Package ‘Dominance’

March 13, 2018

Version 1.1.0

Date 2018-10-03

Title A Package to Calculate and Visualize Dominance Hierarchies

Description Functions to calculate ADI (Average Dominance Index) and FDI (Frequency-Based Dominance Index). Functions to visualize the Data with Social Network Graphs with Dual Directions and Music Notation Graph.

Maintainer Knut Krueger <Knut.Krueger@equine-science.de>

Depends R (>= 3.1.0)

Imports igraph,chron,gdata,XLConnect

Suggests

License GPL-3

URL NA

RoxygenNote 6.0.1

NeedsCompilation no

Author Knut Krueger [aut, cre],
Konstanze Krueger [aut]

Repository CRAN

Date/Publication 2018-03-13 18:54:55 UTC

R topics documented:

Dominance-package *Dominance Calculation and Graphs in Animals*

Description

The package can calculate ADI (Average Dominance Index) and can build social network graphs with dual directions, can build a Music Notation Graph

Details

The following are sources of information on **Dominance** package:

DESCRIPTION file `library(help="Dominance")`

This file `package?Dominance`

Some help files
`ADI`
`FDI`
`Sociogram`
`Musicnotation`

Author(s)

Knut Krueger , Konstanze Krueger

Maintainer: Who to complain to <Knut.Krueger@equine-science.de>

ADI *Function ADI*

Description

A package to calculate Dominance Indices, print Soical Network Graphs and Music Notation Graphs.

Usage

```
ADI(data_sheet, bytes, ...)
```

Arguments

`data_sheet` **either** a data.frame f.e imported from a data sheet containing
"Name","item.number"
"action.from.","action.to","kind.of.action"
"name.of.action","action.number","classification","weighting"

or only "action.from.","action.to","kind.of.action"if exists actions and items

actions: with "name.of.action","action.number","classification","weighting
Weigting the factor which should be used to calculate the behavior (1 for "action.from"" wins -1 for "action.to" wins")
Setting a behaviour to 2 means it is count double
items: with "Name","item.number"

`bytes` a string where each enabled action is set to 1 and each disabled action is set to 0
Setting a behaviour to 2 means it is count double

`...` Additional parameters:

actions (data.frame) with "name.of.action","action.number","classification","weighting"; Classification 1 if "action.from"" wins; Classification 2 if "action.to" wins

Weigting the factor which should be used to calculate the behavior (1 for "action.from"" wins -1 for "action.to" wins")
Setting a behaviour to 2 means it is count double

vcolors as much colors as items, colors will returned as sorted ADI colors means color 1 = item rank 1, color 2 = item rank 2, and so on

workbook the XlConnect Workbook for the Excel file to be changed
note: The workbook must be opened before

sheet the sheet name (ADI will be added to be sure not to delete any data

savecounts if TRUE: save the counts of actions as sheet (availalbe only with workbook

saveAdi if TRUE: save the FDI as sheet (availalbe only with workbook

Value

returns a list with
ADI - the Average Dominance index
Colors - the colors supported by vcolors sorted by ADI of the items
ADI_count_matrix - the counts from which the ADI was calulated

Author(s)

Knut Krueger, `<Knut.Krueger@equine-science.de>`

References

The Construction of Dominance Order: Comparing Performance of Five Methods Using an Individual-Based Model C. K. Hemelrijk, J. Wantia and L. Gygax, Behaviour Vol. 142, No. 8 (Aug., 2005), pp. 1037-1058 http://www.jstor.org/stable/4536286
On using the DomWorld model to evaluate dominance ranking methods , de Vries, Han, Behaviour, Volume 146, Number 6, 2009 , pp. 843-869(27) http://dx.doi.org/10.1163/156853909X412241

Examples

```
{ #you can eihter use:
data_sheet=data.frame   ("action.from"=c(1,4,2,3,4,3,4,3,4,3,4,3,4,3,4),
                         "action.to"=c(4,1,1,4,3,4,3,4,3,4,3,4,3,4,3),
"kind.of.action"= c(4,1,1,4,3,4,3,4,3,4,3,4,3,4,3),stringsAsFactors=FALSE)
items= data.frame ("Name"=c("item1","item2","item3","item4","item5","item6") ,
                   "item.number"=c(1:6),stringsAsFactors=FALSE)
actions=data.frame("name.of.action"= c("leading","following","approach","bite","threat to bite",
                                      "kick","threat to kick", "chase","retreat"),
                  "action.number"=c(1:9),
                  "classification"=c(1,2,1,1,1,1,1,1,2) ,
                  "weighting"=c(1,-1,1,1,1,1,1,1,-1),stringsAsFactors=FALSE)
#all  encounters  without leading and following
bytes= "001111111"
ADI(data_sheet,items=items,actions=actions,bytes)
# or you can use a complete f.e Excel sheet
# you can save this data as basic excel sheet to work with
data(data_ADI)
bytes= "001111111"
ADI(data_ADI,bytes)
}
```

change.action.without.response

changes kind of actions for all action without response

Description

changes kind of actions for all action without response

Usage

```
change.action.without.response(data.set,action,response,newaction, ...)
```

