Package ‘cemtool’

July 23, 2019

Type Package

Title Step By Step Guide to Build a Markov Model

Version 0.4

Author S.R.W. Wijn

Maintainer S.R.W. Wijn <stan.wijn@radboudumc.nl>

Description Markov models are increasingly used to determine the cost-effectiveness of medical innovations. However, medical researchers still use closed source software to build such models. The ceom-package was build to guide researchers to build their own Markov model in R.The current version supports Markov models that consist of 3 to 6 health-states.

License GPL-3

Depends shiny, rhandsontable, diagram, shinydashboard, utils

Encoding UTF-8

LazyData false

RoxygenNote 6.1.1

URL https://github.com/StanWijn/cemtool

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

cemprob — *Second step of the cemtool: Define the transition probabilities, cost and utility values*

Description

The `cemtool()` package provides a step-by-step tool to guide users in building a default Markov model. The tool guides the user though the steps of the development of a Markov model and will present the final result using graphs and tables. The only prerequisite is that the user knows the structure of the model and the transition probabilities, no calculations or coding is required.

Usage

```
cemprob(HS = cemtool.env$HS, HS1 = cemtool.env$HS1,
  HS2 = cemtool.env$HS2, HS3 = cemtool.env$HS3,
  HS4 = cemtool.env$HS4, HS5 = cemtool.env$HS5,
  dead = cemtool.env$dead, n.t = cemtool.env$n.t,
  control = cemtool.env$control,
  intervention = cemtool.env$intervention)
```

Arguments

`HS`	Number of healthstates
`HS1`	String with name of healthstate 1
`HS2`	String with name of healthstate 2
`HS3`	String with name of healthstate 3
`HS4`	String with name of healthstate 4
`HS5`	String with name of healthstate 5
`dead`	String with name of absorption / death state
`n.t`	Number of cycles
`control`	String with name of the usual care strategy
`intervention`	String with name of the intervention strategy

Details

All input arguments can be generated and saved with the `cemtool()` function. (cemtool.env <- cemtool()) There are multiple software systems that can be used to build Markov models for cost-effectiveness analyses like TreeAge, Excel or R. Although there are numerous advantages to use R over the others, the biggest downside is the steep learning curve from R. The `cemtool()` package aims to close this gap by introducing a step-by-step tool to guide users in building a default Markov models. The tool guides the user though the steps of the development of a Markov model and will present the final result using graphs and tables. The only prerequisite is that the user knows the structure of the model and the transition probabilities, no calculations or coding is required.

Value

The following variables will be created in the saved in the cemtool enviroment:

— Empty Markov trace matrices for both strategies (m.M and m.M_treatment)

— A dataframe with the modelinput (modelinput)

— Discount rate for costs (d.rc) and effects (d.re)

The function will automatically run `cemtpm()` after finishing

Author(s)

S.R.W. Wijn MSc <stan.wijn@radboudumc.nl>

Examples

```
## Not run:
cemtool() # Start from stratch (clear the cemtool environment from the Global Environment)
cemprob() # Start from the second phase (definding the parameters)
cemtpm()  # Start from the third phase (modify the transition probability matrix)
cemrun()  # Run the model with the current m.M markov trace and m.P transition probability matrix

cemtool.env <- cemtool() # To save all input for further modification

## End(Not run)
```

---

cemrun *Run the Markov model with input generated from* `cemtool()`*,* `cemprob()`*,*`cemtpm()`

Description

The `cemtool()` package provides a step-by-step tool to guide users in building a default Markov model. The tool guides the user though the steps of the development of a Markov model and will present the final result using graphs and tables. The only prerequisite is that the user knows the structure of the model and the transition probabilities, no calculations or coding is required.

