Our people.
Your success.

BresMed™

R-HTA in LMIC's: The Potential of R Shiny User Interfaces to Support Health Economic Decision Making

Robert Smith[1,2,3,4] & Rose Hart [1]

23/02/2021

1) BresMed Health Solutions, Sheffield
2) School of Health and Related Research, University of Sheffield, UK.
3) Dark Peak Analytics Ltd, Sheffield, UK
4) UK Health Security Agency, DHSC.

Before we start …

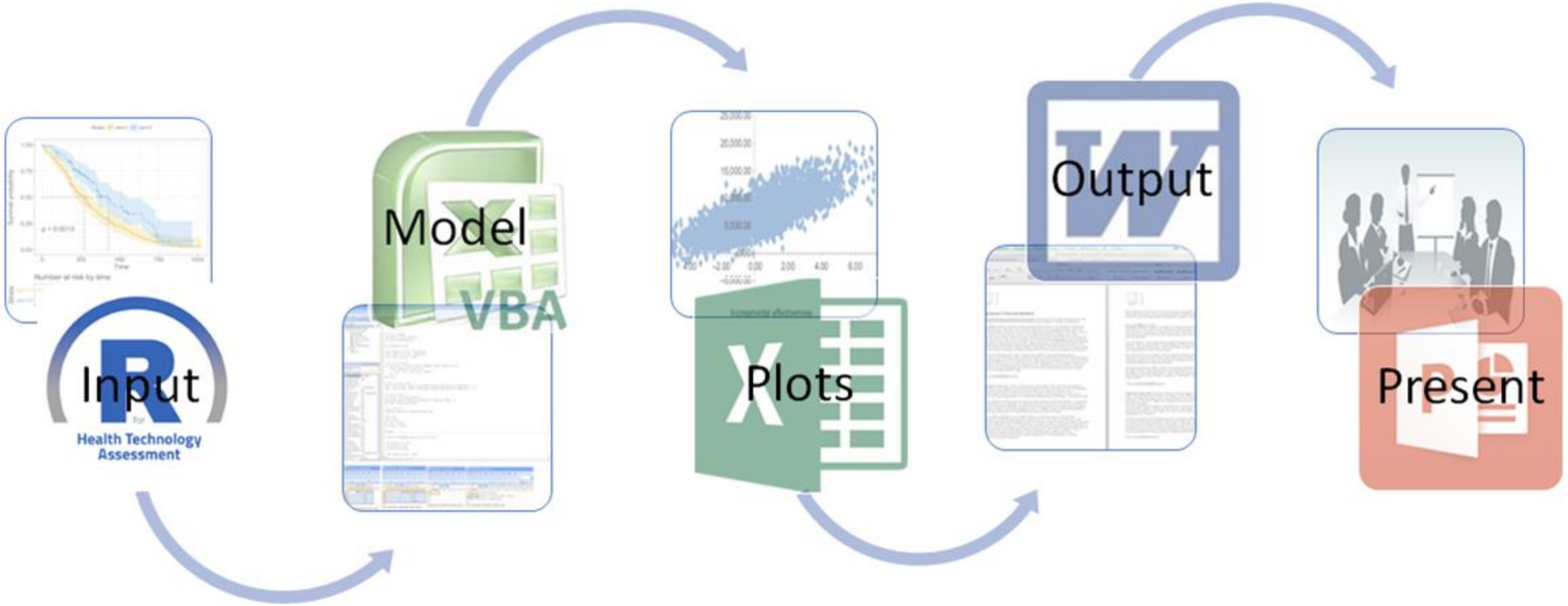
Disclaimer:

The views in this presentation are those of the author and *Dark Peak Analytics*, not of the *University of Sheffield*, *BresMed* or the *UK Health Security Agency*.

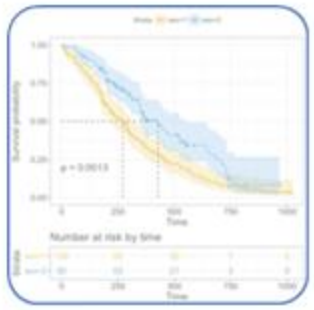
Grant information:

Robert Smith and Paul Schneider are joint funded by the Wellcome Trust Doctoral Training Centre in Public Health Economics and Decision Science [108903] and the University of Sheffield.

