

# Package: ISS (via r-universe)

September 15, 2024

**Type** Package

**Title** Isotonic Subgroup Selection

**Version** 1.0.0

**Description** Methodology for subgroup selection in the context of isotonic regression including methods for sub-Gaussian errors, classification, homoscedastic Gaussian errors and quantile regression. See the documentation of ISS(). Details can be found in the paper by Müller, Reeve, Cannings and Samworth (2023) <[arXiv:2305.04852v2](https://arxiv.org/abs/2305.04852v2)>.

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Imports** parallel, stats, Rdpack (>= 0.7)

**RdMacros** Rdpack

**NeedsCompilation** no

**Author** Manuel M. Müller [aut, cre], Henry W. J. Reeve [aut], Timothy I. Cannings [aut], Richard J. Samworth [aut]

**Maintainer** Manuel M. Müller <[mm2559@cam.ac.uk](mailto:mm2559@cam.ac.uk)>

**Date/Publication** 2023-07-06 22:10:02 UTC

**Repository** <https://manuelmueller.r-universe.dev>

**RemoteUrl** <https://github.com/cran/ISS>

**RemoteRef** HEAD

**RemoteSha** b8969c05893c062ed062296e300aefdb2bf603c4

## Contents

dag_test_FS . . . . .	2
dag_test_Holm . . . . .	3
dag_test_ISS . . . . .	3
dag_test_MG . . . . .	4

get_boundary_points . . . . .	5
get_DAG . . . . .	6
get_p_classification . . . . .	7
get_p_Gaussian . . . . .	8
get_p_subGaussian . . . . .	9
get_p_subGaussian_NM . . . . .	10
get_p_value . . . . .	11
ISS . . . . .	12

<b>Index</b>	<b>15</b>
--------------	-----------

---

dag_test_FS	<i>dag_test_FS</i>
-------------	--------------------

---

## Description

Implements the fixed sequence testing procedure of familywise error rate control. The sequence is given through ordering elements of `p_order` increasingly.

## Usage

```
dag_test_FS(p_order, p, alpha, decreasing = FALSE)
```

## Arguments

<code>p_order</code>	a numeric vector or matrix with one column whose order determines the sequence of tests.
<code>p</code>	a numeric vector taking values in (0, 1] such that <code>length(p) == nrow(p_order)</code> if <code>p_order</code> is a matrix (or <code>length(p) == length(p_order)</code> if <code>p_order</code> is a numeric vector).
<code>alpha</code>	a numeric value in (0, 1] specifying the Type I error rate.
<code>decreasing</code>	a boolean value determining whether the order of <code>p_order</code> should be understood in decreasing order.

## Value

A boolean vector of the same length as `p` with each element being TRUE if the corresponding hypothesis is rejected and FALSE otherwise.

## Examples

```
p_order <- c(0.5, 0, 1)
p <- c(0.01, 0.1, 0.05)
alpha <- 0.05
dag_test_FS(p_order, p, alpha, decreasing = TRUE)
```

---

dag_test_Holm	<i>dag_test_Holm</i>
---------------	----------------------

---

**Description**

Given a vector of p-values, each concerning a row in the matrix  $X_0$ , `dag_test_Holm()` first applies Holm's method to the p-values and then also rejects hypotheses corresponding to points coordinate-wise greater or equal to any point whose hypothesis has been rejected.

**Usage**

```
dag_test_Holm(X0, p, alpha)
```

**Arguments**

$X_0$	a numeric matrix giving points corresponding to hypotheses.
$p$	a numeric vector taking values in $(0, 1]$ such that <code>length(p) == nrow(X0)</code> .
$\alpha$	a numeric value in $(0, 1]$ specifying the Type I error rate.

**Value**

A boolean vector of the same length as  $p$  with each element being TRUE if the corresponding hypothesis is rejected and FALSE otherwise.

**Examples**

```
X0 <- rbind(c(0.5, 0.5), c(0.8, 0.9), c(0.4, 0.6))
p <- c(0.01, 0.1, 0.05)
alpha <- 0.05
dag_test_Holm(X0, p, alpha)
```

---

dag_test_ISS	<i>dag_test_ISS</i>
--------------	---------------------

---

**Description**

Implements the DAG testing procedure given in Algorithm 1 by Müller et al. (2023).

**Usage**

```
dag_test_ISS(X0, p, alpha)
```

**Arguments**

`X0` a numeric matrix giving points corresponding to hypotheses.  
`p` a numeric vector taking values in (0, 1] such that `length(p) == nrow(X0)`.  
`alpha` a numeric value in (0, 1] specifying the Type I error rate.