Usage

```
cemrun(HS = cemtool.env$HS, HS1 = cemtool.env$HS1,
  HS2 = cemtool.env$HS2, HS3 = cemtool.env$HS3,
  HS4 = cemtool.env$HS4, HS5 = cemtool.env$HS5,
  dead = cemtool.env$dead, n.t = cemtool.env$n.t,
  control = cemtool.env$control,
  intervention = cemtool.env$intervention, d.rc = cemtool.env$d.rc,
  d.re = cemtool.env$d.rc, m.M = cemtool.env$m.M,
  m.M_treatment = cemtool.env$m.M_treatment, m.P = cemtool.env$m.P,
  m.P_treatment = cemtool.env$m.P_treatment,
  modelinput = cemtool.env$modelinput)
```

Arguments

`HS`	Number of healthstates
`HS1`	String with name of healthstate 1
`HS2`	String with name of healthstate 2
`HS3`	String with name of healthstate 3
`HS4`	String with name of healthstate 4
`HS5`	String with name of healthstate 5
`dead`	String with name of absorption / death state
`n.t`	Number of cycles
`control`	String with name of the usual care strategy
`intervention`	String with name of the intervention strategy
`d.rc`	Discount rate for costs
`d.re`	Discount rate for effects
`m.M`	Matrix showing the Markov trace of usual care, nrow = n.t + 1, ncol = HS
`m.M_treatment`	Matrix showing the Markov trace of intervention strategy, nrow = n.t + 1, ncol = HS
`m.P`	Matrix showing the transition probability matrix of the usual care, nrow = HS, ncol = HS
`m.P_treatment`	Matrix showing the transition probability matrix of the intervention strategy, nrow = HS, ncol = HS
`modelinput`	Matrix with 2 rows that include all the transition probabilities, costs and effects.

Details

All input arguments can be generated and saved with the `cemtool()` function. (cemtool.env <- cemtool()) There are multiple software systems that can be used to build Markov models for cost-effectiveness analyses like TreeAge, Excel or R. Although there are numerous advantages to use R over the others, the biggest downside is the steep learning curve from R. The `cemtool()` package aims to close this gap by introducing a step-by-step tool to guide users in building a default Markov models. The tool guides the user though the steps of the development of a Markov model and will present the final result using graphs and tables. The only prerequisite is that the user knows the structure of the model and the transition probabilities, no calculations or coding is required.

Value

The following variables will be created in the Global Enviroment:

— Full Markov trace (m.M and m.M_treatment)

— Calculate the costs and effects for both strategies.

— Results are saved (table_output) and shown in the console.

— The model structure and markov trace are both plotted and saved (plot1 and plot2)

Author(s)

S.R.W. Wijn MSc <stan.wijn@radboudumc.nl>

Examples

```
## Not run:
cemtool() # Start from stratch (clear the cemtool environment from the Global Environment)
cemprob() # Start from the second phase (definding the parameters)
cemtpm()  # Start from the third phase (modify the transition probability matrix)
cemrun()  # Run the model with the current m.M markov trace and m.P transition probability matrix

cemtool.env <- cemtool() # To save all input for further modification

## End(Not run)
```

cemtool *Default function to start the cemtool package*

Description

The `cemtool()` package provides a step-by-step tool to guide users in building a default Markov model. The tool guides the user though the steps of the development of a Markov model and will present the final result using graphs and tables. The only prerequisite is that the user knows the structure of the model and the transition probabilities, no calculations or coding is required.

Usage

```
cemtool()
```

Details

There are multiple software systems that can be used to build Markov models for cost-effectiveness analyses like TreeAge, Excel or R. Although there are numerous advantages to use R over the others, the biggest downside is the steep learning curve from R. The `cemtool()` package aims to close this gap by introducing a step-by-step tool to guide users in building a default Markov models. The tool guides the user though the steps of the development of a Markov model and will present the final result using graphs and tables. The only prerequisite is that the user knows the structure of the model and the transition probabilities, no calculations or coding is required.

– Required knowledge / information:

Model structure
The structure of the Markov model: how many healthstates, the names of the healthstates etc.

Transition probabilities
The probabilities to move from one healthstate to another

Costs
The costs for a patient to be on the healhstates for one cycle / year

Effects
The effect size associated with the healthstate. For QALY a value between 1 and 0, from perfect health to death.

Value

The following variables will be created in the Global Enviroment:

— Number of healhstates (HS)

— Names of the healthstates (HS1, HS2 ... HSn, dead)

— Names of both the intervention (intervention) and usual care strategy (control)

— Number of cycles (n.t)

— Vector of the strategy names (Strategies) and a vector of the healhstate names (v.n).