Current Process

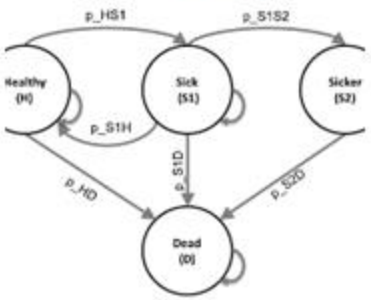


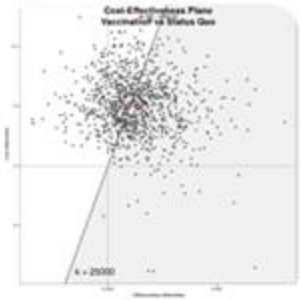
Future Process



R for Health Technology Assessment

hesim health economic simulation modeling





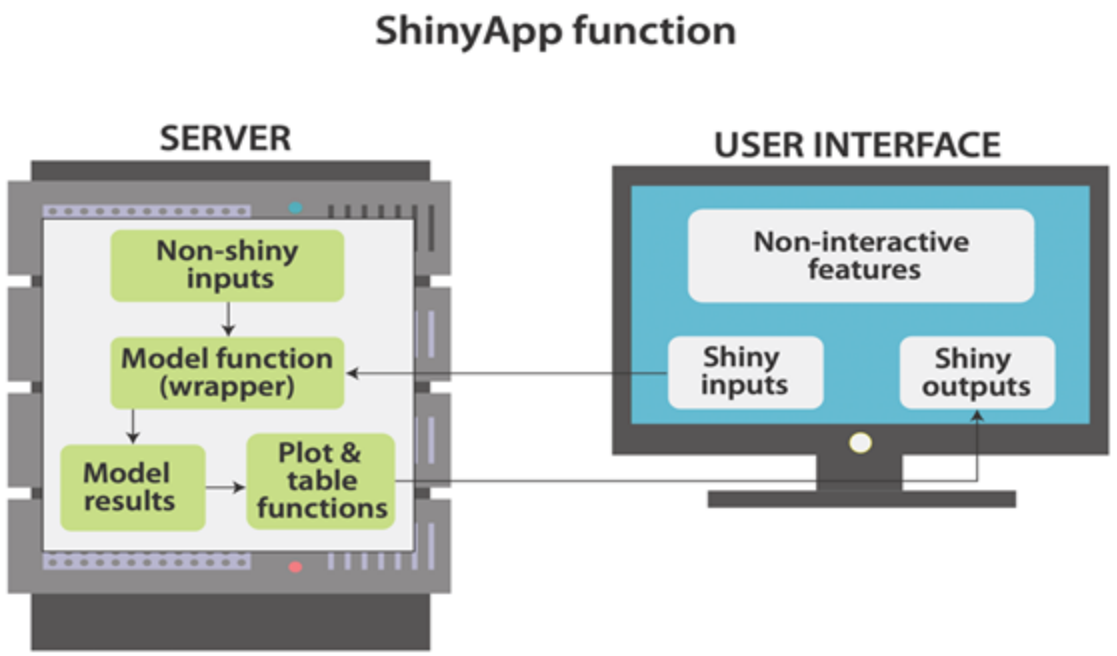
ggplot2

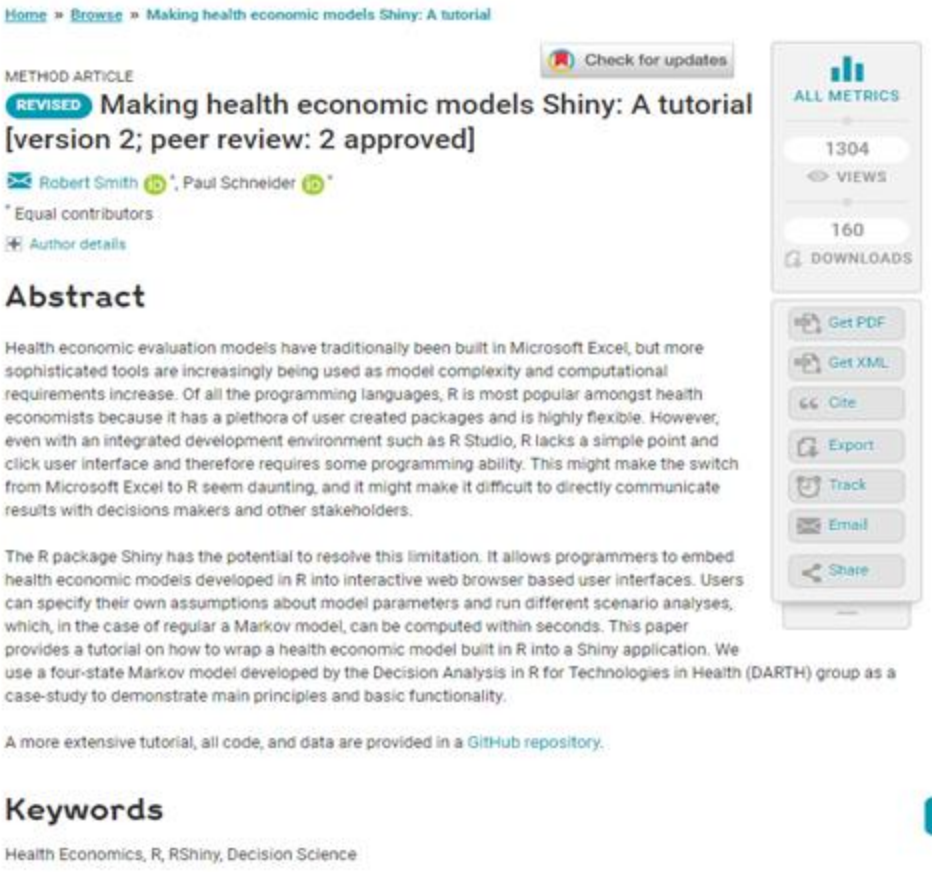
rmarkdown

Shiny

Future Process

Open-source tutorial





Home » Browse » Making health economic models Shiny: A tutorial

Check for updates

METHOD ARTICLE

REVISED Making health economic models Shiny: A tutorial [version 2; peer review: 2 approved]

Robert Smith *, Paul Schneider *

* Equal contributors

Author details

ALL METRICS 1304 VIEWS 160 DOWNLOADS

Get PDF Get XML Cite Export Track Email Share

Abstract

Health economic evaluation models have traditionally been built in Microsoft Excel, but more sophisticated tools are increasingly being used as model complexity and computational requirements increase. Of all the programming languages, R is most popular amongst health economists because it has a plethora of user created packages and is highly flexible. However, even with an integrated development environment such as R Studio, R lacks a simple point and click user interface and therefore requires some programming ability. This might make the switch from Microsoft Excel to R seem daunting, and it might make it difficult to directly communicate results with decisions makers and other stakeholders.

The R package Shiny has the potential to resolve this limitation. It allows programmers to embed health economic models developed in R into interactive web browser based user interfaces. Users can specify their own assumptions about model parameters and run different scenario analyses, which, in the case of regular a Markov model, can be computed within seconds. This paper provides a tutorial on how to wrap a health economic model built in R into a Shiny application. We use a four-state Markov model developed by the Decision Analysis in R for Technologies in Health (DARTH) group as a case-study to demonstrate main principles and basic functionality.

A more extensive tutorial, all code, and data are provided in a GitHub repository.

Keywords

Health Economics, R, RShiny, Decision Science

Paper: https://wellcomeopenresearch.org/articles/5-69
Code: https://github.com/RobertASmith/healthecon_shiny

