

# Package: rsides (via r-universe)

September 9, 2024

**Type** Package

**Title** SIDES-Based Subgroup Search Algorithms

**Description** R implementation of SIDES-based subgroup search algorithms  
(Lipkovich et al. (2017) <[doi:10.1002/sim.7064](https://doi.org/10.1002/sim.7064)>).

**License** GPL (>= 2)

**Version** 0.1

**Date** 2024-05-27

**Date/Publication** 2024-06-10 17:00:02 UTC

**Repository** CRAN

**Author** Alex Dmitrienko [aut, cre]

**Maintainer** Alex Dmitrienko <[admitrienko@mediana.us](mailto:admitrienko@mediana.us)>

**Depends** R (>= 3.1.2)

**Imports** Rcpp (>= 0.12.10), survival, xml2, foreach, parallel,  
doParallel, doRNG, officer, flextable

**Suggests** testthat (>= 3.0.0)

**LinkingTo** Rcpp, RcppEigen, BH

**NeedsCompilation** yes

**Encoding** UTF-8

**LazyLoad** yes

**LazyData** true

**Config/testthat/edition** 3

## Contents

rsides-package . . . . .	2
binary . . . . .	3
continuous . . . . .	4
Example1 . . . . .	4
Example2 . . . . .	9
Example3 . . . . .	14

GenerateReport . . . . .	20
SubgroupSearch . . . . .	22
survival . . . . .	26

<b>Index</b>	<b>27</b>
--------------	-----------

---

rsides-package	<i>Subgroup search</i>
----------------	------------------------

---

## Description

The package implements a family of subgroup search algorithms based on the SIDES (Subgroup Identification based on Differential Effect Search) method for clinical trials with normally distributed, binary and time-to-event endpoints. The package supports complex analysis models with an adjustment for continuous and categorical covariates (analysis of covariance models, logistic regression models, Cox proportional hazards models).

## Details

```

Package:  rsides
Type:    Package
Version:  0.1
Date:    2024-05-27
License:  GPL-2

```

Key functions included in the package:

- [SubgroupSearch](#): Perform a SIDES-based subgroup search.
- [GenerateReport](#): Generate a detailed summary of subgroup search results in a Microsoft Word format.

The package comes with three example data sets:

- [continuous](#): Data set based on a trial with a continuous endpoint.
- [binary](#): Data set based on a trial with a binary endpoint.
- [survival](#): Data set based on a trial with a time-to-event endpoint.

Three case studies are included in this manual to illustrate subgroup identification in clinical trials:

- [Example1](#): Subgroup search in a clinical trial with a continuous endpoint.
- [Example2](#): Subgroup search in a clinical trial with a binary endpoint.
- [Example3](#): Subgroup search in a clinical trial with a time-to-event endpoint.

## References

- Lipkovich, I., Dmitrienko, A., Denne, J., Enas, G. (2011). Subgroup Identification based on Differential Effect Search (SIDES): A recursive partitioning method for establishing response to treatment in patient subpopulations. *Statistics in Medicine*. 30, 2601-2621.
- Lipkovich, I., Dmitrienko A. (2014). Strategies for identifying predictive biomarkers and subgroups with enhanced treatment effect in clinical trials using SIDES. *Journal of Biopharmaceutical Statistics*. 24, 130-153.
- Lipkovich, I., Dmitrienko, A. (2014). Biomarker identification in clinical trials. *Clinical and Statistical Considerations in Personalized Medicine*. Carini, C., Menon, S., Chang, M. (editors). Chapman and Hall/CRC Press, New York.
- Lipkovich, I., Dmitrienko, A., D'Agostino, R.B. (2017). Tutorial in Biostatistics: Data-driven subgroup identification and analysis in clinical trials. *Statistics in Medicine*. 36, 136-196.

---

binary

*Example data set (binary endpoint)*

---

## Description

Example data set based on a trial with a binary endpoint.

## Usage

```
data(binary)
```

## Format

A data set with 359 observations and 11 variables:

outcome Binary outcome variable (0 or 1). The value of 1 indicates the beneficial outcome.

treatment Binary treatment variable (0 or 1). The values of 0 and 1 correspond to the control and experimental treatments, respectively.

cont1 Continuous covariate used in the primary analysis model.

cont2 Continuous covariate used in the primary analysis model.

class1 Class/categorical covariate used in the primary analysis model.

class2 Class/categorical covariate used in the primary analysis model.

biomarker1 Numeric biomarker.

biomarker2 Numeric biomarker.

biomarker3 Numeric biomarker.

biomarker4 Numeric biomarker.

biomarker5 Nominal biomarker.

---

 continuous

*Example data set (continuous endpoint)*


---

### Description

Example data set based on a trial with a continuous endpoint.

### Usage

```
data(continuous)
```

### Format

A data set with 359 observations and 11 variables:

`outcome` Continuous outcome variable. A larger value of the outcome variable indicates a beneficial effect.

`treatment` Binary treatment variable (0 or 1). The values of 0 and 1 correspond to the control and experimental treatments, respectively.

`cont1` Continuous covariate used in the primary analysis model.

`cont2` Continuous covariate used in the primary analysis model.

`class1` Class/categorical covariate used in the primary analysis model.

`class2` Class/categorical covariate used in the primary analysis model.

`biomarker1` Numeric biomarker.

`biomarker2` Numeric biomarker.

`biomarker3` Numeric biomarker.

`biomarker4` Numeric biomarker.

`biomarker5` Nominal biomarker.

---

 Example1

*Subgroup search in a clinical trial with a continuous endpoint*


---

### Description

Consider a clinical trial that was conducted to evaluate the efficacy and safety of an experimental treatment compared to placebo. The primary endpoint is a continuous endpoint. The trial data can be found in the [continuous](#) data set. This data set includes five biomarkers (four numeric biomarkers and one nominal biomarker) as well as several covariates that can be included in the primary analysis model. The trial's sponsor is interested in identifying a set of promising subgroups with enhanced treatment effect. Subgroup search will be performed using the basic SIDES procedure and two SIDEScreen procedures (Fixed and Adaptive SIDEScreen procedures).