Arguments

data.set data.frame f.e imported from a data sheet containing
"Name","item.number"
"action.from.","action.to","kind.of.action"
"name.of.action","action.number","classification","weighting"

`action` action normally with response

`response` the normal response to the action

`newaction` data.frame("name.of.action"="test","action.number"=1,"classification"=2,"weighting"=3)')

`...` workbook : the XlConnect Workbook for the Excel file to be changed
note: The workbook must be opened before

sheet: the sheet name (some random numbers will be added to be sure not to delete any data

Value

change.action.without.response returns the new data.set

Author(s)

Knut Krueger

Examples

```
data(data_ADI)
new_data=change.action.without.response(data_ADI,3,9,
          newaction=data.frame("name.of.action"="test",
          "action.number"=10,"classification"=2,"weighting"=3))
new_data
```

`data_ADI` *Demodata for ADI and FDI*

Description

Demodata to calculate an ADI and FDI

Usage

```
data(data_ADI)
```

Format

A data frame with 17 observations on the following 10 variables.

`Name` a character vector

`item.number` a numeric vector

`action.from.` a numeric vector

`action.to` a numeric vector

`kind.of.action` a numeric vector

`observation.number` a numeric vector

`name.of.action` a character vector

`action.number` a numeric vector

`classification` a numeric vector

`weighting` a numeric vector

Examples

```
data(data_ADI)
```

`data_Musicnotation` *Demodata for Musicnotation*

Description

Demodata to show an Musicnotation Graph

Usage

```
data(data_Musicnotation)
```

Format

A data frame with 15 observations on the following 11 variables.

`action.from` a numeric vector

`action.to` a numeric vector

`kind.of.action` a numeric vector

`Time` a character vector

`Name` a character vector

`item.number` a numeric vector

`dominance.order` a numeric vector

`name.of.action` a character vector

`action.number` a numeric vector

`classification` a numeric vector

`weighting` a numeric vector

Examples

```
data(data_Musicnotation)
```

`data_Network_1` *Demodata for Social network Graph*

Description

A dataset to show a bigger sociogramm

Usage

```
data(data_Network_1)
```

Format

A data frame with 800 observations on the following 16 variables.

`Name` a numeric vector

`Beschreibung` a character vector

`item.number` a numeric vector

`dominance.order` a character vector

`age` a character vector

`sex` a character vector

`action.from.` a numeric vector

`action.to` a numeric vector

`kind.of.action` a numeric vector

`time` a POSIXct

`test.2.kind.of.action` a numeric vector

`test.3.kind.of.acttion` a numeric vector

`name.of.action` a character vector

`action.number` a numeric vector

`classification` a numeric vector

`weighting` a numeric vector

Examples

```
data(data_Network_1)
```

data_Network_2 *Demodata for Social network Graph*

Description

A dataset to show a bigger sociogramm

Usage

```
data(data_Network_2)
```

Format

A data frame with 800 observations on the following 16 variables.

`Name` a numeric vector

`Beschreibung` a character vector

`item.number` a numeric vector

`dominance.order` a character vector

`age` a character vector

`sex` a character vector

`action.from.` a numeric vector

`action.to` a numeric vector

`kind.of.action` a numeric vector

`time` a POSIXct

`test.2.kind.of.action` a numeric vector

`test.3.kind.of.acttion` a numeric vector

`name.of.action` a character vector

`action.number` a numeric vector

`classification` a numeric vector

`weighting` a numeric vector

Examples

```
data(data_Network_2)
```

detect_bits *for internal use only*

Description

for internal use only

Usage

```
detect_bits(bits, set = TRUE)
```

Arguments

`bits`	
`set`	if set = TRUE returns all true bits if set = false returns all false bits

Author(s)

Knut krueger

Examples

```
print('for internal use only')
```

FDI *Function FDI*

Description

A package to calculate Dominance Indices, print Soical Network Graphs and Music Notation Graphs.