Open-source tutorial

Sick Sicker Model in Shiny

Treatment Cost

200

PSA runs

1000

initial age

10 25 80

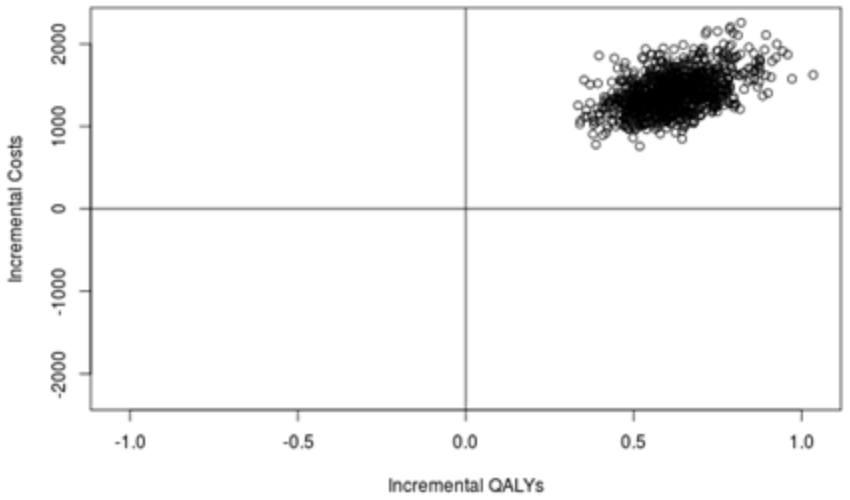
10 17 24 31 38 45 52 59 66 73 80

Run / update model

Results Table

Option	QALYs	Costs	Inc.QALYs	Inc.Costs	ICER
Treatment	18.59	100441.67	0.62	1406.24	2324.54
No Treatment	17.97	99035.43	NA	NA	NA

Cost-effectiveness Plane

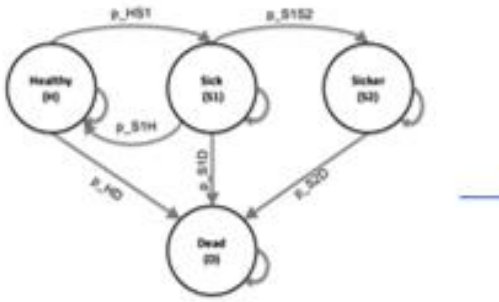


https://robertasmith.shinyapps.io/sick_sicker

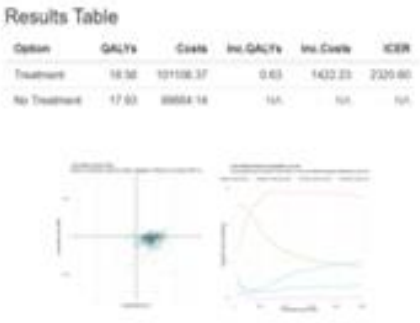
Open-source tutorial

Inputs → Function → Outputs

Parameters		
c_s1	cost1	3
c_s2	cost2	5
c_H	cost3	6
dr	Dis_rate	0.035
n_sim	No. psa	1000



Results Table



Open-source tutorial

Inputs → Function → Outputs

Parameters		
c_s1	cost1	3
c_s2	cost2	5
c_H	cost3	6
dr	Dis_rate	0.035
n_sim	No. psa	1000

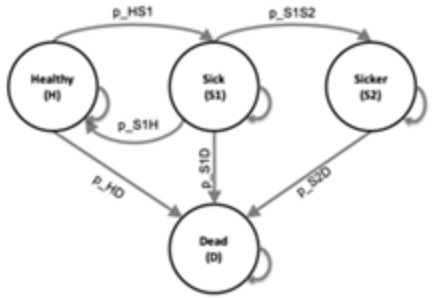
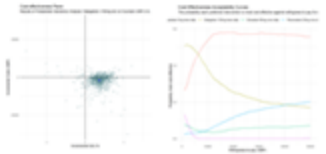


Figure 1: State-transition diagram of the time-independent Sick-Sicker cohort state-transition model with the name of the health states and possible transitions with their corresponding transition probabilities.