The endpoint parameters will be defined as follows. First of all, the name of the outcome variable

needs to be specified, i.e., `outcome_variable = "outcome"`. Since the endpoint is continuous, the type parameter is `"continuous"`, and further the direction parameter is set to 1 since a higher value of the primary endpoint indicates beneficial effect.

Two sets of endpoint parameters will be considered to implement a simple evaluation of the treatment effect based on the two-sample t-test and a more advanced analysis based on an ANCOVA model with an adjustment for important prognostic covariates:

- Analysis strategy 1: The `analysis_method` parameter is set to `"T-test"`.
- Analysis strategy 2: The `analysis_method` parameter is set to `"ANCOVA"`. The covariates to be included in the model need to be defined using the `cont_covariates` and `class_covariates` arguments. Suppose that the ANCOVA model will account for two continuous covariates (`cont1`, `cont2`) and one class/categorical covariate (`class1`), then `cont_covariates` is set to `"cont1, cont2"` and `class_covariates` is set to `"class1"`.

The data set parameters will be specified as follows:

- The data set's name (`data_set`) is `continuous`.
- The treatment variable's name (`treatment_variable_name`) is `"treatment"`.
- The value of the treatment variable that defines the control arm (`treatment_variable_control_value`) is `"0"`.
- The list of candidate biomarkers to be included in the subgroup search (`biomarker_names`) is `c("biomarker1", "biomarker2", "biomarker3", "biomarker4", "biomarker5")`.
- The list of biomarker types (`biomarker_types`) is `c(rep("numeric", 4), "nominal")`.

Finally, the following algorithm parameters will be used in the subgroup search procedures:

- Search depth (`depth`) is 2, which means that patient subgroups will be defined in terms of one or two biomarkers. Note that this is the default value of this parameter and thus it does not need to be explicitly defined.
- Search width (`width`) is 2, i.e., only up to two best child subgroups will be retained for each parent group. This is the default value of this parameter and thus it does not need to be explicitly defined.
- Child-to-parent ratio (`gamma`) equals 1 for each search level. This is the default value of this parameter and thus it does not need to be explicitly defined.
- Minimum total number of patients in a promising subgroup (`min_subgroup_size`) is 60.
- Minimum number of unique values for continuous biomarkers for applying a percentile transformation (`nperc`) is 20, i.e., a percentile transformation will not be applied to a continuous biomarker if there are few than 20 unique values. This is the default value of this parameter and thus it does not need to be explicitly defined.
- Number of permutations to compute multiplicity-adjusted treatment effect p-values within promising subgroups (`n_perms_mult_adjust`) is 10.
- Number of processor cores to be used in parallel calculations (`ncores`) is 1.

The subgroup search will be performed using the basic SIDES procedure as well as two SIDEScreen procedures by calling the [SubgroupSearch](#) function and a comprehensive summary of subgroup search results will be generated by calling the [GenerateReport](#) function.

**See Also**

[Example2](#), [Example3](#)

**Examples**

```
#####

# Primary endpoint parameters

# Analysis strategy 1: Analysis of the continuous endpoint without
# accounting for any covariates
endpoint_parameters = list(outcome_variable = "outcome",
  type = "continuous",
  label = "Outcome",
  analysis_method = "T-test",
  direction = 1)

# Analysis strategy 2: Analysis of the continuous endpoint using an ANCOVA
# model that accounts for two continuous covariates (cont1, cont2) and
# one class/categorical covariate (class1)
endpoint_parameters = list(outcome_variable = "outcome",
  type = "continuous",
  label = "Outcome",
  analysis_method = "ANCOVA",
  cont_covariates = "cont1, cont2",
  class_covariates = "class1",
  direction = 1)

#####

# Data set parameters

# Set of candidate biomarkers
biomarker_names = c("biomarker1", "biomarker2",
  "biomarker3", "biomarker4",
  "biomarker5")

# Biomarker type
biomarker_types = c(rep("numeric", 4), "nominal")

# Data set parameters
data_set_parameters = list(data_set = continuous,
  treatment_variable_name = "treatment",
  treatment_variable_control_value = "0",
  biomarker_names = biomarker_names,
  biomarker_types = biomarker_types)

#####

# Algorithm parameters for the basic SIDES procedure

# Algorithm
```

```
subgroup_search_algorithm = "SIDES procedure"

# Number of permutations to compute multiplicity-adjusted treatment
# effect p-values within promising subgroups
n_perms_mult_adjust = 10

# Number of processor cores (use less or equal number of CPU cores on the current host)
ncores = 1

# Default values for the search depth (2), search width (2),
# maximum number of unique values for continuous biomarkers (20)

# Algorithm parameters
algorithm_parameters = list(
  n_perms_mult_adjust = n_perms_mult_adjust,
  min_subgroup_size = 60,
  subgroup_search_algorithm = subgroup_search_algorithm,
  ncores = ncores,
  random_seed = 3011)

# Perform subgroup search

# List of all parameters
parameters = list(endpoint_parameters = endpoint_parameters,
  data_set_parameters = data_set_parameters,
  algorithm_parameters = algorithm_parameters)

results = SubgroupSearch(parameters)

# Simple summary of subgroup search results
results

# Generate a detailed Word-based report with a summary of subgroup search results
report_information = GenerateReport(results,
  report_title = "Subgroup search report",
  report_filename = tempfile(
    "Continuous endpoint (SIDES).docx",
    fileext=".docx"
  )
)

#####

# Algorithm parameters for the Fixed SIDEScreen procedure

# Algorithm
subgroup_search_algorithm = "Fixed SIDEScreen procedure"

# Number of permutations to compute multiplicity-adjusted treatment
# effect p-values within promising subgroups
n_perms_mult_adjust = 10

# Number of processor cores (use less or equal number of CPU cores on the current host)
```

```

ncores = 1

# Number of biomarkers selected for the second stage in the Fixed SIDEScreen algorithm
n_top_biomarkers = 3

# Default values for the search depth (2), search width (2),
# maximum number of unique values for continuous biomarkers (20)

# Algorithm parameters
algorithm_parameters = list(
  n_perms_mult_adjust = n_perms_mult_adjust,
  min_subgroup_size = 60,
  subgroup_search_algorithm = subgroup_search_algorithm,
  ncores = ncores,
  n_top_biomarkers = n_top_biomarkers,
  random_seed = 3011)

# Perform subgroup search

# List of all parameters
parameters = list(endpoint_parameters = endpoint_parameters,
  data_set_parameters = data_set_parameters,
  algorithm_parameters = algorithm_parameters)

results = SubgroupSearch(parameters)

# Simple summary of subgroup search results
results

# Generate a detailed Word-based report with a summary of subgroup search results
report_information = GenerateReport(results,
  report_title = "Subgroup search report",
  report_filename = tempfile(
    "Continuous endpoint (Fixed SIDEScreen).docx",
    fileext=".docx"
  )
)

#####

# Algorithm parameters for the Adaptive SIDEScreen procedure

# Algorithm
subgroup_search_algorithm = "Adaptive SIDEScreen procedure"

# Number of permutations to compute multiplicity-adjusted treatment
# effect p-values within promising subgroups
n_perms_mult_adjust = 10

# Number of processor cores (use less or equal number of CPU cores on the current host)
ncores = 1

# Multiplier for selecting biomarkers for the second stage