**Value**

A boolean vector of the same length as `p` with each element being TRUE if the corresponding hypothesis is rejected and FALSE otherwise.

**References**

Müller MM, Reeve HWJ, Cannings TI, Samworth RJ (2023). "Isotonic subgroup selection." *arXiv preprint arXiv:2305.04852*.

**Examples**

```
X0 <- rbind(c(0.5, 0.6), c(0.8, 0.9), c(0.9, 0.8))
p <- c(0.02, 0.025, 0.1)
alpha <- 0.05
dag_test_ISS(X0, p, alpha)
```

---

dag\_test\_MG

*dag\_test\_MG*

---

**Description**

Implements the graph-testing procedures proposed by Meijer and Goeman (2015) for one-way logical relationships. Here implemented for the specific application to isotonic subgroup selection.

**Usage**

```
dag_test_MG(
  X0,
  p,
  alpha,
  version = c("all", "any"),
  leaf_weights,
  sparse = FALSE
)
```

**Arguments**

<code>X0</code>	a numeric matrix giving points corresponding to hypotheses.
<code>p</code>	a numeric vector taking values in (0, 1] such that <code>length(p) == nrow(X0)</code> .
<code>alpha</code>	a numeric value in (0, 1] specifying the Type I error rate.
<code>version</code>	either "all" for the all-parent version of the procedure or "any" for the any-parent version of the procedure.
<code>leaf_weights</code>	optional weights for the leaf nodes. Would have to be a numeric vector of the same length as there are leaf nodes in the DAG (resp. polytree, see <code>sparse</code> ) induced by <code>X0</code> .
<code>sparse</code>	a logical value specifying whether <code>X0</code> should be used to induce a DAG (FALSE) or a polytree (TRUE).

**Value**

A boolean vector of the same length as `p` with each element being TRUE if the corresponding hypothesis is rejected and FALSE otherwise.

**References**

Meijer RJ, Goeman JJ (2015). "A multiple testing method for hypotheses structured in a directed acyclic graph." *Biometrical Journal*, **57**(1), 123–143.

**Examples**

```
X0 <- rbind(c(0.5, 0.6), c(0.8, 0.9), c(0.9, 0.8))
p <- c(0.02, 0.025, 0.1)
alpha <- 0.05
dag_test_MG(X0, p, alpha)
dag_test_MG(X0, p, alpha, version = "any")
dag_test_MG(X0, p, alpha, sparse = TRUE)
```

---

`get_boundary_points`     *get\_boundary\_points*

---

**Description**

Given a set of points, returns the minimal subset with the same upper hull.

**Usage**

```
get_boundary_points(X)
```

**Arguments**

<code>X</code>	a numeric matrix with one point per row.
----------------	--

**Value**

A numeric matrix of the same number of columns as  $X$ .

**Examples**

```
X <- rbind(c(0, 1), c(1, 0), c(1, 0), c(1, 1))
get_boundary_points(X)
```

---

get\_DAG

*get\_DAG*

---

**Description**

This function is used to construct the induced DAG, induced polyforest and reverse topological orderings thereof from a numeric matrix  $X_0$ . See Definition 2 in Müller et al. (2023).

**Usage**

```
get_DAG(X0, sparse = FALSE, twoway = FALSE)
```

**Arguments**

$X_0$	a numeric matrix.
sparse	logical. Either the induced DAG (FALSE) or the induced polyforest (TRUE) is constructed.
twoway	logical. If FALSE, only leaves, parents, ancestors and reverse topological ordering are returned. If TRUE, then roots, children and descendants are also provided.

**Value**

A list with named elements giving the leaves, parents, ancestors and reverse topological ordering and additionally, if `twoway == TRUE`, the roots, children and descendants, of the constructed graph.

**References**

Müller MM, Reeve HWJ, Cannings TI, Samworth RJ (2023). “Isotonic subgroup selection.” *arXiv preprint arXiv:2305.04852*.

**Examples**

```
X <- rbind(
  c(0.2, 0.8), c(0.2, 0.8), c(0.1, 0.7),
  c(0.2, 0.1), c(0.3, 0.5), c(0.3, 0)
)
get_DAG(X0 = X)
get_DAG(X0 = X, sparse = TRUE, twoway = TRUE)
```

---

`get_p_classification` *get\_p\_classification*

---

## Description

Calculate the p-value in Definition 21 of Müller et al. (2023).

