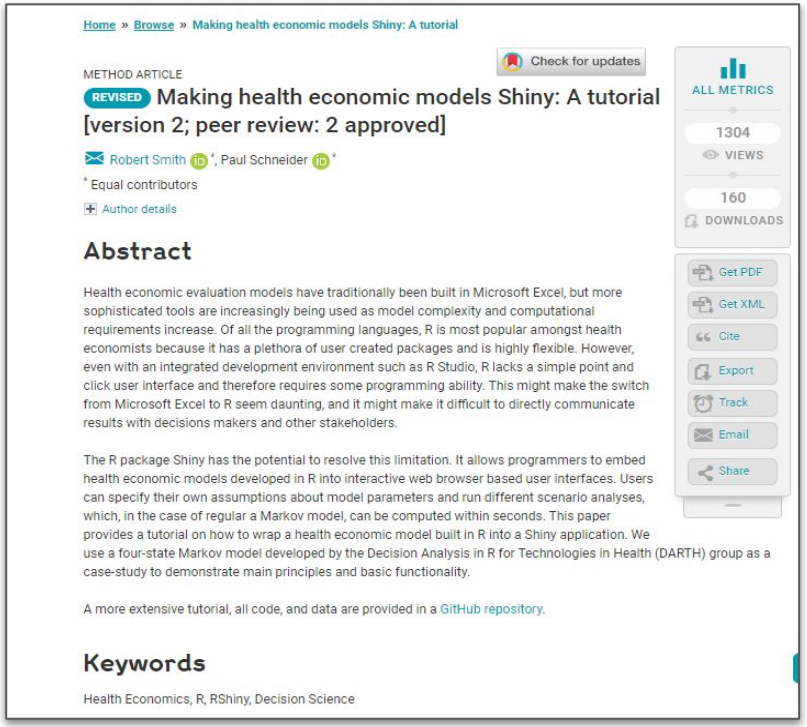
R Shiny for HTA Tutorial

Paul Schneider[1,2], **Robert Smith**[1,2,3], Rose Hart[4], Gianluca Baio[5]

[1]ScHARR, University of Sheffield; [2]Dark Peak Analytics; [3]UK Health Security Agency, DHSC; [4]Bresmed; [5]University College London

ISPOR Workshop – Wednesday, 1 December 2021

R Shiny for HTA 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.