Results Table

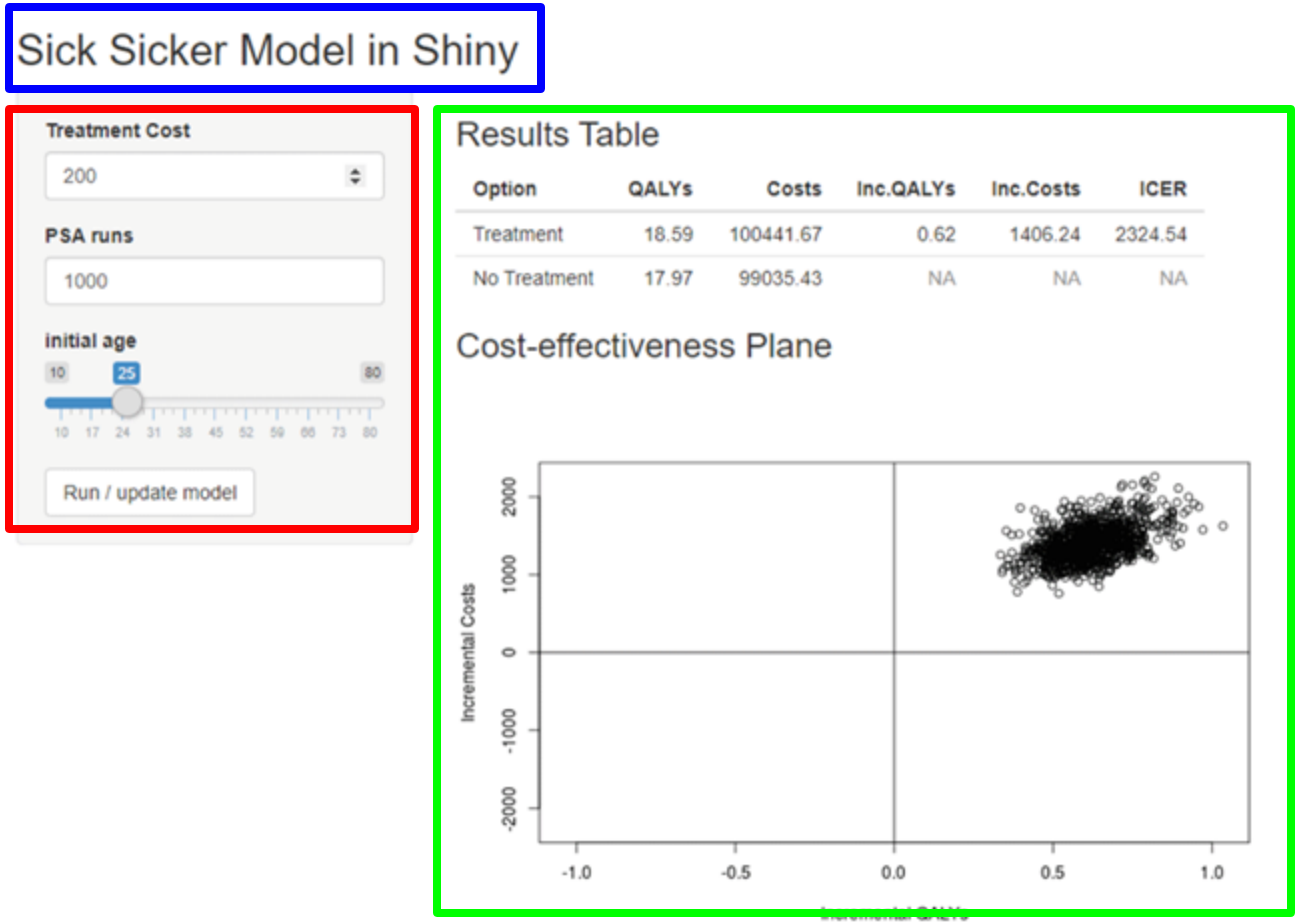
Option	GALYs	Costs	Inc.GALYs	Inc.Costs	ICER
Treatment	18.56	101106.37	0.63	1422.23	2320.60
No Treatment	17.93	99684.14	NA	NA	NA



BresMed™

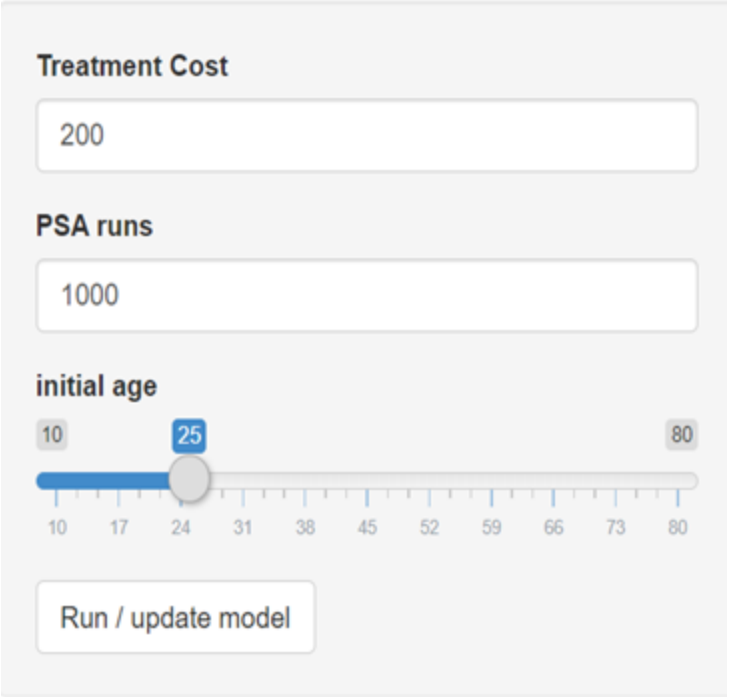
UI code

```
ui <- fluidPage (     # creates empty page

  # title of app
  titlePanel("Sick Sicker Model in Shiny"),

   # layout is a sidebar-layout
   sidebarLayout(

  # open sidebar panel

      < SIDEBAR PANEL CODE >

# open main panel

      < MAIN PANEL CODE >

         )# close sidebarlayout

)  # close UI fluidpage
```