Usage

```
FDI(data_sheet, bytes, ...)
```

Arguments

`data_sheet` **either** a data.frame f.e imported from a data sheet containing
"Name","item.number"
"action.from.","action.to","kind.of.action"
"name.of.action","action.number","classification","weighting"

or only "action.from.","action.to","kind.of.action"if exists actions and items

actions: with "name.of.action","action.number","classification","weighting
items: with "Name","item.number"

`bytes` a string where each enabled action is set to 1 and each disabled action is set to 0
Setting a behaviour to 2 means it is count double

`...` **Additional parameters:**

actions (data.frame) with "name.of.action","action.number","classification","weighting"; Classification 1 if "action.from"" wins; Classification 2 if "action.to" wins

Weigting the factor which should be used to calculate the behavior (1 for "action.from"" wins -1 for "action.to" wins")
Setting a behaviour to 2 means it is count double

vcolors as much colors as items, colors will returned as sorted FDI colors means color 1 = item rank 1, color 2 = item rank 2, and so on

workbook the XlConnect Workbook for the Excel file to be changed
note: The workbook must be opened before

sheet the sheet name (FDI will be added to be sure not to delete any data

savecounts if TRUE: save the counts of actions as sheet (availalbe only with workbook

saveFDI if TRUE: save the FDI as sheet (availalbe only with workbook

Value

returns a list with
FDI - the Frequency Based Dominance index
Colors - the colors supported by vcolors sorted by FDI of the items
FDI_count_matrix - the counts from which the FDI was calulated

Author(s)

Knut Krueger, <Knut.Krueger@equine-science.de>

References

The Construction of Dominance Order: Comparing Performance of Five Methods Using an Individual-Based Model C. K. Hemelrijk, J. Wantia and L. Gygax, Behaviour Vol. 142, No. 8 (Aug., 2005), pp. 1037-1058 http://www.jstor.org/stable/4536286
On using the DomWorld model to evaluate dominance ranking methods , de Vries, Han, Behaviour, Volume 146, Number 6, 2009 , pp. 843-869(27) http://dx.doi.org/10.1163/156853909X412241

Examples

```
{ #you can eihter use:
data_sheet=data.frame   ("action.from"=c(1,4,2,3,4,3,4,3,4,3,4,3,4,3,4),
                         "action.to"=c(4,1,1,4,3,4,3,4,3,4,3,4,3,4,3),
"kind.of.action"= c(4,1,1,4,3,4,3,4,3,4,3,4,3,4,3),stringsAsFactors=FALSE)
items= data.frame ("Name"=c("item1","item2","item3","item4","item5","item6") ,
                   "item.number"=c(1:6),stringsAsFactors=FALSE)
actions=data.frame("name.of.action"= c("leading","following","approach","bite","threat to bite",
                                      "kick","threat to kick", "chase","retreat"),
                  "action.number"=c(1:9),
                  "classification"=c(1,2,1,1,1,1,1,1,2) ,
                  "weighting"=c(1,-1,1,1,1,1,1,1,-1),stringsAsFactors=FALSE)
#all  encounters  without leading and following
bytes= "001111111"
FDI(data_sheet,items=items,actions=actions,bytes)
# or you can use a complete f.e Excel sheet
# you can save this data as basic excel sheet to work with
data(data_ADI)
bytes= "001111111"
FDI(data_ADI,bytes)
   }
```

`musicnotation` *Function musicnotation*

Description

A function to visualize interaction wit a musicnotation graph.

Usage

```
Musicnotation(data_sheet, ...)
```

Arguments

`data_sheet` **either** a data.frame f.e imported from a data sheet containing
"Name","item.number"
"action.from.","action.to","kind.of.action"
"name.of.action","action.number","classification","weighting"

or only "action.from.","action.to","kind.of.action"if exists actions and items

actions: with "name.of.action","action.number","classification","weighting
items: with "Name","item.number"
Setting a behaviour to 2 means it is count double

`...` **Additional parameters:**

colors a factor of colors as much as actions

lwd line width if lwd_arrows is not used also for line width arrows # TODO check this it not working -> no show_items all items will be shown

show_items items to be shown

angel_arrows The angel aof the arrow head default 20

length_arrows the lenght of the arrow default 0.05

lwd_arrows the line width of the arrows default 1

actions_colors a vector of colors for actions f.e to show one special action

starting_time builds the graph with data bewteen starting and ending time

ending_time builds the graph with data bewteen starting and ending time

user_colors a vector of colors as much as items to show differetn colors for items

color_bits a vector of colors as much as items 1 shows the horse colored 0 in black (defined with actions_colors)

Value

returns a list with
ADI - the Average Dominance index

Author(s)

Knut Krueger, <Knut.Krueger@equine-science.de>

References

#Chase, I. D. (2006). Music notation: a new method for visualizing social interaction in animals and humans. Front Zool, 3, 18. http://dx.doi.org/10.1186%2F1742-9994-3-18

