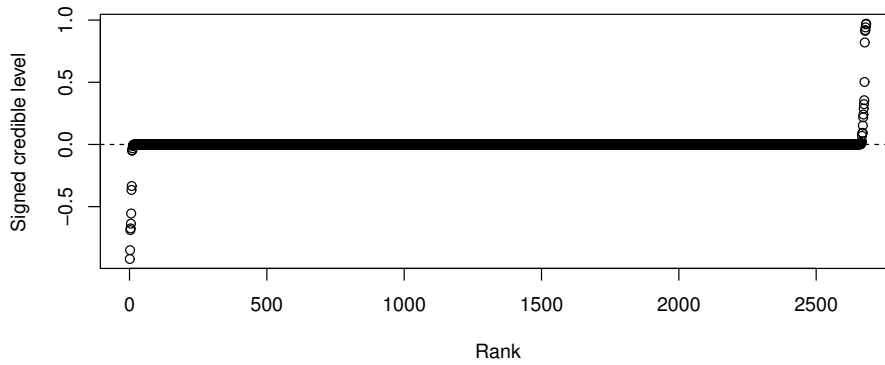
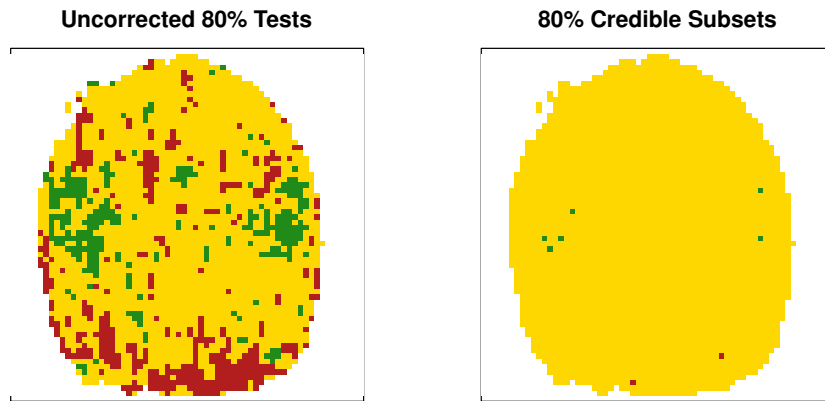
Figure 7: Maximum credible levels for the  $\beta_v$ .

Figure 8: Spatial representations. Red indicates negative activation, and green positive.

larger maximum credible levels, suggests that most of the voxels are in truth not activated. Compare this to the maximum credible levels shown in Figure 3, which steadily climb from near 0 to near 1, suggesting that the treatment effect is positive over most of the covariate space but the data lack the strength to make a conclusion.

The left side of Figure 8 displays the voxels for which the ratio of posterior mean to standard deviation is below the 10th or above the 90th quantile of the standard normal distribution. Note that a great many voxels are “significant” in this sense, but the result appears noisy. The right side displays the 80% credible subsets. Because the posterior distributions of the  $\beta_v$  are uncorrelated due to our independent modeling strategy, the credible subsets procedure is very underpowered. A model in which the  $\alpha_v$  and  $\beta_v$  are spatially correlated, and other parameters are perhaps shared across all voxels, is likely to yield a much larger exclusive credible subset and smaller inclusive credible subset. Regardless of inefficiency, several voxels are identified as positively or negatively activated. These voxels appear to fall into three clusters: The middle left and right green voxels lie in the auditory cortex, while the lower red voxels lie in the visual cortex – sensible conclusions about responses to auditory stimuli.

## 5. Discussion

This article presented the R package **credsubs** for implementing the credible subsets inference procedure, which provides exclusive and inclusive bounds for a target subset of the covariate space. The essential input to package functions is a sample from the joint posterior from an arbitrary regression model, which may be obtained via various other software packages, for example **nimble**. The credible subsets approach was initially developed in order to identify benefiting subgroups in clinical trials, as illustrated by the example in Section 3, but is also applicable to a wide range of other settings, for example the neuroimaging analysis in Section 4.

