

# Package ‘RankMetric’

October 30, 2018

**Type** Package

**Title** Calculates the Distance between Rankings for different Metrics

**Version** 0.0.0.9000

**Author** person(``Lucy", ``Small", email = ``lucy.small@ucdconnect.ie", role =  
c(``aut", ``cre"))

**Maintainer** Lucy Small <lucy.small@ucdconnect.ie>

**Description** This package calculates the distance between any pair of full or  
partial rankings for the metrics given in Metric Methods for Analyzing  
Partially Ranked Data by Critchlow. It allows for tied rankings.

**Depends** R (>= 3.3.1)

**License** GPL-3

**LazyData** true

**LinkingTo** Rcpp

**Imports** Rcpp , LIStest, stats

**RoxygenNote** 6.1.0

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** yes

## R topics documented:

|                    |    |
|--------------------|----|
| cayley . . . . .   | 2  |
| cayleyE . . . . .  | 3  |
| cayleyP . . . . .  | 3  |
| distance . . . . . | 4  |
| fifa16 . . . . .   | 5  |
| ham . . . . .      | 6  |
| hamE . . . . .     | 6  |
| hamG . . . . .     | 7  |
| hamP . . . . .     | 8  |
| inv . . . . .      | 8  |
| kend . . . . .     | 9  |
| kendE . . . . .    | 9  |
| kendG . . . . .    | 10 |
| kendP . . . . .    | 11 |

|                      |           |
|----------------------|-----------|
| labour . . . . .     | 11        |
| spear . . . . .      | 12        |
| spearE . . . . .     | 12        |
| spearfoot . . . . .  | 13        |
| spearfootE . . . . . | 14        |
| spearfootG . . . . . | 14        |
| spearfootP . . . . . | 15        |
| spearG . . . . .     | 16        |
| spearP . . . . .     | 16        |
| ulam . . . . .       | 17        |
| ulamE . . . . .      | 18        |
| ulamG . . . . .      | 18        |
| ulamP . . . . .      | 19        |
| <b>Index</b>         | <b>20</b> |

---

|        |                          |
|--------|--------------------------|
| cayley | <i>Cayley's Distance</i> |
|--------|--------------------------|

---

**Description**

cayley returns Cayley's distance between two full rankings.

**Usage**

cayley(x, y)

**Arguments**

x, y                    integer vectors

**Value**

Returns Cayley's distance between x and y

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
x = c(3,1,2,5,4)
y = c(1,2,3,4,5)
cayley(x,y)
```

---

`cayleyE`*Cayley's Distance with Ties*

---

**Description**

Computes Cayley's distance between two rankings, where items with equal ranking are now permitted. The number of items placed in the *i*th category must be the same. The number of groups must be 2 or 3, as Cayley's distance is undefined in other cases.

**Usage**`cayleyE(x, y)`**Arguments**

|                   |                 |
|-------------------|-----------------|
| <code>x, y</code> | integer vectors |
|-------------------|-----------------|

**Value**

Returns Cayley's distance between the two rankings.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,2,3)
b = c(1,2,2,3,3)
cayleyE(a,b)
```

---

`cayleyP`*Cayley's Distance for Partial rankings*

---

**Description**

Computes Cayley's distance between two partial rankings.

**Usage**`cayleyP(a, b, k)`**Arguments**

|                   |                 |
|-------------------|-----------------|
| <code>a, b</code> | integer vectors |
| <code>k</code>    | integer         |

**Value**

Returns Cayley's distance between two partial rankings, where only the first *k* items have been ranked. No ties are permitted.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,5,4)
b = c(1,2,3,4,5)
k=3
cayleyP(a,b,k)
```

---

|          |                             |
|----------|-----------------------------|
| distance | <i>Metrics for Rankings</i> |
|----------|-----------------------------|

---

**Description**

Calculates the distances between a set of rankings or permutations, using one of six metrics.

**Usage**