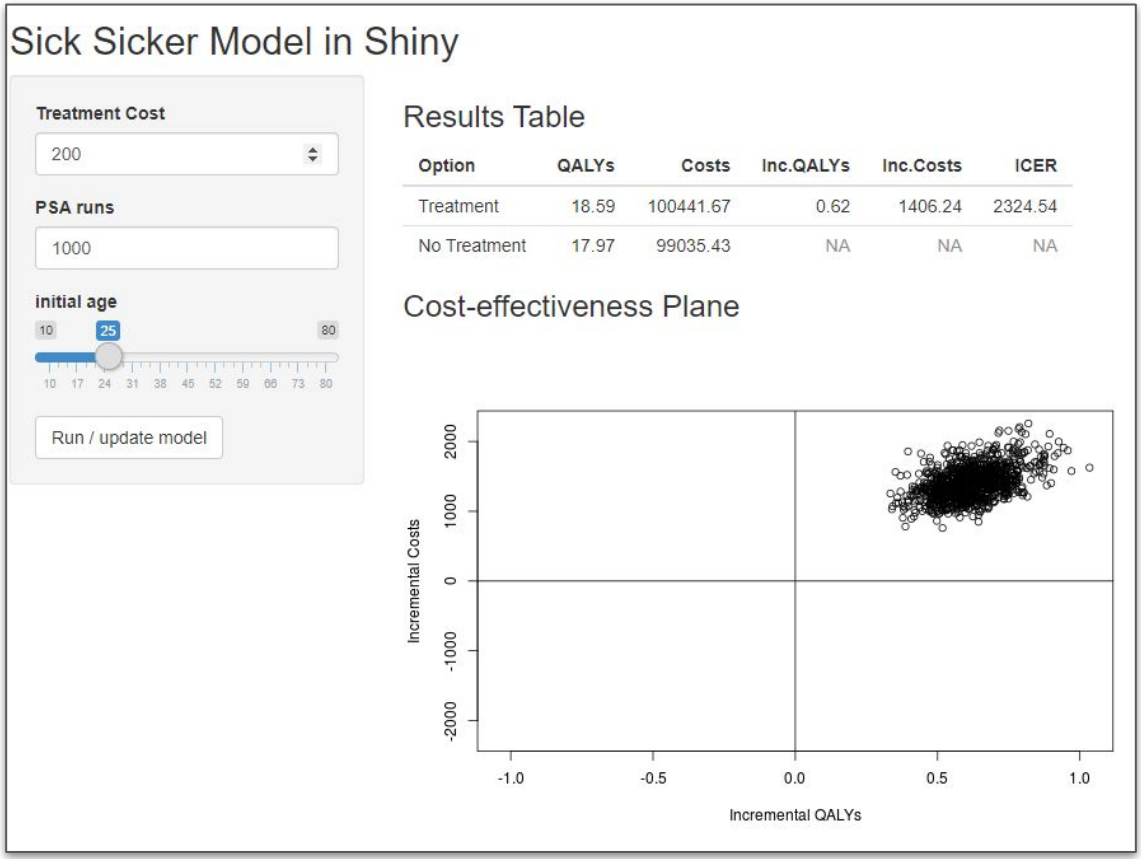
Keywords

Health Economics, R, RShiny, Decision Science

Paper:
https://wellcomeopenresearch.org/articles/5-69

Source code:
https://github.com/RobertASmith/healthecon_shiny

Simple app



https://robertasmith.shinyapps.io/sick_sicker

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

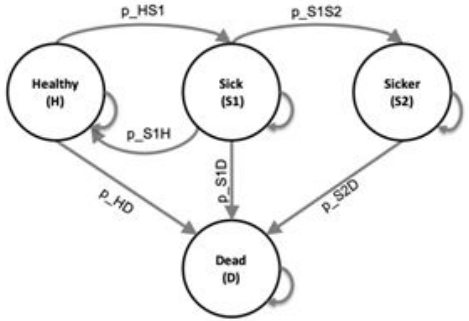
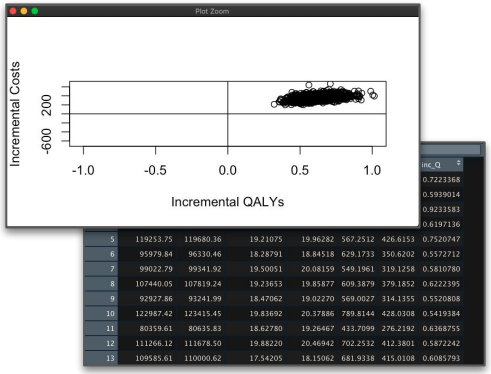


Figure 1: State-transition diagram of the time-independent Sick-Sicker cohort state-transition model with the name of the health states and possible transitions with their corresponding transition probabilities.



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

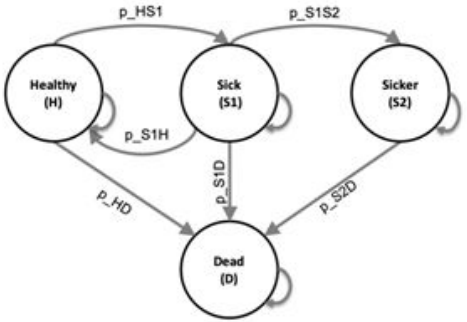
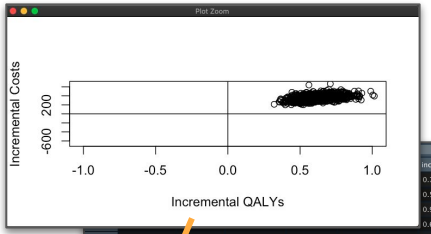
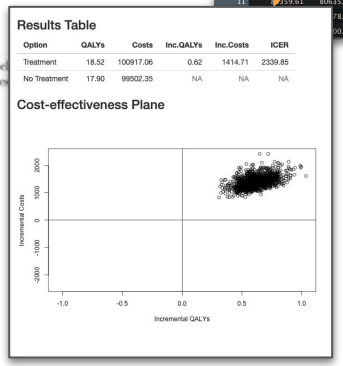


