Package ‘DecisionAnalysis’

May 22, 2020

Type Package

Title Implementation of Multi Objective Decision Analysis

Version 1.1.0

Maintainer Josh Deehr <josh.deehr@gmail.com>

BugReports https://github.com/AFIT-R/DecisionAnalysis

Date 2020-5-21


License GPL (>= 2)

Encoding UTF-8

LazyData true

Depends R (>= 2.10)

Imports stats, ggplot2, tidyr, dplyr, graphics, data.tree, gridExtra, viridisLite, Cairo, methods, qpdf, DiagrammeR

RoxygenNote 6.0.1

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Author Josh Deehr [aut, cre], Christopher Smith [aut], Jason Freels [aut], Emily Meyer [aut]

Repository CRAN

Date/Publication 2020-05-22 08:00:06 UTC
Description

The DecisionAnalysis package contains all of the necessary functions required to plot weighted and unweighted value hierarchy trees, calculate and plot linear, exponential, and categorical single attribute value functions, calculate and graph multi value attribute functions, and conduct sensitivity analysis.

Details

Start with the vignette to learn more about using the DecisionAnalysis package: browseVignettes(package = "DecisionAnalysis")

Author(s)

Maintainer: Josh Deehr <josh.deehr@gmail.com>
Authors:

- Christopher Smith <Cms3am@virginia.edu>
- Jason Freels <auburngrads@live.com>
- Emily Meyer <emily.meyer@theperducogroup.com>

See Also

Report bugs at https://github.com/AFIT-R/DecisionAnalysis
Examples

#Create a value hierarchy tree
branches<- as.data.frame(matrix(ncol=5,nrow=7))
names(branches)<-c("Level1","Level2","Level3","leaves","weights")
branches[1,]<rbind("QB","Elusiveness","Speed","Forty","0.092")
branches[2,]<rbind("QB","Elusiveness","Agility","Shuttle","0.138")
branches[3,]<rbind("QB","Size","","Height","0.096")
branches[4,]<rbind("QB","Size","","Weight","0.224")
branches[5,]<rbind("QB","Intelligence","","Wonderlic","0.07")
branches[6,]<rbind("QB","Strength","Explosiveness","Vertical","0.152")
branches[7,]<rbind("QB","Strength","Power","Broad","0.228")
value_hierarchy_tree(branches$Level1,branches$Level2,branches$Level3,
leaves=branches$leaves,weights=branches$weights)

#subset NFLcombine data from DecisionAnalysis package
qbdata <- NFLcombine[1:7,]

#Create SAVF_matrix
Height <- SAVF_exp_score(qbdata$heightinchestotal, 68, 75.21, 82)
Weight <- SAVF_exp_score(qbdata$weight, 185, 224.34, 275)
Forty <- SAVF_exp_score(qbdata$fortyyd, 4.3, 4.81, 5.4, increasing=FALSE)
Shuttle <- SAVF_exp_score(qbdata$twentyss, 3.8, 4.3, 4.9, increasing=FALSE)
Vertical <- SAVF_exp_score(qbdata$vertical, 21, 32.04, 40)
Broad <- SAVF_exp_score(qbdata$broad, 90, 111.24, 130)
Wonderlic <- SAVF_exp_score(qbdata$wonderlic, 0, 27.08, 50)
SAVF_matrix = cbind(Height, Weight, Forty, Shuttle, Vertical, Broad, Wonderlic)

#Create weights vector
weights = c(0.096, 0.224, 0.092, 0.138, 0.152, 0.228, 0.07)

#Calculate MAVF Score
MAVF_Scores(SAVF_matrix, weights, qbdata$name)

#Plot MAVF Breakout
MAVF_breakout(SAVF_matrix, weights, qbdata$name)

#Plot sensitivity analysis for shuttle criteria
sensitivity_plot(SAVF_matrix, weights, qbdata$name, 4)
Usage

```r
hello()
```

Examples

```r
hello()
```

---

**MAVF_breakout**  
*Multiple Attribute Value Function (MAVF) Breakout*

**Description**

MAVF_breakout takes a matrix of Single Attribute Value Function (SAVF) scores and shows the break out of each alternative’s weighted SAVF scores. The sum of the alternative’s weighted SAVF scores is their MAVF score.