In the clinical trial context, we hope that our package can form the basis for a “lingua franca” between drug developers and regulatory authorities, the former of whom are increasingly interested in targeting therapies to relevant patient subgroups, while also satisfying the latter group with regard to overall Type I error protection. We also hope to explore the capabilities of this approach in other areas. For example, it is likely that the neuroimaging example could be greatly improved by taking into account spatial correlation among the activation levels via a conditional autoregressive (CAR; Besag 1974, 1975) or other smoothing prior.

Other future work may include investigations into frequentist analogs through bootstrap sampling, and parallel computing or other methods to handle larger and higher-dimensional samples.

## Acknowledgments

This work was supported by AbbVie, Inc.

## References

- Besag J (1974). “Spatial Interaction and the Statistical Analysis of Lattice Systems.” *Journal of the Royal Statistical Society B*, **36**(2), 192–225. doi:10.1111/j.2517-6161.1974.tb00999.x.
- Besag J (1975). “Statistical Analysis of Non-Lattice Data.” *The Statistician*, **24**(3), 179–195. doi:10.2307/2987782.
- Breiman L, Friedman J, Stone CJ, Olshen RA (1984). *Classification and Regression Trees*. CRC Press, Boca Raton.
- Carlin BP, Louis TA (2009). *Bayesian Methods for Data Analysis*. CRC Press, Boca Raton.
- Carpenter B, Gelman A, Hoffman M, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A (2017). “Stan: A Probabilistic Programming Language.” *Journal of Statistical Software*, **76**(1), 1–32. doi:10.18637/jss.v076.i01.
- Chang W, Cheng J, Allaire J, Xie Y, McPherson J (2020). **shiny**: *Web Application Framework for R*. R package version 1.4.0.2, URL <https://CRAN.R-project.org/package=shiny>.

- Chipman H, McCulloch R (2016). **BayesTree**: *Bayesian Additive Regression Trees*. R package version 0.3-1.4, URL <https://CRAN.R-project.org/package=BayesTree>.
- Chipman HA, George EI, McCulloch RE (1998). “Bayesian CART Model Search.” *Journal of the American Statistical Association*, **93**(443), 935–948. doi:10.1080/01621459.1998.10473750.
- Chipman HA, George EI, McCulloch RE (2010). “BART: Bayesian Additive Regression Trees.” *The Annals of Applied Statistics*, **4**(1), 266–298. doi:10.1214/09-aoas285.
- de Valpine P, Paciorek C, Turek D, Michaud N, Anderson-Bergman C, Obermeyer F, Wehrhahn Cortes C, Rodriguez A, Paganin S (2019). **nimble**: *MCMC, Particle Filtering, and Programmable Hierarchical Modeling*. R package version 0.9.0, URL <https://CRAN.R-project.org/package=nimble>.
- de Valpine P, Turek D, Paciorek C, Anderson-Bergman C, Temple Lang D, Bodik R (2017). “Programming with Models: Writing Statistical Algorithms for General Model Structures with **nimble**.” *Journal of Computational and Graphical Statistics*, **26**(2), 403–416. doi:10.1080/10618600.2016.1172487.
- Friston KJ, Fletcher P, Josephs O, Holmes A, Rugg MD, Turner R (1998). “Event-Related fMRI: Characterizing Differential Responses.” *NeuroImage*, **7**(1), 30–40. doi:10.1006/nimg.1997.0306.
- Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB (2013). *Bayesian Data Analysis*. 3rd edition. CRC Press, Boca Raton, FL. doi:10.1201/b16018.
- Holm S (1979). “A Simple Sequentially Rejective Multiple Test Procedure.” *Scandinavian Journal of Statistics*, **6**(2), 65–70.
- Jenkinson M, Bannister P, Brady M, Smith S (2002). “Improved Optimization for the Robust and Accurate Linear Registration and Motion Correction of Brain Images.” *NeuroImage*, **17**(2), 825–841. doi:10.1006/nimg.2002.1132.
- Lunn D, Spiegelhalter D, Thomas A, Best N (2009). “The BUGS Project: Evolution, Critique and Future Directions.” *Statistics in Medicine*, **28**(25), 3049–3067. doi:10.1002/sim.3680.
- Marcus R, Peritz E, Ruben KG (1976). “On Closed Testing Procedures with Special Reference to Ordered Analysis of Variance.” *Biometrika*, **63**(3), 655–660. doi:10.1093/biomet/63.3.655.
- Martins TG, Simpson D, Lindgren F, Rue H (2013). “Bayesian Computing with **INLA**: New Features.” *Computational Statistics & Data Analysis*, **67**, 68–83. doi:10.1016/j.csda.2013.04.014.
- Plummer M (2019). **rjags**: *Bayesian Graphical Models Using MCMC*. R package version 4-10, URL <https://CRAN.R-project.org/package=rjags>.
- R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