```



```
# in the Adaptive SIDEScreen algorithm
multiplier = 1

# Number of permutations for computing the null distribution
# of the maximum VI score in the Adaptive SIDEScreen algorithm
n_perms_vi_score = 100

# Default values for the search depth (2), search width (2),
# maximum number of unique values for continuous biomarkers (20)

# Algorithm parameters
algorithm_parameters = list(
  n_perms_mult_adjust = n_perms_mult_adjust,
  min_subgroup_size = 60,
  subgroup_search_algorithm = subgroup_search_algorithm,
  ncores = ncores,
  multiplier = multiplier,
  n_perms_vi_score = n_perms_vi_score,
  random_seed = 3011)

# Perform subgroup search

# List of all parameters
parameters = list(endpoint_parameters = endpoint_parameters,
  data_set_parameters = data_set_parameters,
  algorithm_parameters = algorithm_parameters)

results = SubgroupSearch(parameters)

# Simple summary of subgroup search results
results

# Generate a detailed Word-based report with a summary of subgroup search results
GenerateReport(results,
  report_title = "Subgroup search report",
  report_filename = tempfile(
    "Continuous endpoint (Adaptive SIDEScreen).docx",
    fileext=".docx"
  )
)
```

---

Example2

*Subgroup search in a clinical trial with a binary endpoint*

---

### **Description**

Consider a clinical trial that was conducted to evaluate the efficacy and safety of an experimental treatment compared to placebo using a binary primary endpoint. The endpoint assumes the values of 0 and 1 (1 corresponds to the desirable outcome). The trial's data set (see the [binary](#) data set)

includes five biomarkers (four numeric biomarkers and one nominal biomarker) as well as several covariates that can be included in the primary analysis model. The trial's sponsor is interested in identifying a set of promising subgroups with enhanced treatment effect. Subgroup search will be performed using the basic SIDES procedure and two SIDEScreen procedures (Fixed and Adaptive SIDEScreen procedures).

The endpoint parameters will be defined as follows. The name of the outcome variable is specified using `outcome_variable = "outcome"` and, since the endpoint is binary, the type parameter is set to `"binary"`. The desirable outcome for the endpoint is 1, which means that the direction parameter is set to 1 (a higher value of the endpoint indicates beneficial effect).

Two sets of endpoint parameters will be defined to implement a simple evaluation of the treatment effect based on the Z-test for proportions as well as a more advanced analysis based on a logistic regression model with an adjustment for important prognostic covariates:

- Analysis strategy 1: The `analysis_method` parameter is set to `"Z-test for proportions"`.
- Analysis strategy 2: The `analysis_method` parameter is set to `"Logistic regression"`. The covariates to be included in the logistic model need to be defined using the `cont_covariates` and `class_covariates` arguments. To adjust for two continuous covariates (`cont1`, `cont2`) and two class/categorical covariate (`class1`, `class2`), `cont_covariates` is set to `"cont1, cont2"` and `class_covariates` is set to `"class1, class2"`.

The data set parameters will be specified as follows:

- The data set's name (`data_set`) is `binary`.
- The treatment variable's name (`treatment_variable_name`) is `"treatment"`.
- The value of the treatment variable that defines the control arm (`treatment_variable_control_value`) is `"0"`.
- The list of candidate biomarkers to be included in the subgroup search (`biomarker_names`) is `c("biomarker1", "biomarker2", "biomarker3", "biomarker4", "biomarker5")`.
- The list of biomarker types (`biomarker_types`) is `c(rep("numeric", 4), "nominal")`.

The following algorithm parameters will be used in the subgroup search procedures:

- Search depth (`depth`) is 2, which means that patient subgroups will be defined in terms of one or two biomarkers. Note that this is the default value of this parameter and thus it does not need to be explicitly defined.
- Search width (`width`) is 2, i.e., only up to two best child subgroups will be retained for each parent group. This is the default value of this parameter and thus it does not need to be explicitly defined.
- Child-to-parent ratio (`gamma`) equals 1 for each search level. This is the default value of this parameter and thus it does not need to be explicitly defined.
- Minimum total number of patients in a promising subgroup (`min_subgroup_size`) is 60.
- Minimum number of unique values for continuous biomarkers for applying a percentile transformation (`nperc`) is 20, i.e., a percentile transformation will not be applied to a continuous biomarker if there are few than 20 unique values. This is the default value of this parameter and thus it does not need to be explicitly defined.
- Number of permutations to compute multiplicity-adjusted treatment effect p-values within promising subgroups (`n_perms_mult_adjust`) is 10.

