# Package 'TailClassifier'

October 12, 2022

**Type** Package

**Title** Tail Classifier for Thick-Tailed Discrete Data

**Version** 0.1.0

**Author** Jialin Zhang (JZ)

**Maintainer** Jialin Zhang (JZ) <jzhang@math.msstate.edu>

**Description** Function TailClassifier() in this package is a Tail-Classifier function. The function suggests one of the following types of tail for your discrete data: 1) Power decaying tail; 2) Sub-exponential decaying tail; and 3) Near-exponential decaying tail.

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**Imports** readr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-06-17 07:50:11 UTC

## R topics documented:



---

`TailClassifier` *Tail Classifier for Thick-Tailed Discrete Data*

---

### Description

Function TailClassifier() in this package is a Tail-Classifier function. The function suggests one of the following types of tail for your discrete data: 1) Power decaying tail; 2) Sub-exponential decaying tail; and 3) Near-exponential decaying tail.

## Usage

```
TailClassifier(
  sample_frequencies,
  v.left = 5,
  v.right = min(floor(sum(sample_frequencies)/20), 500)
)
```

## Arguments

| | |
|---|---|
| sample_frequencies | The frequency counts for your discrete sample data. |
| v.left | The starting point of tail profile. 5 is recommended. A smaller v.left may lead to unreliable results. A larger v.left might be adopted if the sample size is extremely large. |
| v.right | The ending point of tail profile. Recommendation is 5% of the sample size but no more than 500. For example, a sample with size 1000 could choose v.right to be 50; and a sample with size 20000 could choose v.right to be 500. |

## Value

A statement on the type of tail.

## Examples

```
## read built-in random sample that was generated under a sub-exponential distribution
csv <- system.file("extdata", "sample_data.csv", package = "TailClassifier")
sample_data <- readr::read_csv(csv)
## generate the frequency table of the sample
sample_freq=table(sample_data)
## make a classification
TailClassifier(sample_freq)
```

# Index