# Package 'SSVS'

**Title** Functions for Stochastic Search Variable Selection (SSVS)

**Version** 2.1.0

**Description** Functions for performing stochastic search variable selection (SSVS)
for binary and continuous outcomes and visualizing the results.
SSVS is a Bayesian variable selection method used to estimate the probability
that individual predictors should be included in a regression model.
Using MCMC estimation, the method samples thousands of regression models
in order to characterize the model uncertainty regarding both the predictor
set and the regression parameters. For details see Bainter, McCauley, Wager,
and Losin (2020) Improving practices for selecting a subset of important
predictors in psychology: An application to predicting pain, Advances in
Methods and Practices in Psychological Science 3(1), 66-80
<DOI:10.1177/2515245919885617>.

**URL** https://github.com/sabainter/SSVS

**BugReports** https://github.com/sabainter/SSVS/issues

**Depends** R (>= 3.5.0)

**Imports** bayestestR, BoomSpikeSlab, checkmate, ggplot2, graphics,
rlang, stats, dplyr, magrittr, gridExtra

**Suggests** AER, bslib, foreign, glue, knitr, mice, psych, reactable,
readxl, rmarkdown, scales, shiny, shinyjs, shinyWidgets,
testthat (>= 3.0.0), tools, utils

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.2

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Sierra Bainter [cre, aut] (<https://orcid.org/0000-0001-7461-0803>),
Thomas McCauley [aut],
Mahmoud Fahmy [aut],
Dean Attali [aut] (<https://orcid.org/0000-0002-5645-3493>)

**Maintainer** Sierra Bainter <sbainter@miami.edu>

**Repository** CRAN

**Date/Publication** 2025-03-19 16:40:02 UTC

# Contents

---

| dat | *Example dataset for* `ssvs` *function @format A data frame with 74 records and 76 variables* |
|-----|-----|

---

## Description

Example dataset for `ssvs` function @format A data frame with 74 records and 76 variables

## Usage

```
dat
```

## Format

An object of class `data.frame` with 74 rows and 76 columns.

---

| imputed_mtcars | *Imputed mtcars Dataset* |
|-----|-----|

---

## Description

This dataset is a version of the `mtcars` dataset where random missing values were introduced, and multiple imputation was performed using the `mice` package.

## Usage

```
imputed_mtcars
```

## Format

A data frame with 160 rows and 13 variables

## Details

Random missingness was introduced into 10% of the values in the original `mtcars` dataset. Multiple imputation was then performed using the `mice` package with the following parameters:

- 5 multiple imputations (`m = 5`).

- 50 iterations per imputation (`maxit = 50`).

- Seed set to 123 for reproducibility.

The dataset included here is the first completed dataset resulting from the multiple imputation process.

## Source

Original dataset from `datasets::mtcars`, with missing values introduced and imputed.

## Examples

```
data(imputed_mtcars)
head(imputed_mtcars)
```

---

| launch | *Run an interactive analysis tool (Shiny app) that lets you perform SSVS in a browser* |
|---|---|

---

## Description

Run an interactive analysis tool (Shiny app) that lets you perform SSVS in a browser

## Usage

```
launch()
```

---

plot.ssvs                           *Plot results of an SSVS model*

---

### Description

Plot results of an SSVS model

### Usage

```
## S3 method for class 'ssvs'
plot(x, threshold = 0.5, legend = TRUE, title = NULL, color = TRUE, ...)
```

### Arguments

| | |
|---|---|
| x | An ssvs result object obtained from `ssvs()` |
| threshold | An MIP threshold to show on the plot, must be between 0-1. If `NULL`, no threshold is used. |
| legend | If `TRUE`, show a legend for the shapes based on the threshold. Ignored if `threshold = NULL`. |
| title | The title of the plot. Set to `NULL` to use a default title. |
| color | If `TRUE`, the data points will be colored based on the threshold. |
| ... | Ignored |

### Value

Creates a plot of the inclusion probabilities by variable

### Examples

```
outcome <- "qsec"
predictors <- c("cyl", "disp", "hp", "drat", "wt", "vs", "am", "gear", "carb", "mpg")
results <- ssvs(x = predictors, y = outcome, data = mtcars, progress = FALSE)
plot(results)
```

---

plot.ssvs_mi                *Plot SSVS-MI Estimates and Marginal Inclusion Probabilities (MIP)*

---

### Description

This function creates a plot of SSVS-MI estimates with minimum and maximum and a plot for marginal inclusion probabilities (MIP) optional thresholds for highlighting significant predictors..