- Number of processor cores to be used in parallel calculations (ncores) is 1.

The subgroup search will be performed using the basic SIDES procedure as well as two SIDEScreen procedures by calling the [SubgroupSearch](#) function and a comprehensive summary of subgroup search results will be generated by calling the [GenerateReport](#) function.

### See Also

[Example1](#), [Example3](#)

### Examples

```
#####
# Primary endpoint parameters

# Analysis strategy 1: Analysis of the binary endpoint without
# accounting for any covariates
endpoint_parameters = list(outcome_variable = "outcome",
  type = "binary",
  label = "Outcome",
  analysis_method = "Z-test for proportions",
  direction = 1)

# Analysis strategy 2: Analysis of the continuous endpoint using an ANCOVA
# model that accounts for two continuous covariates (cont1, cont2) and
# two class/categorical covariates (class1, class2)
endpoint_parameters = list(outcome_variable = "outcome",
  type = "binary",
  label = "Outcome",
  analysis_method = "Logistic regression",
  cont_covariates = "cont1, cont2",
  class_covariates = "class1, class2",
  direction = 1)

#####
# Data set parameters

# Set of candidate biomarkers
biomarker_names = c("biomarker1", "biomarker2",
  "biomarker3", "biomarker4",
  "biomarker5")

# Biomarker type
biomarker_types = c(rep("numeric", 4), "nominal")

# Data set parameters
data_set_parameters = list(data_set = binary,
  treatment_variable_name = "treatment",
  treatment_variable_control_value = "0",
  biomarker_names = biomarker_names,
  biomarker_types = biomarker_types)
```

```
#####

# Algorithm parameters for the basic SIDES procedure

# Algorithm
subgroup_search_algorithm = "SIDES procedure"

# Number of permutations to compute multiplicity-adjusted treatment
# effect p-values within promising subgroups
n_perms_mult_adjust = 10

# Number of processor cores (use less or equal number of CPU cores on the current host)
ncores = 1

# Default values for the search depth (2), search width (2),
# maximum number of unique values for continuous biomarkers (20)

# Algorithm parameters
algorithm_parameters = list(
  n_perms_mult_adjust = n_perms_mult_adjust,
  min_subgroup_size = 60,
  subgroup_search_algorithm = subgroup_search_algorithm,
  ncores = ncores,
  random_seed = 3011)

# Perform subgroup search

# List of all parameters
parameters = list(endpoint_parameters = endpoint_parameters,
  data_set_parameters = data_set_parameters,
  algorithm_parameters = algorithm_parameters)

results = SubgroupSearch(parameters)

# Simple summary of subgroup search results
results

# Generate a detailed Word-based report with a summary of subgroup search results
report_information = GenerateReport(results,
  report_title = "Subgroup search report",
  report_filename = tempfile(
    "Binary endpoint (SIDES).docx",
    fileext=".docx"
  )
)

#####

# Algorithm parameters for the Fixed SIDEScreen procedure

# Algorithm
subgroup_search_algorithm = "Fixed SIDEScreen procedure"
```

```

# Number of permutations to compute multiplicity-adjusted treatment
# effect p-values within promising subgroups
n_perms_mult_adjust = 10

# Number of processor cores (use less or equal number of CPU cores on the current host)
ncores = 1

# Number of biomarkers selected for the second stage in the Fixed SIDEScreen algorithm
n_top_biomarkers = 3

# Default values for the search depth (2), search width (2),
# maximum number of unique values for continuous biomarkers (20)

# Algorithm parameters
algorithm_parameters = list(
  n_perms_mult_adjust = n_perms_mult_adjust,
  min_subgroup_size = 60,
  subgroup_search_algorithm = subgroup_search_algorithm,
  ncores = ncores,
  n_top_biomarkers = n_top_biomarkers,
  random_seed = 3011)

# Perform subgroup search

# List of all parameters
parameters = list(endpoint_parameters = endpoint_parameters,
  data_set_parameters = data_set_parameters,
  algorithm_parameters = algorithm_parameters)

results = SubgroupSearch(parameters)

# Simple summary of subgroup search results
results

# Generate a detailed Word-based report with a summary of subgroup search results
report_information = GenerateReport(results,
  report_title = "Subgroup search report",
  report_filename = tempfile(
    "Binary endpoint (Fixed SIDEScreen).docx",
    fileext=".docx"
  )
)

#####

# Algorithm parameters for the Adaptive SIDEScreen procedure

# Algorithm
subgroup_search_algorithm = "Adaptive SIDEScreen procedure"

# Number of permutations to compute multiplicity-adjusted treatment
# effect p-values within promising subgroups

```

```

n_perms_mult_adjust = 10

# Number of processor cores (use less or equal number
# of CPU cores on the current host)
ncores = 1

# Multiplier for selecting biomarkers for the second stage
# in the Adaptive SIDEScreen algorithm
multiplier = 1

# Number of permutations for computing the null distribution
# of the maximum VI score in the Adaptive SIDEScreen algorithm
n_perms_vi_score = 100

# Default values for the search depth (2), search width (2),
# maximum number of unique values for continuous biomarkers (20)

# Algorithm parameters
algorithm_parameters = list(
  n_perms_mult_adjust = n_perms_mult_adjust,
  min_subgroup_size = 60,
  subgroup_search_algorithm = subgroup_search_algorithm,
  ncores = ncores,
  multiplier = multiplier,
  n_perms_vi_score = n_perms_vi_score,
  random_seed = 3011)

# Perform subgroup search

# List of all parameters
parameters = list(endpoint_parameters = endpoint_parameters,
  data_set_parameters = data_set_parameters,
  algorithm_parameters = algorithm_parameters)

results = SubgroupSearch(parameters)

# Simple summary of subgroup search results
results

# Generate a detailed Word-based report with a summary of subgroup search results
GenerateReport(results,
  report_title = "Subgroup search report",
  report_filename = tempfile(
    "Binary endpoint (Adaptive SIDEScreen).docx",
    fileext=".docx"
  )
)