```
distance(x, metric, perm = TRUE, ranktype = "full", k = 2)
```

**Arguments**

|          |   |
|----------|---|
| x        | matrix where each row is a ranking or a permutation   |
| metric   | a distance metric, one of <ul style="list-style-type: none"> <li>• "kendall"</li> <li>• "ulam"</li> <li>• "spearrho", spearman's rho</li> <li>• "spearfoot", spearman's footrule</li> <li>• "hamming"</li> <li>• "cayley"</li> </ul>  |
| perm     | TRUE for a matrix of permutations, FALSE for a matrix of rankings   |
| ranktype | indicates the type of ranking <ul style="list-style-type: none"> <li>• "full" full ranking</li> <li>• "partial", partial ranking with no ties</li> <li>• "sometimes", ties allowed, the same number must be ranked 1st, 2nd etc</li> <li>• "anyties", ties allowed any number can be ranked 1st, 2nd etc</li> </ul> |
| k        | the number of items ranked in a partial ranking   |

**Value**

Returns a matrix of the distances between all the rankings/ permutations in the input matrix. If ranktype is "anyties" the function will work for any type of partial ranking or ranking with ties. However using "full" for complete rankings or "partial" for partial ranking with no ties for that kind of data will run faster.

If the ranking of some of the items is not known or given they can be left as NA.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**References**

<http://www.springer.com/gp/book/9780387962887>

**Examples**

```
x = t(matrix(replicate(10,sample(1:5,5)),ncol=10))
distance(x,metric = "spearfoot",perm = FALSE,"full")
```

---

fifa16

*Voting data from 2016 FIFA Best Player of the Year*

---

**Description**

This is the voting data with voter covariates from the FIFA Best Player of the Year 2016 award. There were 23 candidates on the shortlist and 450 voters. Each voter provides their top-3 choices. The voters were the national captains, manager, and one media representative from each country.

FIFA, the world football governing body, divides member countries into six continental confederations, which each organise continental national and club competitions. The confederation of each voter is given.

**Usage**

```
data(fifa16)
```

**Format**

A data frame with 450 voters (rows) and 30 variables (columns). The first four columns give the voter name, role (captain, manager or media), country of origin and confederation(AFC, CAF, CONCAF,CONMEBOL or UEFA). Columns vote1, vote2, vote3 give the names of the candidates chosen by each voter as their top-3 ranking. The remaining columns (8:30) give the full partial rankings in permutation form. The votes are arbitrarily filled in after the top-3.

**Source**

[http://resources.fifa.com/mm/Document/the-best/PlayeroftheYear-Men/02/86/27/05/faward\\_MenPlayer2016\\_Neutral.pdf](http://resources.fifa.com/mm/Document/the-best/PlayeroftheYear-Men/02/86/27/05/faward_MenPlayer2016_Neutral.pdf)

---

|     |                         |
|-----|-------------------------|
| ham | <i>Hamming Distance</i> |
|-----|-------------------------|

---

**Description**

Returns the Hamming distance between two full rankings.

**Usage**

```
ham(a, b)
```

**Arguments**

a, b                      integer vectors

**Value**

The Hamming distance between the two rankings.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,5,4)
b = c(1,2,3,4,5)
ham(a,b)
```

---

|      |                                   |
|------|-----------------------------------|
| hamE | <i>Hamming Distance with Ties</i> |
|------|-----------------------------------|

---

**Description**

Computes the Hamming distance between two rankings, where items with equal ranking are now permitted. The number of items placed in the *i*th category must be the same.

**Usage**

```
hamE(x, y)
```

**Arguments**

x, y                      integer vectors

**Value**

Returns the Hamming distance between the two rankings.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,2,3)
b = c(1,2,2,3,3)
hamE(a,b)
```

---

hamG

*Hamming Distance for any Number of Ties*

---

**Description**

Computes Hamming distance between two rankings, where any number of items with equal rankings are now permitted in each ranking.

**Usage**

```
hamG(x, y)
```

**Arguments**

x, y                      integer vectors

**Value**

Returns Hamming distance between the two rankings.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,2,3)
b = c(1,2,3,4,4)
hamG(a,b)
```

---

hamP

*Hamming Distance for Partial rankings*


---

**Description**

Computes the Hamming distance between two partial rankings.

**Usage**

hamP(a, b, k)

**Arguments**

|      |                 |
|------|-----------------|
| a, b | integer vectors |
| k    | integer         |

**Value**

Returns the Hamming distance between the two partial rankings, where only the first k items have been ranked. No ties are permitted.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,5,4)
b = c(1,2,3,4,5)
k=3
hamP(a,b,k)
```

---

inv

*Inverse Permutation*


---

**Description**

Computes the inverse of a permutation.

**Usage**

inv(x)

**Arguments**

|   |                   |
|---|-------------------|
| x | an integer vector |
|---|-------------------|

**Value**

Returns the inverse permutation of a vector.



**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,5,4)
inv(a)
```

---

|      |                      |
|------|----------------------|
| kend | <i>Kendall's Tau</i> |
|------|----------------------|

---

**Description**

Computes Kendall's tau between two full rankings.