- Richard J, Pillemer DB (1984). *Summing Up: The Science of Reviewing Research*. Harvard University Press.
- Schnell P, Carlin B (2020). **credsubs**: *Credible Subsets*. R package version 1.1.1, URL <https://CRAN.R-project.org/package=credsubs>.
- Schnell PM, Müller P, Tang Q, Carlin BP (2018). “Multiplicity-Adjusted Semiparametric Benefiting Subgroup Identification in Clinical Trials.” *Clinical Trials*, **15**(1), 75–86. doi:10.1177/1740774517729167.
- Schnell PM, Tang Q, Müller P, Carlin BP (2017). “Subgroup Inference for Multiple Treatments and Multiple Endpoints.” *The Annals of Applied Statistics*, **11**(2), 949–966. doi:10.1214/17-aos1024.
- Schnell PM, Tang Q, Offen WW, Carlin BP (2016). “A Bayesian Credible Subgroups Approach to Identifying Patient Subgroups with Positive Treatment Effects.” *Biometrics*, **72**(4), 1026–1036. doi:10.1111/biom.12522.
- Simon R (2002). “Bayesian Subset Analysis: Application to Studying Treatment-by-Gender Interactions.” *Statistics in Medicine*, **21**(19), 2909–2916. doi:10.1002/sim.1295.
- Stan Development Team (2020). “**rstan**: R Interface to Stan.” R package version 2.19.3, URL <http://mc-stan.org/>.
- Sturtz S, Ligges U, Gelman A (2005). “**R2WinBUGS**: A Package for Running WinBUGS from R.” *Journal of Statistical Software*, **12**(3), 1–16. doi:10.18637/jss.v012.i03.
- The FIL Methods Group (2014). **SPM12 Manual**. URL <http://www.fil.ion.ucl.ac.uk/spm/software/>.
- Uusipaikka E (1983). “Exact Confidence Bands for Linear Regression over Intervals.” *Journal of the American Statistical Association*, **78**(383), 638–644. doi:10.1080/01621459.1983.10478023.
- Woolrich MW, Jbabdi S, Patenaude B, Chappell M, Makni S, Behrens T, Beckmann C, Jenkinson M, Smith SM (2009). “Bayesian Analysis of Neuroimaging Data in **FSL**.” *NeuroImage*, **45**(1), S173–S186. doi:10.1016/j.neuroimage.2008.10.055.
- Zhang Z, Seibold H, Vettore MV, Song WJ, François V (2018). “Subgroup Identification in Clinical Trials: An Overview of Available Methods and Their Implementations with R.” *Annals of Translational Medicine*, **6**(7). doi:10.21037/atm.2018.03.07.

**Affiliation:**

Patrick M. Schnell  
Division of Biostatistics  
College of Public Health  
The Ohio State University  
1841 Neil Ave  
Columbus, OH 43210, United States of America  
E-mail: [schnell.31@osu.edu](mailto:schnell.31@osu.edu)