Sidebar Panel Code

```
sidebarPanel( # open sidebar panel

    numericInput(inputId = "SI_c_Trt",
                 label = "Treatment Cost",
                 value = 200,
                 min = 0,
                 max = 400),

    numericInput(inputId = "SI_n_sim",
                 label = "PSA runs",
                 value = 1000,
                 min = 0,
                 max = 400),

    sliderInput(inputId = "SI_n_age_init",
                label = "Initial Age",
                value = 25,
                min = 10,
                max = 80),

    # action button runs model when pressed
    actionButton(inputId = "run_model",
                 label   = "Run model")

 )   # close sidebarPanel
```



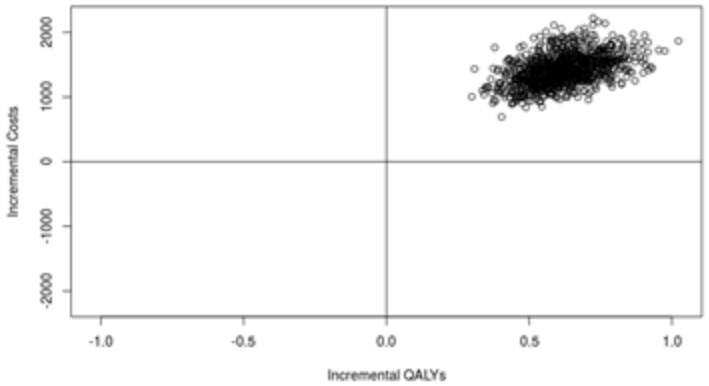
Main Panel Code

```
  mainPanel(

# heading (results table)
    h3("Results Table"),

# tableOutput id = icer_table, from server
    tableOutput(outputId = "SO_icer_table"),

# heading (Cost effectiveness plane)
    h3("Cost–effectiveness Plane"),

# plotOutput id = SO_CE_plane, from server
    plotOutput(outputId = "SO_CE_plane")

            ) # close mainpanel
```

Results Table

Option	QALYs	Costs	Inc.QALYs	Inc.Costs	ICER
Treatment	18.61	101016.42	0.62	1412.82	2335.56
No Treatment	17.99	99603.60	NA	NA	NA