The function will automatically run `cemprob()` after finishing

Author(s)

S.R.W. Wijn MSc <stan.wijn@radboudumc.nl>

Examples

```
## Not run:
cemtool() # Start from stratch (clear the cemtool environment from the Global Environment)
cemprob() # Start from the second phase (definding the parameters)
cemtpm()  # Start from the third phase (modify the transition probability matrix)
cemrun()  # Run the model with the current m.M markov trace and m.P transition probability matrix

cemtool.env <- cemtool() # To save all input for further modification

## End(Not run)
```

---

cemtpm — *OPTIONAL: Third step of the cemtool: Alter the transition probability matrix*

Description

The `cemtool()` package provides a step-by-step tool to guide users in building a default Markov model. The tool guides the user though the steps of the development of a Markov model and will present the final result using graphs and tables. The only prerequisite is that the user knows the structure of the model and the transition probabilities, no calculations or coding is required.

Usage

```
cemtpm(HS = cemtool.env$HS, HS1 = cemtool.env$HS1,
  HS2 = cemtool.env$HS2, HS3 = cemtool.env$HS3,
  HS4 = cemtool.env$HS4, HS5 = cemtool.env$HS5,
  dead = cemtool.env$dead, n.t = cemtool.env$n.t,
  control = cemtool.env$control,
  intervention = cemtool.env$intervention, d.rc = cemtool.env$d.rc,
  d.re = cemtool.env$d.rc, m.M = cemtool.env$m.M,
  m.M_treatment = cemtool.env$m.M_treatment, m.P = cemtool.env$m.P,
  m.P_treatment = cemtool.env$m.P_treatment,
  modelinput = cemtool.env$modelinput)
```

Arguments

`HS`	Number of healthstates
`HS1`	String with name of healthstate 1
`HS2`	String with name of healthstate 2
`HS3`	String with name of healthstate 3
`HS4`	String with name of healthstate 4
`HS5`	String with name of healthstate 5
`dead`	String with name of absorption / death state
`n.t`	Number of cycles
`control`	String with name of the usual care strategy
`intervention`	String with name of the intervention strategy
`d.rc`	Discount rate for costs
`d.re`	Discount rate for effects
`m.M`	Matrix showing the Markov trace of usual care, nrow = n.t + 1, ncol = HS
`m.M_treatment`	Matrix showing the Markov trace of intervention strategy, nrow = n.t + 1, ncol = HS
`m.P`	Matrix showing the transition probability matrix of the usual care, nrow = HS, ncol = HS
`m.P_treatment`	Matrix showing the transition probability matrix of the intervention strategy, nrow = HS, ncol = HS
`modelinput`	Matrix with 2 rows that include all the transition probabilities, costs and effects.

Details

All input arguments can be generated and saved with the `cemtool()` function. (cemtool.env <- cemtool()) There are multiple software systems that can be used to build Markov models for cost-effectiveness analyses like TreeAge, Excel or R. Although there are numerous advantages to use R over the others, the biggest downside is the steep learning curve from R. The `cemtool()` package aims to close this gap by introducing a step-by-step tool to guide users in building a default Markov models. The tool guides the user though the steps of the development of a Markov model and will present the final result using graphs and tables. The only prerequisite is that the user knows the structure of the model and the transition probabilities, no calculations or coding is required.

Value

The following variables will be created in the saved in the cemtool enviroment:

— Modified transition probability matrix for both strategies (M.P and m.P_treatment).

The function will automatically run `cemrun()` after finishing

Author(s)

S.R.W. Wijn MSc <stan.wijn@radboudumc.nl>

Examples

```
## Not run:
cemtool() # Start from stratch (clear the cemtool environment from the Global Environment)
cemprob() # Start from the second phase (definding the parameters)
cemtpm()  # Start from the third phase (modify the transition probability matrix)
cemrun()  # Run the model with the current m.M markov trace and m.P transition probability matrix

cemtool.env <- cemtool() # To save all input for further modification

## End(Not run)
```

Index