**Usage**

```r
MAVF_breakout(SAVF_matrix, weights, names)
```

**Arguments**

- `SAVF_matrix`: Matrix of SAVF scores
- `weights`: Numeric vector of SAVF weights
- `names`: Vector of attribute names

**Value**

MAVF breakout graph

**Examples**

```r
qbdata <- NFLcombine[1:7,]
Height <- SAVF_exp_score(qbdata$heightinchestotal, 68, 75.21, 82)
Weight <- SAVF_exp_score(qbdata$weight, 185, 224.34, 275)
Forty <- SAVF_exp_score(qbdata$fortyyd, 4.3, 4.81, 5.4, increasing=FALSE)
Shuttle <- SAVF_exp_score(qbdata$twentyss, 3.8, 4.3, 4.9, increasing=FALSE)
Vertical <- SAVF_exp_score(qbdata$vertical, 21, 32.04, 40)
Broad <- SAVF_exp_score(qbdata$broad, 90, 111.24, 130)
Wonderlic <- SAVF_exp_score(qbdata$wonderlic, 0, 27.08, 50)

SAVF_matrix = cbind(Height, Weight, Forty, Shuttle,
                     Vertical, Broad, Wonderlic)
weights = c(0.096, 0.224, 0.092, 0.138, 0.152, 0.228, 0.07)

MAVF_breakout(SAVF_matrix, weights, qbdata$name)
```
**MAVF_Scores**

**Multiple Attribute Value Function (MAVF) Scores**

**Description**

- Takes a matrix of Single Attribute Value Functions (SAVF) scores and multiplies it by the weights vector to calculate the alternatives MAVF score.

**Usage**

```r
MAVF_Scores(SAVF_matrix, weights, names)
```

**Arguments**

- **SAVF_matrix**: Matrix of SAVF scores
- **weights**: Numeric vector of SAVF weights
- **names**: Vector of attribute names

**Value**

MAVF Scores

**Examples**

```r
qbdata <- NFLcombine[1:7,]
Height <- SAVF_exp_score(qbdata$heightinchestotal, 68, 75.21, 82)
Weight <- SAVF_exp_score(qbdata$weight, 185, 224.34, 275)
Forty <- SAVF_exp_score(qbdata$fortyyd, 4.3, 4.81, 5.4, increasing=FALSE)
Shuttle <- SAVF_exp_score(qbdata$twentyss, 3.8, 4.3, 4.9, increasing=FALSE)
Vertical <- SAVF_exp_score(qbdata$vertical, 21, 32.04, 40)
Broad <- SAVF_exp_score(qbdata$broad, 90, 111.24, 130)
Wonderlic <- SAVF_exp_score(qbdata$wonderlic, 0, 27.08, 50)

SAVF_matrix = cbind(Height, Weight, Forty, Shuttle,
                     Vertical, Broad, Wonderlic)
weights = c(0.096, 0.224, 0.092, 0.138, 0.152, 0.228, 0.07)

MAVF_Scores(SAVF_matrix, weights, qbdata$name)
```
NFL Scouting Combine data

**Description**

A dataset containing the biographical data and scores for 4947 NFL draft candidates that attended the NFL Scouting Combine from 1999 to 2015.

**Usage**

NFLcombine

**Format**

A data frame with 4947 rows and 26 variables:

- **year**: Year that the NFL draft candidate attended the NFL combine event
- **name**: First and last name of the NFL draft candidate
- **firstname**: First name of the NFL draft candidate
- **lastname**: Last name of the NFL draft candidate
- **position**: Position of the NFL draft candidate
- **heightfeet**: Candidate’s height, only the feet portion
- **heightinches**: Candidate’s height, only the inches portion
- **heightinchestotal**: Candidate’s total height in inches
- **weight**: Total weight in lbs
- **arms**: Candidate’s arm length in inches
- **hands**: Candidate’s hand size in inches
- **forty**: Time in seconds to run forty yards
- **twenty**: Time in seconds to run twenty yards
- **ten**: Time in seconds to run ten yards
- **twentyss**: Time in seconds to complete the twenty yard shuttle sprint
- **threecone**: Time in seconds to complete the three cone drill
- **vertical**: Height candidate jumped vertically in inches
- **broad**: Distance traveled during broad jump in inches
- **bench**: Number of repetitions a candidate bench pressed 225lbs
- **round**: The round the candidate was selected in the draft
- **college**: College the candidate attended
- **pick**: The candidate’s pick number in the round that they got drafted, followed by the candidate’s overall pick number for that year’s NFL draft
- **pickround**: The candidate’s pick number in the round that they got drafted
- **picktotal**: The candidate’s overall pick number for that year’s NFL draft
- **wonderlic**: Raw score received on the Wonderlic test
- **nflgrade**: The grade the candidate is given on NFL.com
SAVF_calc_rho

Source

http://www.nflsavant.com/about.php

---

**SAVF_calc_rho**

*Single Attribute Value Function (SAVF) Calculate Rho*

---

**Description**

: Calculates rho for an exponential function.

