

# Package ‘GRNNs’

October 12, 2022

**Title** General Regression Neural Networks Package

**Version** 0.1.0

**Description** This General Regression Neural Networks Package uses various distance functions. It was motivated by Specht (1991, ISBN:1045-9227), and updated from previous published paper Li et al. (2016) <doi:10.1016/j.palaeo.2015.11.005>. This package includes various functions, although “euclidean” distance is used traditionally.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Imports** cvTools, rdist, scales, stats, vegan

**Depends** R (>= 3.5.0)

**Suggests** rmarkdown, knitr, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Shufeng LI [aut, cre] (<<https://orcid.org/0000-0002-5144-4432>>)

**Maintainer** Shufeng LI <lisf@xtbg.org.cn>

**Repository** CRAN

**Date/Publication** 2021-09-08 09:30:04 UTC

## R topics documented:

findSpread . . . . .	2
findSpreadRdist . . . . .	2
findSpreadVegan . . . . .	3
grnn . . . . .	4
grnn.distance . . . . .	5
grnn.kfold . . . . .	5
met . . . . .	6
physg . . . . .	7
veg.distance . . . . .	8

**Index****9**

---

findSpread	<i>Find best spread</i>
------------	-------------------------

---

**Description**

Find best spread

**Usage**

```
findSpread(p_train, v_train, k, fun, scale = TRUE)
```

**Arguments**

p_train	The dataframe of training predictor dataset
v_train	The dataframe of training response variables
k	The numeric number of k folds
fun	The distance function
scale	The logic statements (TRUE/FALSE)

**Value**

Best spread

**Examples**

```
data("met")
data("physg")
## Not run: best.spread<-findSpread(physg,met,10,"bray",scale=TRUE)
```

---

findSpreadRdist	<i>find best spreads using Rdist</i>
-----------------	--------------------------------------

---

**Description**

find best spreads using Rdist

**Usage**

```
findSpreadRdist(x, y, k, fun, scale = TRUE)
```

**Arguments**

x	The dataframe of training predictor dataset
y	The dataframe of training response variables
k	The numeric number of k folds
fun	The distance function
scale	The logic statements (TRUE/FALSE)

**Value**

The vector of best spreads

---

<i>findSpreadVegan</i>	<i>Find best spread using vegan function</i>
------------------------	--

---

**Description**

Find best spread using vegan function

**Usage**

```
findSpreadVegan(x, y, k, fun, scale = TRUE)
```

**Arguments**

x	The dataframe of training predictor dataset
y	The dataframe of training response variables
k	The numeric number of k folds
fun	The distance function
scale	The logic statements (TRUE/FALSE)

**Value**

The vector of best spreads

---

 grnn

 General Regression Neural Networks (GRNNs)
 

---

### Description

This GRNNs uses various distance functions including: "euclidean", "minkowski", "manhattan", "maximum", "canberra", "angular", "correlation", "absolute\_correlation", "hamming", "jaccard", "bray", "kulczynski", "gower", "altGower", "morisita", "horn", "mountford", "raup", "binomial", "chao", "cao", "mahalanobis".

### Usage

```
grnn(p_input, p_train, v_train, fun = "euclidean", best.spread, scale = TRUE)
```

### Arguments

p_input	The dataframe of input predictors
p_train	The dataframe of training predictor dataset
v_train	The dataframe of training response variables
fun	The distance function
best.spread	The vector of best spreads
scale	The logic statements (TRUE/FALSE)

### Value

The predictions

### Examples

```
data("met")
data("physg")
best.spread<-c(0.33,0.33,0.31,0.34,0.35,0.35,0.32,0.31,0.29,0.35,0.35)
predict<-physg[1,]
physg.train<-physg[-1,]
met.train<-met[-1,]
prediction<-grnn(predict,physg.train,met.train,fun="euclidean",best.spread,scale=TRUE)
```

---

grnn.distance	<i>grnn distance</i>
---------------	----------------------

---

**Description**

grnn distance

**Usage**

```
grnn.distance(x, y, fun)
```

**Arguments**

x	The dataframe of training predictor dataset
y	The dataframe of training response variables
fun	The distance function

**Value**

The matrix of distance between a and b

**Examples**

```
data("physg")
physg.train<-physg[1:10,]
physg.test<-physg[11:30,]
distance<-grnn.distance(physg.test,physg.train,"bray")
```

---

grnn.kfold	<i>General Regression Neural Networks (GRNNs)</i>
------------	---

---

**Description**

General Regression Neural Networks (GRNNs)