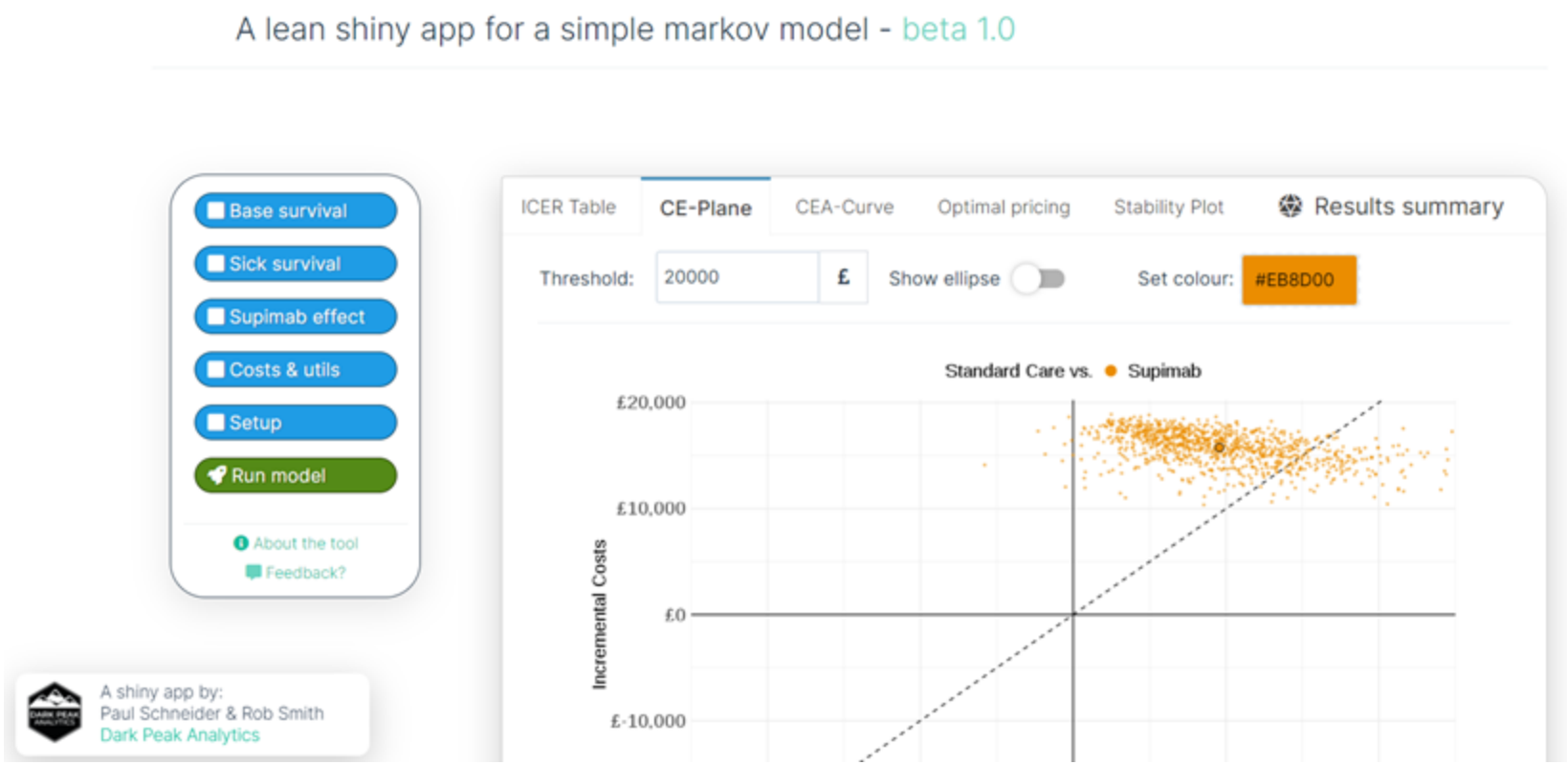
Cost-effectiveness Plane



Server Code

```
server <- function(input, output){

 observeEvent(input$run_model, # WHEN ACTION BUTTON PRESSED
              ignoreNULL = F, {

# Run  model function with Shiny inputs
 df_model_res = f_wrapper(c_Trt = input$SI_c_Trt,
                          n_age_init = input$SI_n_age_init,
                                   n_sim = input$SI_n_sim)

#-- CREATE COST EFFECTIVENESS TABLE --#
# renderTable continuously updates table
 output$SO_icer_table <- renderTable({  < ICER TABLE FUNCTION >    }) # table plot end.

#-- CREATE COST EFFECTIVENESS PLANE --#
# render plot repeatedly updates.
output$SO_CE_plane <- renderPlot({     < CE PLANE FUNCTION >  }) # renderplot end

}) # Observe event end

} # Server end
```

More Sophisticated App



https://darkpeakanalytics.shinyapps.io/sadm-mk2/

Open-source materials

Simple materials:

App: https://robertasmith.shinyapps.io/sick_sicker/
Paper: https://wellcomeopenresearch.org/articles/5-69
Code: https://github.com/RobertASmith/paper_makeHEshiny
Tutorial: https://r-hta.org/tutorial/markov_models_shiny/

More advanced materials:

App: https://darkpeakanalytics.shinyapps.io/sadm-mk2/
Code: https://github.com/bitowaqr/sadm-mk2
Package: https://github.com/RobertASmith/darkpeak

R-HTA in LMIC’s

BresMed™