## Usage

```
## S3 method for class 'ssvs_mi'
plot(
  x,
  type = "both",
  threshold = 0.5,
  legend = TRUE,
  est_title = NULL,
  mip_title = NULL,
  color = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| x | An ssvs result object obtained from [ssvs_mi()](#) |
| type | Defaults to "both", can change to "estimate" or "MIP". |
| threshold | A numeric value (between 0 and 1) specifying the MIP threshold to highlight significant predictors. Defaults to 0.5. |
| legend | Logical indicating whether to include a legend for the threshold. Defaults to TRUE. |
| est_title | A character string specifying the plot title. Defaults to ″SSVS-MI estimates″. |
| mip_title | A character string specifying the plot title. Defaults to ″Multiple Inclusion Probability for SSVS-MI″. |
| color | Logical indicating whether to use color to highlight thresholds. Defaults to TRUE. |
| ... | Ignored |

## Value

Two ggplot2 objects representing the plot of SSVS estimates and the plot of MIP with thresholds.

## Examples

```
data(imputed_mtcars)
outcome <- 'qsec'
predictors <- c('cyl', 'disp', 'hp', 'drat', 'wt', 'vs', 'am', 'gear', 'carb','mpg')
imputation <- '.imp'
results <- ssvs_mi(data = imputed_mtcars, y = outcome, x = predictors, imp = imputation)
plot(results)
```

---

ssvs                          *Perform SSVS for continuous and binary outcomes*

---

### Description

For continuous outcomes, a basic Gibbs sampler is used. For binary outcomes, `BoomSpikeSlab::logit.spike()` is used.

### Usage

```
ssvs(
  data,
  y,
  x,
  continuous = TRUE,
  inprob = 0.5,
  runs = 20000,
  burn = 5000,
  a1 = 0.01,
  b1 = 0.01,
  prec.beta = 0.1,
  progress = TRUE
)
```

### Arguments

| | |
|---|---|
| data | The dataframe used to extract predictors and response values |
| y | The response variable |
| x | The set of predictor variables |
| continuous | If `TRUE`, treat the response variable as continuous. If `FALSE`, treat the response variable as binary. |
| inprob | Prior inclusion probability value, which applies to all predictors. The prior inclusion probability reflects the prior belief that each predictor should be included in the model. A prior inclusion probability of .5 reflects the belief that each predictor has an equal probability of being included or excluded. Note that a value of .5 also implies a prior belief that the true model contains half of the candidate predictors. The prior inclusion probability will influence the magnitude of the marginal inclusion probabilities (MIPs), but the relative pattern of MIPs is expected to remain fairly consistent, see Bainter et al. (2020) for more information. |
| runs | Total number of iterations (including burn-in). Results are based on the Total - Burn-in iterations. |
| burn | Number of burn-in iterations. Burn-in iterations are discarded warmup iterations used to achieve MCMC convergence. You may increase the number of burn-in iterations if you are having convergence issues. |

| | |
|---|---|
| a1 | Prior parameter for Gamma(a,b) distribution on the precision (1/variance) residual variance. Only used when `continuous` = `TRUE`. |
| b1 | Prior parameter for Gamma(a,b) distribution on the precision (1/variance) residual variance. Only used when `continuous` = `TRUE`. |
| prec.beta | Prior precision (1/variance) for beta coefficients. Only used when `continuous` = `TRUE`. |
| progress | If `TRUE`, show progress of the model creation. When `continuous` = `TRUE`, progress plots will be created for every 1000 iterations. When `continuous` = `FALSE`, 10 progress messages will be printed. Only used when `continuous` = `TRUE`. |

### Value

An ssvs object that can be used in [summary()](#) or [plot()](#).

### Examples