**Usage**

```
kend(a, b)
```

**Arguments**

a, b                      integer vectors

**Value**

Returns Kendall's tau between two full rankings.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,5,4)
b = c(1,2,3,4,5)
kend(a,b)
```

---

|       |   |
|-------|---|
| kendE | <i>Kendall's Tau for Tankings with Ties</i> |
|-------|---|

---

**Description**

Computes Kendall's tau between two rankings, where items with equal ranking are now permitted. The number of items placed in the *i*th category must be the same.

**Usage**

```
kendE(x, y)
```

**Arguments**

x, y                      integer vectors

**Value**

Returns Kendall's tau between the two rankings.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,2,3)
b = c(1,2,2,3,3)
kendE(a,b)
```

---

kendG

*Kendall's Tau for any Number of Ties*

---

**Description**

Computes Kendall's tau between two rankings, where any number of items with equal ranking are now permitted in each ranking.

**Usage**

```
kendG(x, y)
```

**Arguments**

x, y                      integer vectors

**Value**

Returns Kendall's tau between the two rankings.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,2,3)
b = c(1,2,3,4,4)
kendG(a,b)
```

---

|       |  |
|-------|--|
| kendP | <i>Kendall's Distance for Partial rankings</i> |
|-------|--|

---

**Description**

Computes Kendall's distance between two partial rankings.

**Usage**

```
kendP(a, b, k)
```

**Arguments**

|      |                 |
|------|-----------------|
| a, b | integer vectors |
| k    | integer         |

**Value**

Returns Kendall's distance between two rankings x and y, where we care about the first k ranked items only

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,5,4)
b = c(1,2,3,4,5)
k=3
kendP(a,b,k)
```

---

|        |   |
|--------|---|
| labour | <i>Voting data from the 2010 UK Labour leadership election.</i> |
|--------|---|

---

**Description**

There are five candidates 266 rankings, some of which are partial rankings

**Usage**

```
data(labour)
```

**Format**

A data frame with 234 rows and 11 variables

**Source**

[https://docs.google.com/spreadsheets/d/1e-gx4Km2ywG85kJCf\\_byJdMZvdP4QkPHGjPKy\\_me030/edit?hl=en&hl=en#gid=0](https://docs.google.com/spreadsheets/d/1e-gx4Km2ywG85kJCf_byJdMZvdP4QkPHGjPKy_me030/edit?hl=en&hl=en#gid=0)

**References**

<https://web.archive.org/web/20110101171158/http://www2.labour.org.uk/leadership-mps-and-meps>

---

|       |                       |
|-------|-----------------------|
| spear | <i>Spearman's Rho</i> |
|-------|-----------------------|

---

**Description**

Computes Spearman's rho between two full rankings.

**Usage**

```
spear(a, b)
```

**Arguments**

a, b                      integer vectors

**Value**

Returns Spearman's rho between the two rankings.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,5,4)
b = c(1,2,3,4,5)
spear(a,b)
```

---

|        |  |
|--------|--|
| spearE | <i>Spearman's Rho for rankings with Ties</i> |
|--------|--|

---

**Description**

Computes Spearman's rho between two rankings, where items with equal ranking are now permitted. The number of items placed in the *i*th category must be the same.

**Usage**

```
spearE(x, y)
```

**Arguments**

x, y                      integer vectors

**Value**

Returns Spearman's rho between the two rankings.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,2,3)
b = c(1,2,2,3,3)
spearE(a,b)
```

---

spearfoot

*Spearman's Footrule*

---

**Description**

Computes Spearman's Footrule between two full rankings.

**Usage**

```
spearfoot(a, b)
```

**Arguments**

a, b                      integer vectors

**Value**

Returns Spearman's footrule between the two rankings.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,5,4)
b = c(1,2,3,4,5)
spearfoot(a,b)
```

---

spearfootE

*Spearman's Footrule with Ties*


---

**Description**

Computes Spearman's footrule between two rankings, where items with equal ranking are now permitted. The number of items placed in the  $i$ th category must be the same.

**Usage**

```
spearfootE(x, y)
```

**Arguments**

$x$ ,  $y$                       integer vectors

**Value**

Returns Spearman's footrule between the two rankings.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,2,3)
b = c(1,2,2,3,3)
spearfootE(a,b)
```

---

spearfootG

*Spearman's Footrule for any Number of Ties*


---

**Description**

Computes Spearman's footrule between two rankings, where any number of items with equal rankings are now permitted in each ranking.