**Usage**

```
grnn.kfold(x, y, k, fun, scale = TRUE)
```

**Arguments**

x	The dataframe of training predictor dataset
y	The dataframe of training response variables
k	The numeric number of k folds
fun	The distance function
scale	The logic statements (TRUE/FALSE)

**Value**

rmse,stdae,stdev,mae,r,pvalue,best spread

**Examples**

```
data("met")
data("physg")
results_kfold<-grnn.kfold(physg,met,10,"euclidean",scale=TRUE)
```

---

met	<i>meteorological dataset</i>
-----	-------------------------------

---

**Description**

Data from a global collection by Robert A. Spicer. It include 11 climate variables from 378 sites.

**Usage**

```
met
```

**Format**

A data frame with 378 rows and 11 variables:

```
MAT double COLUMN_DESCRIPTION
WMT double COLUMN_DESCRIPTION
CMT double COLUMN_DESCRIPTION
GROWSEAS double COLUMN_DESCRIPTION
GSP double COLUMN_DESCRIPTION
MMGSP double COLUMN_DESCRIPTION
Three_WET double COLUMN_DESCRIPTION
Three_DRY double COLUMN_DESCRIPTION
RH double COLUMN_DESCRIPTION
SH double COLUMN_DESCRIPTION
ENTHAL double COLUMN_DESCRIPTION
```

**Details**

DETAILS

---

 physg

*physiognomy dataset*


---

**Description**

Data from a global collection by Robert A. Spicer. It include 31 leaf physiognomies variables from 378 sites.

**Usage**

physg

**Format**

A data frame with 378 rows and 31 variables:

Lobed double COLUMN\_DESCRIPTION  
 No.Teeth double COLUMN\_DESCRIPTION  
 Regular.teeth double COLUMN\_DESCRIPTION  
 Close.teeth double COLUMN\_DESCRIPTION  
 Round.teeth double COLUMN\_DESCRIPTION  
 Acute.teeth double COLUMN\_DESCRIPTION  
 Compound.teeth double COLUMN\_DESCRIPTION  
 Nanophyll double COLUMN\_DESCRIPTION  
 Leptophyll.1 double COLUMN\_DESCRIPTION  
 Leptophyll.2 double COLUMN\_DESCRIPTION  
 Microphyll.1 double COLUMN\_DESCRIPTION  
 Microphyll.2 double COLUMN\_DESCRIPTION  
 Microphyll.3 double COLUMN\_DESCRIPTION  
 Mesophyll.1 double COLUMN\_DESCRIPTION  
 Mesophyll.2 double COLUMN\_DESCRIPTION  
 Mesophyll.3 double COLUMN\_DESCRIPTION  
 Emarginate.apex double COLUMN\_DESCRIPTION  
 Round.apex double COLUMN\_DESCRIPTION  
 Acute.apex double COLUMN\_DESCRIPTION  
 Attenuate.apex double COLUMN\_DESCRIPTION  
 Cordate.base double COLUMN\_DESCRIPTION  
 Round.base double COLUMN\_DESCRIPTION  
 Acute.base double COLUMN\_DESCRIPTION  
 L.W..1.1 double COLUMN\_DESCRIPTION

L.W.1.2.1 double COLUMN\_DESCRIPTION  
 L.W.2.3.1 double COLUMN\_DESCRIPTION  
 L.W.3.4.1 double COLUMN\_DESCRIPTION  
 L.W..4.1 double COLUMN\_DESCRIPTION  
 Obovate double COLUMN\_DESCRIPTION  
 Elliptic double COLUMN\_DESCRIPTION  
 Ovate double COLUMN\_DESCRIPTION

## Details

DETAILS

---

veg.distance	<i>distance using vegdist</i>
--------------	-------------------------------

---

## Description

distance using vegdist

## Usage

```
veg.distance(a, b, fun = "bray")
```

## Arguments

a	The dataframe of training predictor dataset
b	The dataframe of validation predictor dataset
fun	The distance function

## Value

The matrix of distance between a and b

## Examples

```
data("physg")
physg.train<-physg[1:10,]
physg.test<-physg[11:30,]
distance<-veg.distance(physg.test,physg.train,"bray")
```



# Index

## \* datasets

met, [6](#)

physg, [7](#)

findSpread, [2](#)

findSpreadRdist, [2](#)

findSpreadVegan, [3](#)

grnn, [4](#)

grnn.distance, [5](#)

grnn.kfold, [5](#)

met, [6](#)

physg, [7](#)

veg.distance, [8](#)