**Usage**

```
SAVF_calc_rho(x_low, x_mid, x_high, increasing = TRUE)
```

**Arguments**

- `x_low`  
  Lowest value
- `x_mid`  
  Midpoint value
- `x_high`  
  Highest value
- `increasing`  
  TRUE=increasing, FALSE=decreasing, Default: TRUE

**Details**

For \( Z=\frac{(x_{mid} - x_{low})}{(x_{high} - x_{low})} \), \( Z \) can not be in \((0.51, 0.49)\).

**Value**

Rho

**Examples**

```
SAVF_calc_rho(0, 90, 150, FALSE)
```
SAVF_categorical_score

*Single Attribute Value Function (SAVF) Categorical Score*

Description

: Calculates the Single Attribute Value Function (SAVF) score for a categorical value.

Usage

```r
SAVF_categorical_score(x, categories, weights)
```

Arguments

- `x` : Attribute raw value
- `categories` : Vector of categories
- `weights` : Numeric vector of category weights

Value

Categorical SAVF Score

Examples

```r
SAVF_categorical_score("Tom", c("Tom", "Bill", "Jerry"), c(0.1, 0.25, 0.65))
```

SAVF_cat_plot

*Single Attribute Value Function (SAVF) Categorical Plot*

Description

: Plots the categorical Single Attribute Value Function (SAVF) graph. Categories may be any value, but category scores must be numeric. The function checks to ensure the total of scores sums to one.

Usage

```r
SAVF_cat_plot(categories, scores, fillcolor = "blue")
```

Arguments

- `categories` : Vector of categories
- `scores` : Numeric vector of category scores
- `fillcolor` : Fill color for the chart, default is blue
**Description**

Plots an increasing or decreasing exponential Single Attribute Value Function (SAVF) curve. It calls the SAVF_calc_rho and SAVF_exp_score functions and plots your score on the curve with a blue dot.

**Usage**

```
SAVF_exp_plot(x, x_low, x_mid, x_high, increasing = TRUE)
```

**Arguments**

- `x` Attribute raw value
- `x_low` Lowest Value
- `x_mid` Midpoint value
- `x_high` Highest value
- `increasing` TRUE=increasing, FALSE=decreasing, Default: TRUE

**Details**

For $Z=((x_{mid} - x_{low}) / (x_{high} - x_{low}))$, $Z$ can not be in (0.51, 0.49)

**Value**

Exponential SAVF curve with attribute plotted

**Examples**

```
SAVF_exp_plot(90, 0, 120, 150)
```
**SAVF_exp_score**  
*Single Attribute Value Function (SAVF) Exponential Score*

**Description**
: Calculates the Single Attribute Values Function (SAVF) score for an exponentially increasing or decreasing function. It calls the SAVF_calc_rho function, so knowing rho beforehand is not necessary.

**Usage**

```r
SAVF_exp_score(x, x_low, x_mid, x_high, increasing = TRUE)
```

**Arguments**

- `x` : Attribute raw value
- `x_low` : Lowest value
- `x_mid` : Midpoint value
- `x_high` : Highest value
- `increasing` : TRUE=increasing, FALSE=decreasing, Default: TRUE

**Details**

For $Z=((x_{mid} - x_{low}) / (x_{high} - x_{low}))$, $Z$ can not be in $(0.51,0.49)$

**Value**

Exponential SAVF Score

**Examples**

```r
SAVF_exp_score(70, 0, 90, 150, FALSE)
```

---

**SAVF_linear_plot**  
*Single Attribute Value Function (SAVF) Linear Plot*

**Description**
: Plots the linear Single Attribute Value Function (SAVF) graph for an increasing or decreasing function. It calls the SAVF_linear_score function to calculate the score based on the midpoint value method and plots it with a blue dot.

**Usage**

```r
SAVF_linear_plot(x, x_low, x_mid, x_high, increasing = TRUE)
```
Arguments

- **x**: Attribute raw value
- **x_low**: Lowest value
- **x_mid**: Midpoint value
- **x_high**: Highest value
- **increasing**: TRUE=increasing, FALSE=decreasing, Default: TRUE

Value

Linear SAVF curve with attribute plotted

Examples

```r
SAVF_linear_plot(10, 0, 25, 100, FALSE)
```

---

Description

Calculates the Single Attribute Value Function (SAVF) score for a linearly increasing or decreasing function. It calculates the score based on the midpoint value method.

Usage

```r
SAVF_linear_score(x, x_low, x_mid, x_high, increasing = TRUE)
```

Arguments

- **x**: Attribute raw value
- **x_low**: Lowest value
- **x_mid**: Midpoint value
- **x_high**: Highest value
- **increasing**: TRUE=increasing, FALSE=decreasing, Default: TRUE

Value

Linear SAVF Score

Examples

```r
SAVF_linear_score(10, 0, 25, 100, FALSE)
```
**sensitivity_plot**

---

### Description

Takes a matrix of Single Attribute Value Function (SAVF) scores and shows how each alternative’s MAVF scores change as the weight for that criteria changes from zero to one. The vertical black line represents the current weight.