```

### Description

Consider a clinical trial that was conducted to evaluate the efficacy and safety of an experimental treatment compared to a control treatment with a time-to-event primary endpoint (see the [survival](#) data set). A larger value of this endpoint indicates beneficial effect, e.g., longer survival. As in the other two examples, the trial's data set includes five biomarkers (four numeric biomarkers and one nominal biomarker) in addition to several covariates that can be included in the primary analysis model. The trial's sponsor is interested in identifying a set of promising subgroups with enhanced treatment effect. Subgroup search will be performed using the basic SIDES procedure and two SIDEScreen procedures (Fixed and Adaptive SIDEScreen procedures).

The endpoint parameters will be defined as follows. The endpoint measures the time to an event of interest and thus the type parameter is set to "survival". The names of the outcome and censoring variables need to be defined using `outcome_variable` and `outcome_censor_variable`, e.g., `outcome_variable = "outcome"` and `outcome_censor_variable = "outcome_censor"`. The value of the censoring variable that corresponds to a censored observation is defined using `outcome_censor_value`, e.g., `outcome_censor_value = "1"` if the value of 1 indicates that the patient did not experience the event of interest by the end of the treatment period. Since a larger value of the primary endpoint is associated with beneficial effect, the `direction` parameter is set to 1.

Two sets of endpoint parameters will be defined to implement a simple assessment of the treatment effect based on the log-rank test as well as an advanced analysis based on a Cox proportional hazards regression model with an adjustment for important prognostic covariates:

- Analysis strategy 1: The `analysis_method` parameter is set to "Log-rank test".
- Analysis strategy 2: The `analysis_method` parameter is set to "Cox regression". The covariates to be included in this model need to be defined using the `cont_covariates` and `class_covariates` arguments. To adjust for two continuous covariates (`cont1`, `cont2`) and two class/categorical covariates (`class1`, `class2`), `cont_covariates` is set to "`cont1`, `cont2`" and `class_covariates` is set to "`class1`, `class2`".

The data set parameters will be specified as follows:

- The data set's name (`data_set`) is `survival`.
- The treatment variable's name (`treatment_variable_name`) is "`treatment`".
- The value of the treatment variable that defines the control arm (`treatment_variable_control_value`) is "`0`".
- The list of candidate biomarkers to be included in the subgroup search (`biomarker_names`) is `c("biomarker1", "biomarker2", "biomarker3", "biomarker4", "biomarker5")`.
- The list of biomarker types (`biomarker_types`) is `c(rep("numeric", 4), "nominal")`.

The following algorithm parameters will be used in the subgroup search procedures:

- Search depth (`depth`) is 2, which means that patient subgroups will be defined in terms of one or two biomarkers. Note that this is the default value of this parameter and thus it does not need to be explicitly defined.
- Search width (`width`) is 2, i.e., only up to two best child subgroups will be retained for each parent group. This is the default value of this parameter and thus it does not need to be explicitly defined.
- Child-to-parent ratio (`gamma`) equals 1 for each search level. This is the default value of this parameter and thus it does not need to be explicitly defined.

- Minimum total number of patients in a promising subgroup (`min_subgroup_size`) is 60.
- Minimum number of unique values for continuous biomarkers for applying a percentile transformation (`nperc`) is 20, i.e., a percentile transformation will not be applied to a continuous biomarker if there are few than 20 unique values. This is the default value of this parameter and thus it does not need to be explicitly defined.
- Number of permutations to compute multiplicity-adjusted treatment effect p-values within promising subgroups (`n_perms_mult_adjust`) is 10.
- Number of processor cores to be used in parallel calculations (`ncores`) is 1.

The subgroup search will be performed using the basic SIDES procedure as well as two SIDEScreen procedures by calling the [SubgroupSearch](#) function and a comprehensive summary of subgroup search results will be generated by calling the [GenerateReport](#) function.

### See Also

[Example1](#), [Example2](#)

### Examples

```
#####

# Primary endpoint parameters

# Analysis strategy 1: Analysis of the continuous endpoint without
# accounting for any covariates
endpoint_parameters = list(outcome_variable = "outcome",
  outcome_censor_variable = "outcome_censor",
  outcome_censor_value = "1",
  type = "survival",
  label = "Outcome",
  analysis_method = "Log-rank test",
  direction = 1)

# Analysis strategy 2: Analysis of the continuous endpoint using a Cox model
# that accounts for two continuous covariates (cont1, cont2) and
# two class/categorical covariates (class1, class2)
endpoint_parameters = list(outcome_variable = "outcome",
  outcome_censor_variable = "outcome_censor",
  outcome_censor_value = "1",
  type = "survival",
  label = "Outcome",
  analysis_method = "Cox regression",
  cont_covariates = "cont1, cont2",
  class_covariates = "class1, class2",
  direction = 1)

#####

# Data set parameters

# Set of candidate biomarkers
```



```

biomarker_names = c("biomarker1", "biomarker2",
                    "biomarker3", "biomarker4",
                    "biomarker5")

# Biomarker type
biomarker_types = c(rep("numeric", 4), "nominal")

# Data set parameters
data_set_parameters = list(data_set = survival,
                           treatment_variable_name = "treatment",
                           treatment_variable_control_value = "0",
                           biomarker_names = biomarker_names,
                           biomarker_types = biomarker_types)

#####

# Algorithm parameters for the basic SIDES procedure

# Algorithm
subgroup_search_algorithm = "SIDES procedure"

# Number of permutations to compute multiplicity-adjusted treatment
# effect p-values within promising subgroups
n_perms_mult_adjust = 10

# Number of processor cores (use less or equal number of CPU cores on the current host)
ncores = 1

# Default values for the search depth (2), search width (2),
# maximum number of unique values for continuous biomarkers (20)

# Algorithm parameters
algorithm_parameters = list(
  n_perms_mult_adjust = n_perms_mult_adjust,
  min_subgroup_size = 60,
  subgroup_search_algorithm = subgroup_search_algorithm,
  ncores = ncores,
  random_seed = 3011)

# Perform subgroup search

# List of all parameters
parameters = list(endpoint_parameters = endpoint_parameters,
                  data_set_parameters = data_set_parameters,
                  algorithm_parameters = algorithm_parameters)

results = SubgroupSearch(parameters)

# Simple summary of subgroup search results
results

# Generate a detailed Word-based report with a summary of subgroup search results
report_information = GenerateReport(results,