```
# Example 1: continuous response variable
outcome <- "qsec"
predictors <- c("cyl", "disp", "hp", "drat", "wt", "vs", "am", "gear", "carb", "mpg")
results <- ssvs(data = mtcars, x = predictors, y = outcome, progress = FALSE)

# Example 2: binary response variable
library(AER)
data(Affairs)
Affairs$hadaffair[Affairs$affairs > 0] <- 1
Affairs$hadaffair[Affairs$affairs == 0] <- 0
outcome <- "hadaffair"
predictors <- c("gender", "age", "yearsmarried", "children", "religiousness",
"education", "occupation", "rating")
results <- ssvs(data = Affairs, x = predictors, y = outcome, continuous = FALSE, progress = FALSE)
```

---

ssvs_mi *Perform SSVS on Multiply Imputed Datasets*

---

### Description

This function performs Stochastic Search Variable Selection (SSVS) analysis on multiply imputed datasets for a given set of predictors and a response variable. It supports continuous response variables and calculates aggregated results across multiple imputations.

### Usage

```
ssvs_mi(
  data,
  y,
  x,
  imp,
```

```
    imp_num = 5,
    interval = 0.9,
    continuous = TRUE,
    progress = FALSE
)
```

## Arguments

| | |
|---|---|
| `data` | A dataframe containing the variables of interest, including an `.imp` column for imputation identifiers. |
| `y` | The response variable (character string). |
| `x` | A vector of predictor variable names. |
| `imp` | The imputation variable. |
| `imp_num` | The number of imputations to process (default is 5). |
| `interval` | Confidence interval level for summary results (default is 0.9). |
| `continuous` | Logical indicating if the response variable is continuous (default is TRUE). |
| `progress` | Logical indicating whether to display progress (default is FALSE). |

## Value

An ssvs_mi object containing aggregated results across imputations that can be used in [`summary()`](#).

## Examples

```
data(imputed_mtcars)
outcome <- 'qsec'
predictors <- c('cyl', 'disp', 'hp', 'drat', 'wt', 'vs', 'am', 'gear', 'carb','mpg')
imputation <- '.imp'
agg_results <- ssvs_mi(data = imputed_mtcars, y = outcome, x = predictors, imp = imputation)
```

---

summary.ssvs            *Summarize results of an SSVS model*

---

## Description

Summarize results from SSVS including marginal inclusion probabilities, Bayesian model averaged parameter estimates, and 95% highest posterior density credible intervals. Estimates and credible intervals are based on standardized X variables.

## Usage

```
## S3 method for class 'ssvs'
summary(object, interval = 0.89, threshold = 0, ordered = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `object` | An SSVS result object obtained from `ssvs()` |
| `interval` | The desired probability for the credible interval, specified as a decimal |
| `threshold` | Minimum MIP threshold where a predictor will be shown in the output, specified as a decimal |
| `ordered` | If `TRUE`, order the results based on MIP (in descending order) |
| `...` | Ignored |

## Value

A dataframe with results

## Examples

```
outcome <- "qsec"
predictors <- c("cyl", "disp", "hp", "drat", "wt", "vs", "am", "gear", "carb", "mpg")
results <- ssvs(data = mtcars, x = predictors, y = outcome, progress = FALSE)
summary(results, interval = 0.9, ordered = TRUE)
```

---

summary.ssvs_mi          *Calculate Summary Statistics for SSVS-MI Results*

---

### Description

Computes summary statistics (average, minimum, and maximum) for beta coefficients, MIP and average nonzero beta coefficients from an SSVS result object.

### Usage

```
## S3 method for class 'ssvs_mi'
summary(object, ...)
```

### Arguments

| | |
|---|---|
| `object` | An ssvs_mi result object obtained from `ssvs_mi()` |
| `...` | Ignored |

### Value

A data frame with results

## Examples

```
data(imputed_mtcars)
outcome <- 'qsec'
predictors <- c('cyl', 'disp', 'hp', 'drat', 'wt', 'vs', 'am', 'gear', 'carb','mpg')
imputation <- '.imp'
results <- ssvs_mi(data = imputed_mtcars, y = outcome, x = predictors, imp = imputation)
summary_MI<-summary(results)
print(summary_MI)
```

# Index