### Usage

```r
sensitivity_plot(SAVF_matrix, weights, names, criteria, title = TRUE)
```

### Arguments

- **SAVF_matrix**: Matrix of SAVF scores
- **weights**: Numeric vector of SAVF weights
- **names**: The names of the alternatives
- **criteria**: Numeric value equal to the column number of the SAVF_matrix that contains the desired criteria to conduct sensitivity analysis on
- **title**: True=The title is the column name associated with the selected criteria, False=no title, Default: TRUE

### Value

Sensitivity Analysis graph

### Examples

```r
library(dplyr)

qbdata <- NFLcombine %>%
  filter(year == 2011, position == 'QB', wonderlic != 0) %>%
  select(c(2, 8, 9, 12, 15, 17, 18, 25, 20))
qbdata[qbdata == 0] = NA

Height <- SAVF_exp_score(qbdata$heightinchestotal, 68, 75.21, 82)
Weight <- SAVF_exp_score(qbdata$weight, 185, 224.34, 275)
Forty <- SAVF_exp_score(qbdata$fortyyd, 4.3, 4.81, 5.4, increasing=FALSE)
Shuttle <- SAVF_exp_score(qbdata$twentyss, 3.8, 4.3, 4.9, increasing=FALSE)
Vertical <- SAVF_exp_score(qbdata$vertical, 21, 32.04, 40)
Broad <- SAVF_exp_score(qbdata$broad, 90, 111.24, 130)
Wonderlic <- SAVF_exp_score(qbdata$wonderlic, 0, 27.08, 50)

SAVF_matrix = cbind(Height, Weight, Forty, Shuttle, Vertical, Broad, Wonderlic)
weights = c(0.096, 0.224, 0.092, 0.138, 0.152, 0.228, 0.07)
```
sensitivity_plot(SAVF_matrix, weights, qbdata$name, 4)

value_hierarchy_tree

---

Value Hierarchy Tree

Description

: Plots a value hierarchy tree

Usage

value_hierarchy_tree(..., leaves, weights, nodestyle = "filled, rounded", nodeshape = "box", nodefillcolor = "white", nodefontname = "helvetica", nodefontcolor = "black", leavesshape = "egg", leavesfillcolor = "gray", leavesfontcolor = "black", leavesfontname = "helvetica")

Arguments

... One or more character vectors containing a single level of nodes. The character vector containing the end nodes should not be entered here. If there isn’t a node for a level of the branch, it should be entered as ""
leaves Character vector of values containing the last node of the branches
weights Character or numeric vector of weights associated with the end node of the branches (Optional)
nodestyle Style of the nodes, default is filled, rounded
nodeshape Shape of the nodes, default is box
nodefillcolor Fill color of the nodes, default is white
nodefontname Font of the nodes, default is helvetica
nodefontcolor Font color of the nodes, default is black
leavesshape Shape of the leaves, default is egg
leavesfillcolor Fill color of the leaves, default is gray
leavesfontcolor Font color of the leaves, default is black
leavesfontname Font of the leaves, default is helvetica

Value

Value hierarchy tree plot
Examples

```r
branches <- as.data.frame(matrix(ncol=5, nrow=7))
names(branches) <- c("Level1", "Level2", "Level3", "leaves", "weights")
branches[1,] <- rbind("QB", "Elusiveness", "Speed", "Forty", "0.092")
branches[2,] <- rbind("QB", "Elusiveness", "Agility", "Shuttle", "0.138")
branches[3,] <- rbind("QB", "Size", "", "Height", "0.096")
branches[4,] <- rbind("QB", "Size", "", "Weight", "0.224")
branches[5,] <- rbind("QB", "Intelligence", "", "Wonderlic", "0.07")
branches[6,] <- rbind("QB", "Strength", "Explosiveness", "Vertical", "0.152")
branches[7,] <- rbind("QB", "Strength", "Power", "Broad", "0.228")

value_hierarchy_tree(branches$Level1, branches$Level2, branches$Level3,
leaves=branches$leaves, weights=branches$weights)
```
Index

*Topic **datasets**
  NFLcombine, 6

DecisionAnalysis
  (DecisionAnalysis-package), 2
DecisionAnalysis-package, 2

hello, 3

MAVF_breakout, 4
MAVF_Scores, 5

NFLcombine, 6

SAVF_calc_rho, 7
SAVF_cat_plot, 8
SAVF_categorical_score, 8
SAVF_exp_plot, 9
SAVF_exp_score, 10
SAVF_linear_plot, 10
SAVF_linear_score, 11
sensitivity_plot, 12

value_hierarch_tree, 13