

# Package ‘PKLMtest’

May 7, 2026

**Type** Package

**Title** Classification Based MCAR Test

**Version** 1.0.1

**Description** Implementation of a KL-based (Kullback-Leibler) test for MCAR (Missing Completely At Random) in the context of missing data as introduced in Michel et al. (2021) <[doi:10.48550/arXiv.2109.10150](https://doi.org/10.48550/arXiv.2109.10150)>.

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**Depends** parallel, stats, ranger

**NeedsCompilation** no

**Author** Meta-Lina Spohn [aut, cre],  
Loris Michel [aut],  
Jeffrey Naef [aut]

**Maintainer** Meta-Lina Spohn <[metalina.spohn@stat.math.ethz.ch](mailto:metalina.spohn@stat.math.ethz.ch)>

**Repository** CRAN

**Date/Publication** 2021-11-05 16:10:02 UTC

## Contents

genU . . . . .	2
PKLMtest . . . . .	2
truncProb . . . . .	3
<b>Index</b>	<b>4</b>

genU *Generate the test statistic*

---

### Description

Generate the test statistic

### Usage

```
genU(st, lab)
```

### Arguments

st            a ranger forest object.  
lab           an integer value containing the class labels

### Value

the likelihood-based test statistic

---

PKLMtest *PKLMtest: compute a p-value for testing MCAR*

---

### Description

PKLMtest: compute a p-value for testing MCAR

### Usage

```
PKLMtest(  
  X,  
  num.proj = 300,  
  num.trees.per.proj = 10,  
  nrep = 500,  
  min.node.size = 10,  
  size.resp.set = 2,  
  compute.partial.pvals = FALSE,  
  ...  
)
```

**Arguments**

**X** a numeric matrix containing missing values encoded as NA, the data.  
**num.proj** a positive integer specifying the number of projections to consider for the score.  
**num.trees.per.proj** a positive integer, the number of trees per projection.  
**nrep** a positive integer, the number of permutations.  
**min.node.size** a positive number, the minimum number of nodes in a tree.  
**size.resp.set** an integer ( $\geq 2$ ), maximum number of classes allowed to be compared in each projection.  
**compute.partial.pvals** a boolean, indicate if partial p-values should be computed as well.  
**...** additional parameters.

**Value**

a numeric value, the p-value(s) for the MCAR test, the first value is always the global p-value and if `compute.partial.pvals` is set to `TRUE`, the next values are the partial p-values for the relative importance of each variable.

**Examples**

```

n <- 100
X <- cbind(rnorm(n), rnorm(n))
X.NA <- X
X.NA[,1] <- ifelse(stats::runif(n) <= 0.2, NA, X[,1])

pval <- PKLMtest(X.NA, num.proj = 5)

```

---

truncProb	<i>Truncation of probability</i>
-----------	----------------------------------

---

**Description**

Truncation of probability

**Usage**

```
truncProb(p)
```

**Arguments**

**p** a numeric value between 0 and 1 to be truncated

**Value**

a numeric value with truncated probabilities

# Index

genU, [2](#)

PKLMtest, [2](#)

truncProb, [3](#)