```

```

report_title = "Subgroup search report",
report_filename = tempfile(
  "Time-to-event endpoint (SIDES).docx",
  fileext=".docx"
)
)

#####

# Algorithm parameters for the Fixed SIDEScreen procedure

# Algorithm
subgroup_search_algorithm = "Fixed SIDEScreen procedure"

# Number of permutations to compute multiplicity-adjusted treatment
# effect p-values within promising subgroups
n_perms_mult_adjust = 10

# Number of processor cores (use less or equal number of CPU cores on the current host)
ncores = 1

# Number of biomarkers selected for the second stage in the Fixed SIDEScreen algorithm
n_top_biomarkers = 3

# Default values for the search depth (2), search width (2),
# maximum number of unique values for continuous biomarkers (20)

# Algorithm parameters
algorithm_parameters = list(
  n_perms_mult_adjust = n_perms_mult_adjust,
  min_subgroup_size = 60,
  subgroup_search_algorithm = subgroup_search_algorithm,
  ncores = ncores,
  n_top_biomarkers = n_top_biomarkers,
  random_seed = 3011)

# Perform subgroup search

# List of all parameters
parameters = list(endpoint_parameters = endpoint_parameters,
  data_set_parameters = data_set_parameters,
  algorithm_parameters = algorithm_parameters)

results = SubgroupSearch(parameters)

# Simple summary of subgroup search results
results

# Generate a detailed Word-based report with a summary of subgroup search results
report_information = GenerateReport(results,
  report_title = "Subgroup search report",
  report_filename = tempfile(
    "Time-to-event endpoint (Fixed SIDEScreen).docx",

```

```

        fileext=".docx"
    )
)

#####

# Algorithm parameters for the Adaptive SIDEScreen procedure

# Algorithm
subgroup_search_algorithm = "Adaptive SIDEScreen procedure"

# Number of permutations to compute multiplicity-adjusted treatment
# effect p-values within promising subgroups
n_perms_mult_adjust = 10

# Number of processor cores (use less or equal number of CPU cores on the current host)
ncores = 1

# Multiplier for selecting biomarkers for the second stage
# in the Adaptive SIDEScreen algorithm
multiplier = 1

# Number of permutations for computing the null distribution
# of the maximum VI score in the Adaptive SIDEScreen algorithm
n_perms_vi_score = 100

# Default values for the search depth (2), search width (2),
# maximum number of unique values for continuous biomarkers (20)

# Algorithm parameters
algorithm_parameters = list(
  n_perms_mult_adjust = n_perms_mult_adjust,
  min_subgroup_size = 60,
  subgroup_search_algorithm = subgroup_search_algorithm,
  ncores = ncores,
  multiplier = multiplier,
  n_perms_vi_score = n_perms_vi_score,
  random_seed = 3011)

# Perform subgroup search

# List of all parameters
parameters = list(endpoint_parameters = endpoint_parameters,
  data_set_parameters = data_set_parameters,
  algorithm_parameters = algorithm_parameters)

results = SubgroupSearch(parameters)

# Simple summary of subgroup search results
results

# Generate a detailed Word-based report with a summary of subgroup search results
GenerateReport(results,

```

```

report_title = "Subgroup search report",
report_filename = tempfile(
  "Time-to-event endpoint (Adaptive SIDEScreen).docx",
  fileext=".docx"
)
)

```

---

GenerateReport

*Generate a Word-based summary of subgroup search results*


---

## Description

This function creates a detailed summary of subgroup search results in a Microsoft Word format.

## Usage

```
GenerateReport(results, report_title, report_filename)
```

## Arguments

results	Object of class SubgroupSearchResults created by the SubgroupSearch function.
report_title	Character value defining the report's title.
report_filename	Character value defining the report's filename. The report is saved in the current working directory.

## Value

No return value, called for side effects

## See Also

[SubgroupSearch](#)

## Examples

```

# Example of a subgroup search in a trial with a binary endpoint
# This example used an example data set (binary) that comes with the package
# Primary endpoint parameters

```

```

endpoint_parameters = list(outcome_variable = "outcome",
                           type = "binary",
                           label = "Outcome",
                           analysis_method = "Z-test for proportions",
                           direction = 1)