## Usage

```
get_p_classification(X, y, x0, tau)
```

## Arguments

<code>X</code>	a numeric matrix specifying the covariates.
<code>y</code>	a numeric vector with $\text{length}(y) == \text{nrow}(X)$ and $\text{all}((y \geq 0) \& (y \leq 1))$ specifying the responses.
<code>x0</code>	a numeric vector specifying the point of interest, such that $\text{length}(x0) == \text{ncol}(X)$ .
<code>tau</code>	a single numeric value in $[0,1)$ specifying the threshold of interest.

## Value

A single numeric value in  $(0, 1]$ .

## References

Müller MM, Reeve HWJ, Cannings TI, Samworth RJ (2023). “Isotonic subgroup selection.” *arXiv preprint arXiv:2305.04852*.

## Examples

```
set.seed(123)
n <- 100
d <- 2
X <- matrix(runif(d * n), ncol = d)
eta <- function(x) sum(x)
X_eta <- apply(X, MARGIN = 1, FUN = function(x) 1 / (1 + exp(-eta(x))))
y <- as.numeric(runif(n) < X_eta)
get_p_classification(X, y, x0 = c(1, 1), tau = 0.6)
get_p_classification(X, y, x0 = c(1, 1), tau = 0.9)
```

---

get_p_Gaussian	<i>get_p_Gaussian</i>
----------------	-----------------------

---

### Description

Calculate the p-value in Definition 19 of Müller et al. (2023).

### Usage

```
get_p_Gaussian(X, y, x0, tau)
```

### Arguments

<code>X</code>	a numeric matrix specifying the covariates.
<code>y</code>	a numeric vector with $\text{length}(y) == \text{nrow}(X)$ specifying the responses.
<code>x0</code>	a numeric vector specifying the point of interest, such that $\text{length}(x0) == \text{ncol}(X)$ .
<code>tau</code>	a single numeric value specifying the threshold of interest.

### Value

A single numeric value in  $(0, 1]$ .

### References

Müller MM, Reeve HWJ, Cannings TI, Samworth RJ (2023). “Isotonic subgroup selection.” *arXiv preprint arXiv:2305.04852*.

### Examples

```
set.seed(123)
n <- 100
d <- 2
X <- matrix(runif(d * n), ncol = d)
eta <- function(x) sum(x)
y <- apply(X, MARGIN = 1, FUN = eta) + rnorm(n, sd = 1)
get_p_Gaussian(X, y, x0 = c(1, 1), tau = 1)
get_p_Gaussian(X, y, x0 = c(1, 1), tau = -1)
```



---

get\_p\_subGaussian      *get\_p\_subGaussian*

---

## Description

Calculate the p-value in Definition 1 of Müller et al. (2023).

## Usage

```
get_p_subGaussian(X, y, x0, sigma2, tau)
```

## Arguments

X	a numeric matrix specifying the covariates.
y	a numeric vector with $\text{length}(y) == \text{nrow}(X)$ specifying the responses.
x0	a numeric vector specifying the point of interest, such that $\text{length}(x0) == \text{ncol}(X)$ .
sigma2	a single positive numeric value specifying the variance parameter.
tau	a single numeric value specifying the threshold of interest.

## Value

A single numeric value in (0, 1].

## References

Müller MM, Reeve HWJ, Cannings TI, Samworth RJ (2023). “Isotonic subgroup selection.” *arXiv preprint arXiv:2305.04852*.

## Examples

```
set.seed(123)
n <- 100
d <- 2
X <- matrix(runif(d*n), ncol = d)
eta <- function(x) sum(x)
y <- apply(X, MARGIN = 1, FUN = eta) + rnorm(n, sd = 0.5)
get_p_subGaussian(X, y, x0 = c(1, 1), sigma2 = 0.25, tau = 1)
get_p_subGaussian(X, y, x0 = c(1, 1), sigma2 = 0.25, tau = 3)
```

---

`get_p_subGaussian_NM` *get\_p\_subGaussian\_NM*

---

### Description

Calculate the p-value in Definition 18 of Müller et al. (2023).

### Usage

```
get_p_subGaussian_NM(X, y, x0, sigma2, tau, rho = 0.5)
```

### Arguments

<code>X</code>	a numeric matrix specifying the covariates.
<code>y</code>	a numeric vector with $\text{length}(y) == \text{nrow}(X)$ specifying the responses.
<code>x0</code>	a numeric vector specifying the point of interest, such that $\text{length}(x0) == \text{ncol}(X)$ .
<code>sigma2</code>	a single positive numeric value specifying the variance parameter.
<code>tau</code>	a single numeric value specifying the threshold of interest.
<code>rho</code>	a single positive numeric value serving as hyperparameter.

### Value

A single numeric value in  $(0, 1]$ .

### References

Müller MM, Reeve HWJ, Cannings TI, Samworth RJ (2023). “Isotonic subgroup selection.” *arXiv preprint arXiv:2305.04852*.