Examples

```
{ #you can eihter use:
data_sheet=data.frame   ("action.from"=c(1,4,2,3,4,3,4,3,4,3,4,3,4,3,4),
                         "action.to"=c(4,1,1,4,3,4,3,4,3,4,3,4,3,4,3),
                         "kind.of.action"= c(4,1,1,4,3,4,3,4,3,4,3,4,3,4,3),
                        "Time"=c('03:15:00','03:17:30','03:20:00','03:20:30','03:21:00',
                                 '03:21:30','03:22:00','03:22:30','03:23:00','03:23:30',
                                '03:25:00','03:25:30','03:26:00','03:26:30','03:27:00'),
                         stringsAsFactors=FALSE)
items= data.frame ("Name"=c("item1","item2","item3","item4","item5","item6") ,
                   "item.number"=c(1:6),stringsAsFactors=FALSE)
actions=data.frame("name.of.action"= c("leading","following","approach","bite","threat to bite",
                                       "kick","threat to kick", "chase","retreat"),
                  "action.number"=c(1:9),
                  "classification"=c(1,2,1,1,1,1,1,1,2) ,
                  "weighting"=c(1,-1,1,1,1,1,1,1,-1),stringsAsFactors=FALSE)
# set colors for special encounters
color= c("green","green","red","red","red","red","red","red")
Musicnotation(data_sheet=data_sheet,actions=actions,items=items,sort_dominance=TRUE)
#or you can use a complete f.e Excel sheet
#you can save this data as basic excel sheet to work with
data(data_Musicnotation)
Musicnotation(data_sheet=data_Musicnotation,sort_dominance=TRUE) }
```

search.win.lose *for internal use only*

Description

counts the wins and loses

Usage

```
search.win.lose(data_sheet, ...)
```

Arguments

data_sheet

...

Author(s)

Knut Krueger

Examples

```
print('for internal use only')
```

Sociogram *Social Network Graphs*

Description

computes social network graphs with igraph

Usage

```
Sociogram(data_sheet, bits, ...)
```

Arguments

`data_sheet` either a data.frame f.e imported from a data sheet containing
"Name","item.number"
"action.from.","action.to","kind.of.action"
"name.of.action","action.number","classification","weighting"
or only "action.from.","action.to","kind.of.action"if exists actions and items
data_sheet: with "action.from.","action.to","kind.of.action"
items with "Name","item.number"

`bits` as sting where each enabled action is set to 1 and each disabled action is set to 0

`...` Postscript: FALSE (default) or path to PS output file
soziogram_layout: layout.auto,layout.random,layout.circle,layout.sphere,layout.fruchterman.reingold,lay
curved: how much the lines between the nodes ar curved 0 ist stright, default is 0.2
scal_value: the multiplicator of the nodes, default: 1/3
linesize_add: value to add to the linesize (helpful for a graph with thin lines. default 0
log: log2 size of linewitdh, default= false
canvas.width: default 1000
canvas.height: default 800
tkplot: interactiv tkplot, default = true

Value

`sociogram` the igraph object

`counts_circles`
the count of cirles

`count_interactions`
the count of interactions

`line_size` the used linesize

`counts_circles`
vector of min #' and max line size

the last for are helpful to change circle size and #' linewidth

Author(s)

Knut Krueger, <`Knut.Krueger@equine-science.de`>

Examples

```
{
## you can eihter use:
data_sheet=data.frame   ("action.from"=c(1,4,2,3,4,3,4,3,4,3,4,3,4,3,4),
                   "action.to"=c(4,1,1,4,3,4,3,4,3,4,3,4,3,4,3),
               "kind.of.action"= c(4,1,1,4,3,4,3,4,3,4,3,4,3,4,3),stringsAsFactors=FALSE)
items= data.frame ("Name"=c("item1","item2","item3","item4","item5","item6") ,
                   "item.number"=c(1:6),stringsAsFactors=FALSE)
actions=data.frame("name.of.action"= c("leading","following","approach","bite","threat to bite",
                   "kick","threat to kick", "chase","retreat"),
                   "action.number"=c(1:9),
                   "classification"=c(1,2,1,1,1,1,1,1,2) ,
                   "weighting"=c(1,-1,1,1,1,1,1,1,-1),stringsAsFactors=FALSE)
## all  encounters  without leading and following
bytes= "001111111"
Sociogram(data_sheet,items=items,actions=actions,bytes)
## mor you can use a complete f.e Excel sheet
## you can save this data as basic excel sheet to work with
data(data_Network_1)
## set 1 for action you want to show
bytes= "0011111111000000000"
Sociogram(data_Network_1,bytes)
}
```

Index