**Usage**

```
spearfootG(x, y)
```

**Arguments**

$x$ ,  $y$                       integer vectors

**Value**

Returns Spearman's footrule between the two rankings.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,2,3)
b = c(1,2,3,4,4)
spearfootG(a,b)
```

---

spearfootP

*Spearman's Footrule for Partial rankings*

---

**Description**

Computes Spearman's footrule between two partial rankings.

**Usage**

```
spearfootP(a, b, k)
```

**Arguments**

|      |                 |
|------|-----------------|
| a, b | integer vectors |
| k    | integer         |

**Value**

Returns Spearman's footrule between two partial rankings, where only the first k items have been ranked. No ties are permitted.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,5,4)
b = c(1,2,3,4,5)
k=3
spearfootP(a,b,k)
```

---

spearG

*Spearman's Rho for any Number of Ties*


---

**Description**

Computes Spearman's rho between two rankings x and y, where any number of items with equal rankings are now permitted in each ranking. The number of items ranked r for the two rankings can vary.

**Usage**

```
spearG(x, y)
```

**Arguments**

x, y                      integer vectors

**Value**

Returns Spearman's rho between two rankings x and y

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,2,3)
b = c(1,2,3,4,4)
spearG(a,b)
```

---

spearP

*Spearman's Rho for Partial rankings*


---

**Description**

Computes Spearman's rho between two partial rankings.

**Usage**

```
spearP(a, b, k)
```

**Arguments**

a, b                      integer vectors  
k                          integer

**Value**

Returns Spearman's rho between two rankings x and y, where we care about the first k ranked items only



**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,5,4)
b = c(1,2,3,4,5)
k=3
spearP(a,b,k)
```

---

ulam

*Ulam's Distance*

---

**Description**

Computes Ulam's distance between two full rankings.

**Usage**

```
ulam(a, b)
```

**Arguments**

a, b                      integer vectors

**Value**

Returns Ulam's distance between the two rankings.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,5,4)
b = c(1,2,3,4,5)
ulam(a,b)
```

---

|       |                                  |
|-------|----------------------------------|
| ulamE | <i>Ulam's distance with Ties</i> |
|-------|----------------------------------|

---

**Description**

Computes Ulam's distance between two rankings, where items with equal ranking are now permitted. The number of items placed in the *i*th category must be the same.

**Usage**

```
ulamE(x, y)
```

**Arguments**

*x*, *y*                      integer vectors

**Value**

Returns Ulam's distance between the two rankings.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,2,3)
b = c(1,2,2,3,3)
ulamE(a,b)
```

---

|       |   |
|-------|---|
| ulamG | <i>Ulam's distance for any Number of Ties</i> |
|-------|---|

---

**Description**

Computes Ulam's distance between two rankings, where any number of items with equal rankings are now permitted in each ranking.

**Usage**

```
ulamG(x, y)
```

**Arguments**

*x*, *y*                      integer vectors

**Value**

Returns Ulam's distance between the two rankings.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,2,3)
b = c(1,2,3,4,4)
ulamG(a,b)
```

---

ulamP

*Ulam's Distance for Partial rankings*

---

**Description**

Computes Ulam's distance between two partial rankings.

**Usage**

```
ulamP(a, b, k)
```

**Arguments**

|      |                 |
|------|-----------------|
| a, b | integer vectors |
| k    | integer         |

**Value**

Returns Ulam's distance between two partial rankings, where only the first k items have been ranked. No ties are permitted.

**Author(s)**

Lucy Small, <lucy.small@ucdconnect.ie>

**Examples**

```
a = c(3,1,2,5,4)
b = c(1,2,3,4,5)
k=3
ulamP(a,b,k)
```

# Index

## \*Topic **datasets**

fifa16, [5](#)

labour, [11](#)

cayley, [2](#)

cayleyE, [3](#)

cayleyP, [3](#)

distance, [4](#)

fifa16, [5](#)

ham, [6](#)

hamE, [6](#)

hamG, [7](#)

hamP, [8](#)

inv, [8](#)

kend, [9](#)

kendE, [9](#)

kendG, [10](#)

kendP, [11](#)

labour, [11](#)

spear, [12](#)

spearE, [12](#)

spearfoot, [13](#)

spearfootE, [14](#)

spearfootG, [14](#)

spearfootP, [15](#)

spearG, [16](#)

spearP, [16](#)

ulam, [17](#)

ulamE, [18](#)

ulamG, [18](#)

ulamP, [19](#)