### Examples

```
set.seed(123)
n <- 100
d <- 2
X <- matrix(runif(d * n), ncol = d)
eta <- function(x) sum(x)
y <- apply(X, MARGIN = 1, FUN = eta) + rnorm(n, sd = 0.5)
get_p_subGaussian_NM(X, y, x0 = c(1, 1), sigma2 = 0.25, tau = 3)
get_p_subGaussian_NM(X, y, x0 = c(1, 1), sigma2 = 0.25, tau = 1)
get_p_subGaussian_NM(X, y, x0 = c(1, 1), sigma2 = 0.25, tau = 1, rho = 2)
```

---

get_p_value	<i>get_p_value</i>
-------------	--------------------

---

### Description

A wrapper function used to call the correct function for calculating the p-value.

### Usage

```
get_p_value(
  p_value_method = c("sub-Gaussian-normalmixture", "sub-Gaussian", "Gaussian",
    "classification", "quantile"),
  X,
  y,
  x0,
  tau,
  sigma2,
  rho = 1/2,
  theta = 1/2
)
```

### Arguments

p_value_method	one of c("sub-Gaussian", "sub-Gaussian-normalmixture", "Gaussian", "classification", "quantile") specifying which p-value construction should be used. See Definitions 1, 18, 19 and 21 and Lemma 24 by Müller et al. (2023) respectively. For p_value_method == "quantile", the version with the p-value from Definition 19 is implemented.
X	a numeric matrix specifying the covariates.
y	a numeric vector with length(y) == nrow(X) specifying the responses.
x0	a numeric vector specifying the point of interest, such that length(x0) == ncol(X).
tau	a single numeric value specifying the threshold of interest.
sigma2	a single positive numeric value specifying the variance parameter (required only if p_value_method %in% c("sub-Gaussian", "sub-Gaussian-normalmixture")).
rho	a single positive numeric value serving as hyperparameter (required only if p_value_method == "sub-Gaussian-normalmixture").
theta	a single numeric value in (0, 1) specifying the quantile of interest when p_value_method == "quantile". Defaults to 1/2, i.e.-the median.

### Value

A single numeric value in (0, 1].

## References

Müller MM, Reeve HWJ, Cannings TI, Samworth RJ (2023). “Isotonic subgroup selection.” *arXiv preprint arXiv:2305.04852*.

## Examples

```
set.seed(123)
n <- 100
d <- 2
X <- matrix(runif(d * n), ncol = d)
eta <- function(x) sum(x)
X_eta <- apply(X, MARGIN = 1, FUN = function(x) 1 / (1 + exp(-eta(x))))
y <- as.numeric(runif(n) < X_eta)
get_p_value(p_value_method = "classification", X, y, x0 = c(1, 1), tau = 0.6)
get_p_value(p_value_method = "classification", X, y, x0 = c(1, 1), tau = 0.9)

X_eta <- apply(X, MARGIN = 1, FUN = eta)
y <- X_eta + rcauchy(n)
get_p_value(p_value_method = "quantile", X, y, x0 = c(1, 1), tau = 1/2)
get_p_value(p_value_method = "quantile", X, y, x0 = c(1, 1), tau = 3)
get_p_value(p_value_method = "quantile", X, y, x0 = c(1, 1), tau = 3, theta = 0.95)
```

---

ISS

ISS

---

## Description

The function implements the combination of p-value calculation and familywise error rate control through DAG testing procedures described in Müller et al. (2023).

## Usage

```
ISS(
  X,
  y,
  tau,
  alpha = 0.05,
  m = nrow(X),
  p_value = c("sub-Gaussian-normalmixture", "sub-Gaussian", "Gaussian", "classification",
    "quantile"),
  sigma2,
  rho = 1/2,
  FWER_control = c("ISS", "Holm", "MG all", "MG any", "split", "split oracle"),
  minimal = FALSE,
  split_proportion = 1/2,
  eta = NA,
  theta = 1/2
)
```