```

```
#####
```

```
# Data set parameters

# Set of candidate biomarkers
biomarker_names = c("biomarker1", "biomarker2",
                    "biomarker3", "biomarker4",
                    "biomarker5")

# Biomarker type
biomarker_types = c(rep("numeric", 4), "nominal")

# Data set parameters
data_set_parameters = list(data_set = binary,
                           treatment_variable_name = "treatment",
                           treatment_variable_control_value = "0",
                           biomarker_names = biomarker_names,
                           biomarker_types = biomarker_types)

#####

# Algorithm parameters for the basic SIDES procedure

# Algorithm
subgroup_search_algorithm = "SIDES procedure"

# Number of permutations to compute multiplicity-adjusted treatment
# effect p-values within promising subgroups
n_perms_mult_adjust = 10

# Number of processor cores (use less or equal number of CPU cores on the current host)
ncores = 1

# Default values for the search depth (2), search width (2),
# maximum number of unique values for continuous biomarkers (20)

# Algorithm parameters
algorithm_parameters = list(
  n_perms_mult_adjust = n_perms_mult_adjust,
  min_subgroup_size = 60,
  subgroup_search_algorithm = subgroup_search_algorithm,
  ncores = ncores,
  random_seed = 3011)

# Perform subgroup search

# List of all parameters
parameters = list(endpoint_parameters = endpoint_parameters,
                  data_set_parameters = data_set_parameters,
                  algorithm_parameters = algorithm_parameters)

# Perform subgroup search
results = SubgroupSearch(parameters)
```

```

# Simple summary of subgroup search results
results

# Generate a detailed Word-based report with a summary of subgroup search results
GenerateReport(
  results,
  report_title = "Subgroup search report",
  report_filename = tempfile(
    "Subgroup search report.docx",
    fileext=".docx"
  )
)

```

---

SubgroupSearch

*Perform a SIDES-based subgroup search*


---

### Description

This function performs a SIDES-based subgroup search for clinical trials with normally distributed, binary and time-to-event endpoints. The function implements the following subgroup search procedures:

- SIDES procedure: Basic subgroup search procedure (Lipkovich et al., 2011).
- Fixed and Adaptive SIDEScreen procedures: Two-stage subgroup search procedure with biomarker selection (Lipkovich and Dmitrienko, 2014).

### Usage

```
SubgroupSearch(parameters)
```

### Arguments

- |            |  |
|------------|--|
| parameters | <p>List defining the subgroup search's parameters. The list includes three sublists: <code>endpoint_parameters</code>, <code>data_set_parameters</code>, <code>algorithm_parameters</code>. The parameters that need to be defined with each of these lists are defined below.</p> <ul style="list-style-type: none"> <li>• <code>endpoint_parameters</code>: List defining the parameters of the primary outcome variable and analysis method. The following parameters need to be specified: <ul style="list-style-type: none"> <li>– <code>outcome_variable</code>: Character value defining the name of the outcome variable in the data set specified in <code>data_set_parameters</code>.</li> <li>– <code>label</code>: Character value defining the outcome variable's label.</li> <li>– <code>type</code>: Character value defining the outcome variable's type: "continuous" if the outcome variable is a continuous endpoint, "binary" if the outcome variable is a binary endpoint, "survival" if the outcome variable is a time-to-event endpoint.</li> </ul> </li> </ul> |
|------------|--|

- `outcome_censor_variable`: Character value defining the name of the censoring variable in the data set. This argument is required only if the outcome variable is a time-to-event endpoint.
- `outcome_censor_value`: Character value defining the value of the censoring variable that corresponds to censored outcomes. This argument is required only if the outcome variable is a time-to-event endpoint.
- `direction`: Numeric value defining the direction of a beneficial effect: 1 if a higher value of the outcome variable indicates a beneficial effect, -1 if a lower value of the outcome variable indicates a beneficial effect.
- `analysis_method`: Character value defining the analysis method for the outcome variable:  
 "T-test", "Z-test for proportions" or "Log-rank test" for continuous, binary and time-to-event endpoint without covariate adjustment, respectively,  
 ("ANCOVA", "Logistic regression" or "Cox regression" for continuous, binary and time-to-event endpoint with covariate adjustment, respectively. The covariates to be included in the model are specified using `cont_covariates` and `class_covariates`.
- `cont_covariates`: Vector of character values defining the names of the continuous covariates to be included in the model if `analysis_method` is set to "ANCOVA", "Logistic regression" or "Cox regression". This argument is not required if `analysis_method` is set to "T-test", "Z-test for proportions" or "Log-rank test".
- `class_covariates`: Vector of character values defining the names of the class/categorical covariates to be included in the model if `analysis_method` is set to "ANCOVA", "Logistic regression" or "Cox regression". This argument is not required if `analysis_method` is set to "T-test", "Z-test for proportions" or "Log-rank test".
- `data_set_parameters`: List defining the data set and its characteristics. The following parameters need to be specified:
  - `data_set`: Character value defining the name of the clinical trial data set. The package comes with three data sets that are used in the examples:  
`continuous`: Data set based on a trial with a continuous endpoint.  
`binary`: Data set based on a trial with a binary endpoint.  
`survival`: Data set based on a trial with a time-to-event endpoint.
  - `treatment_variable_name`: Character value defining the name of the treatment variable in the data set. Only two-arm trials are supported.
  - `treatment_variable_control_value`: Character value defining the value of the treatment variable that corresponds to the control arm.
  - `biomarker_names`: Vector of character values defining the names of the candidate biomarkers.
  - `biomarker_types`: Vector of character values defining the types of the candidate biomarkers:

- "numeric" if the biomarker is a continuous variable,
  - "nominal" if the biomarker is a nominal variable.
- biomarker\_levels: Vector of numeric values defining the first subgroup search level at which each biomarker is introduced. For example, if the level is 1, the biomarker will be used in all subgroups and, if the level is 2, the biomarker will be used only in the second-level and deeper subgroups. By default, the level is set to 1 for each biomarker.
- algorithm\_parameters: List of the subgroup search algorithm's parameters. The following parameters need to be specified:
  - subgroup\_search\_algorithm: Character value defining the name of the subgroup search algorithm:
    - "SIDES procedure": Basic subgroup search procedure,
    - "Fixed SIDEScreen procedure": SIDEScreen procedure with a fixed number of biomarkers selected for the second stage,
    - "Adaptive SIDEScreen procedure": SIDEScreen procedure with a data-driven number of biomarkers selected for the second stage.
  - depth: Integer value defining the subgroup search depth. The default value is 2.
  - width: Integer value defining the subgroup search width. The default value is 2.
  - gamma: Vector of numeric values defining the complexity parameters (also known as the child-to-parent ratios). The complexity parameters must be between 0 and 1 (unless no complexity control is applied at a certain search level in which case the complexity parameter at this level is set to NA) and the vector's length must be equal to the search depth. The default value is 1 at each search level, i.e., by default gamma is equal to  $c(1, 1)$  if the depth parameter is set to 2.
  - n\_perms\_mult\_adjust: Integer value defining the number of permutations for computing multiplicity-adjusted treatment effect p-values within the promising subgroups. The default value is 1000.
  - ncores: Integer value defining the number of processor cores that will be used for computing multiplicity-adjusted treatment effect p-values. The default value is 1.
  - nperc: Integer value defining the minimum number of unique values for continuous biomarkers for applying a percentile transformation. The default value is 20.
  - min\_subgroup\_size: Integer value defining the minimum total number of patients in a promising subgroup. The default value is 30.
  - n\_top\_biomarkers: Integer value defining the number of best biomarkers selected for the second stage of the SIDEScreen procedure. This argument is only required for the Fixed SIDEScreen procedure. The default value is 3.
  - multiplier: Numeric value defining the multiplier in the data-driven biomarker selection rule for the second stage of the SIDEScreen procedure. This argument is only required for the Adaptive SIDEScreen procedure. The default value is 1.



- `n_perms_vi_score`: Numeric value defining the number of permutations used in the data-driven biomarker selection rule for the second stage of the SIDEScreen procedure. This argument is only required for the Adaptive SIDEScreen procedure. The default value is 100.
- `random_seed`: Integer value defining the random seed that will be used for computing permutation-based multiplicity-adjusted treatment effect p-values. The default value is 49291.

## Value

The function returns an object of class ‘SubgroupSearchResults’. This object is a list with the following components:

`parameters` a list containing the user-specified parameters, i.e., endpoint, data set and algorithm parameters.

`patient_subgroups` a list containing the subgroup search results, in particular, a summary of the subgroup effects, a variable importance summary and a brief summary of the algorithm’s parameters. The summary of subgroup effects provides information on the treatment effect in the overall population and promising subgroups identified by the selected algorithm. The summary includes the number of patients in each subgroup by trial arm, treatment effect estimate as well as raw and multiplicity-adjusted p-values. For a continuous primary endpoint, the treatment effect estimate is defined as the sample mean difference or the mean difference computed from the ANCOVA model. For a binary primary endpoint, the treatment effect estimate is defined as the sample difference in proportions if the Z-test for proportions is carried out or the odds ratio computed from the logistic regression model. Finally, if the primary endpoint is a time-to-event endpoint, the treatment effect estimate is defined as the hazard ratio based on an exponential distribution assumption if the analysis is based on the log-rank test or the hazard ratio computed from the Cox proportional hazards model if a model-based analysis is employed.

A detailed summary of the subgroup search results can be generated using the [GenerateReport](#) function.

## References

Lipkovich, I., Dmitrienko, A., Denne, J., Enas, G. (2011). Subgroup Identification based on Differential Effect Search (SIDES): A recursive partitioning method for establishing response to treatment in patient subpopulations. *Statistics in Medicine*. 30, 2601-2621.

Lipkovich, I., Dmitrienko A. (2014). Strategies for identifying predictive biomarkers and subgroups with enhanced treatment effect in clinical trials using SIDES. *Journal of Biopharmaceutical Statistics*. 24, 130-153.

Lipkovich, I., Dmitrienko, A., D’Agostino, R.B. (2017). Tutorial in biostatistics: data-driven subgroup identification and analysis in clinical trials. *Statistics in Medicine*.36, 136-196.

**See Also**[GenerateReport](#)

---

`survival`*Example data set (time-to-event endpoint)*

---

**Description**

Example data set based on a trial with a time-to-event endpoint.

**Usage**

```
data(survival)
```

**Format**

A data set with 359 observations and 12 variables:

`outcome` Time-to-event outcome variable. A larger value of the outcome variable (longer survival) indicates a beneficial effect.

`outcome_censor` Binary censoring variable (0 or 1). The value of 1 indicates censoring.

`treatment` Binary treatment variable (0 or 1). The values of 0 and 1 correspond to the control and experimental treatments, respectively.

`cont1` Continuous covariate used in the primary analysis model.

`cont2` Continuous covariate used in the primary analysis model.

`class1` Class/categorical covariate used in the primary analysis model.

`class2` Class/categorical covariate used in the primary analysis model.

`biomarker1` Numeric biomarker.

`biomarker2` Numeric biomarker.

`biomarker3` Numeric biomarker.

`biomarker4` Numeric biomarker.

`biomarker5` Nominal biomarker.

# Index

## \* datasets

- binary, [3](#)
- continuous, [4](#)
- rsides-package, [2](#)
- survival, [26](#)

binary, [2](#), [3](#), [9](#), [23](#)

continuous, [2](#), [4](#), [4](#), [23](#)

Example1, [2](#), [4](#), [11](#), [16](#)

Example2, [2](#), [6](#), [9](#), [16](#)

Example3, [2](#), [6](#), [11](#), [14](#)

GenerateReport, [2](#), [5](#), [11](#), [16](#), [20](#), [25](#), [26](#)

print.SubgroupSearchResults  
(SubgroupSearch), [22](#)

rsides (rsides-package), [2](#)

rsides-package, [2](#)

SubgroupSearch, [2](#), [5](#), [11](#), [16](#), [20](#), [22](#)

survival, [2](#), [15](#), [23](#), [26](#)