**Arguments**

<code>X</code>	a numeric matrix specifying the covariates.
<code>y</code>	a numeric vector with $\text{length}(y) == \text{nrow}(X)$ specifying the responses.
<code>tau</code>	a single numeric value specifying the threshold of interest.
<code>alpha</code>	a numeric value in $(0, 1]$ specifying the Type I error rate.
<code>m</code>	an integer value between 1 and $\text{nrow}(X)$ specifying the size of the subsample of $X$ at which the hypotheses should be tested.
<code>p_value</code>	one of <code>c("sub-Gaussian", "sub-Gaussian-normalmixture", "Gaussian", "classification", "quantile")</code> specifying which p-value construction should be used. See Definitions 1, 18, 19 and 21 and Lemma 24 by Müller et al. (2023) respectively. For <code>p_value == "quantile"</code> , the version with the p-value from Definition 19 is implemented.
<code>sigma2</code>	a single positive numeric value specifying the variance parameter (only needed if <code>p_value %in% c("sub-Gaussian", "sub-Gaussian-normalmixture")</code> ).
<code>rho</code>	a single positive numeric value serving as hyperparameter (only used if <code>p_value == "sub-Gaussian-normalmixture"</code> ).
<code>FWER_control</code>	one of <code>c("ISS", "Holm", "MG all", "MG any", "split", "split oracle")</code> , specifying how the familywise error rate is controlled. The first corresponds to Algorithm 1 by Müller et al. (2023), the second is Holm's procedure, the two starting with "MG" correspond to the procedures by Meijer and Goeman (2015) for one-way logical relationships, and the final two containing "split" to the sample splitting techniques in Appendix B of Müller et al. (2023).
<code>minimal</code>	a logical value determining whether the output should be reduced to the minimal number of points leading to the same selected set.
<code>split_proportion</code>	when <code>FWER_control %in% c("split", "split oracle")</code> , the number of data points in the first split of the data is $\text{ceiling}(\text{split\_proportion} * \text{nrow}(X))$ .
<code>eta</code>	when <code>FWER_control == "split oracle"</code> , this parameter needs to be used to provide the true regression function, which should take a vector of covariates as inputs and output a single numeric value.
<code>theta</code>	a single numeric value in $(0, 1)$ specifying the quantile of interest when <code>p_value_method == "quantile"</code> . Defaults to $1/2$ , i.e.~the median.

**Value**

A numeric matrix giving the points in  $X$  determined to lie in the  $\tau$ -superlevel set of the regression function with probability at least  $1 - \alpha$  or, if `minimal == TRUE`, a subset of points thereof that have the same upper hull.

**References**

Meijer RJ, Goeman JJ (2015). "A multiple testing method for hypotheses structured in a directed acyclic graph." *Biometrical Journal*, **57**(1), 123–143.

Müller MM, Reeve HWJ, Cannings TI, Samworth RJ (2023). "Isotonic subgroup selection." *arXiv preprint arXiv:2305.04852v2*.

**Examples**

```

d <- 2
n <- 1000
m <- 100
sigma2 <- (1 / 4)^2
tau <- 0.5
alpha <- 0.05

X <- matrix(runif(n * d), nrow = n)
eta_X <- apply(X, MARGIN = 1, max)
y <- eta_X + rnorm(n, sd = sqrt(sigma2))
X_rej <- ISS(X = X, y = y, tau = tau, alpha = alpha, m = m, sigma2 = sigma2)

if (d == 2) {
  plot(0, type = "n", xlim = c(0, 1), ylim = c(0, 1), xlab = NA, ylab = NA)
  for (i in 1:nrow(X_rej)) {
    rect(
      xleft = X_rej[i, 1], xright = 1, ybottom = X_rej[i, 2], ytop = 1,
      border = NA, col = "indianred"
    )
  }

  points(X, pch = 16, cex = 0.5, col = "gray")
  points(X[1:m, ], pch = 16, cex = 0.5, col = "black")
  lines(x = c(0, tau), y = c(tau, tau), lty = 2)
  lines(x = c(tau, tau), y = c(tau, 0), lty = 2)

  legend(
    x = "bottomleft",
    legend = c(
      "superlevel set boundary",
      "untested covariate points",
      "tested covariate points",
      "selected set"
    ),
    col = c("black", "gray", "black", "indianred"),
    lty = c(2, NA, NA, NA),
    lwd = c(1, NA, NA, NA),
    pch = c(NA, 16, 16, NA),
    fill = c(NA, NA, NA, "indianred"),
    border = c(NA, NA, NA, "indianred")
  )
}

```

# Index

[dag\\_test\\_FS](#), [2](#)  
[dag\\_test\\_Holm](#), [3](#)  
[dag\\_test\\_ISS](#), [3](#)  
[dag\\_test\\_MG](#), [4](#)

[get\\_boundary\\_points](#), [5](#)  
[get\\_DAG](#), [6](#)  
[get\\_p\\_classification](#), [7](#)  
[get\\_p\\_Gaussian](#), [8](#)  
[get\\_p\\_subGaussian](#), [9](#)  
[get\\_p\\_subGaussian\\_NM](#), [10](#)  
[get\\_p\\_value](#), [11](#)

[ISS](#), [12](#)