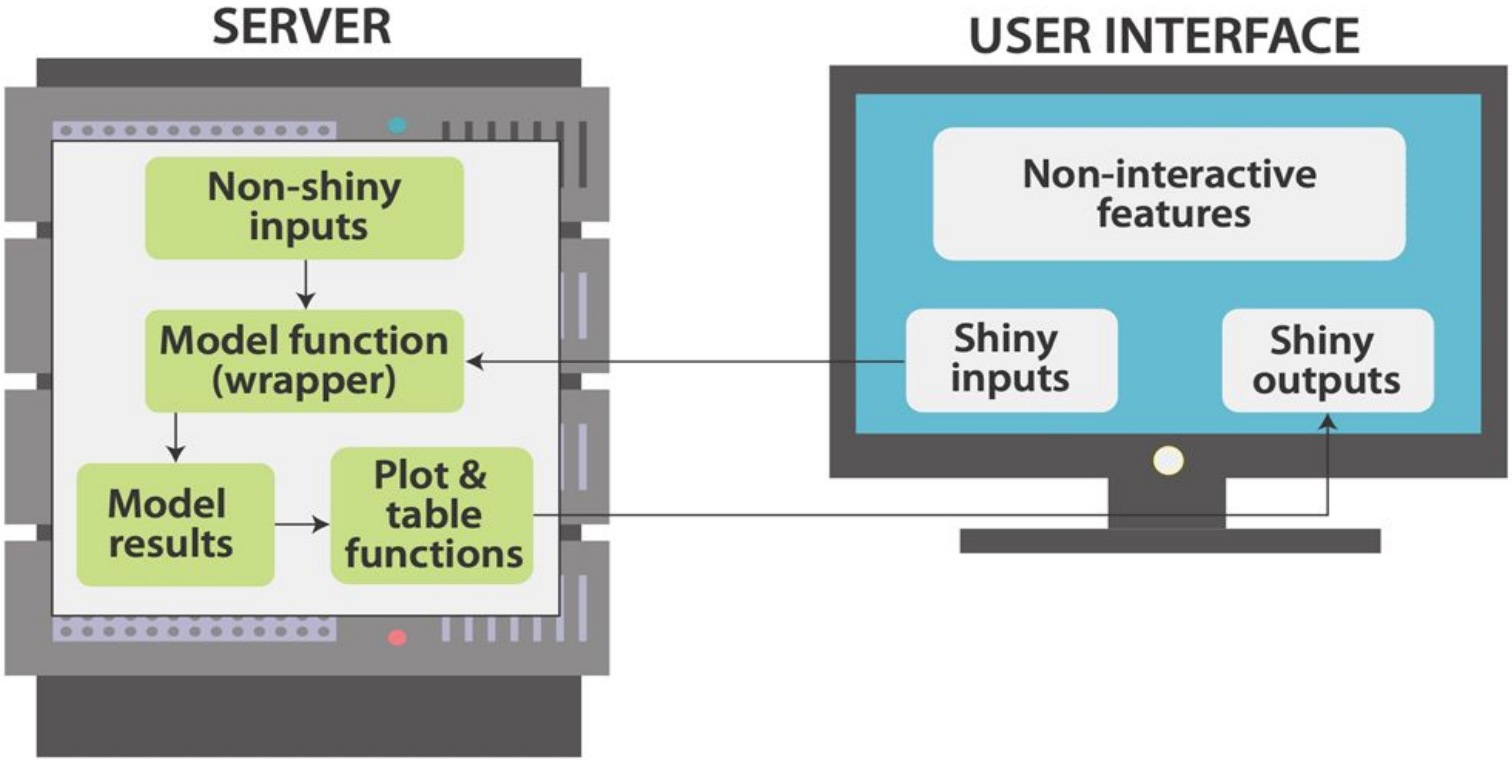
Figure 1: State-transition diagram of the time-independent Sick-Sicker cohort state-transition model
name of the health states and possible transitions with their corresponding transition probabilities



5	119253.75	119680.36	19.21075	19.96282	567.2512	426.6153	0.7520747
6	95979.84	96330.46	18.28791	18.84518	629.1733	350.6202	0.5572712
7	99022.[illegible]	99341.92	19.50051	20.08159	549.1961	319.1258	0.5810780
8	107440.[illegible]	107819.24	19.23653	19.85877	609.3879	379.1852	0.6222395
9	9[illegible].86	93241.99	18.47062	19.02270	569.0027	314.1355	0.5520808
10	12[illegible].42	123415.45	19.83692	20.37886	789.8144	428.0308	0.5419384
11	[illegible]159.61	80635.83	18.62780	19.26467	433.7099	276.2192	0.6368755
		[illegible]78.50	19.88220	20.46942	702.2532	412.3801	0.5872242
		[illegible]00.62	17.54205	18.15062	681.9338	415.0108	0.6085793



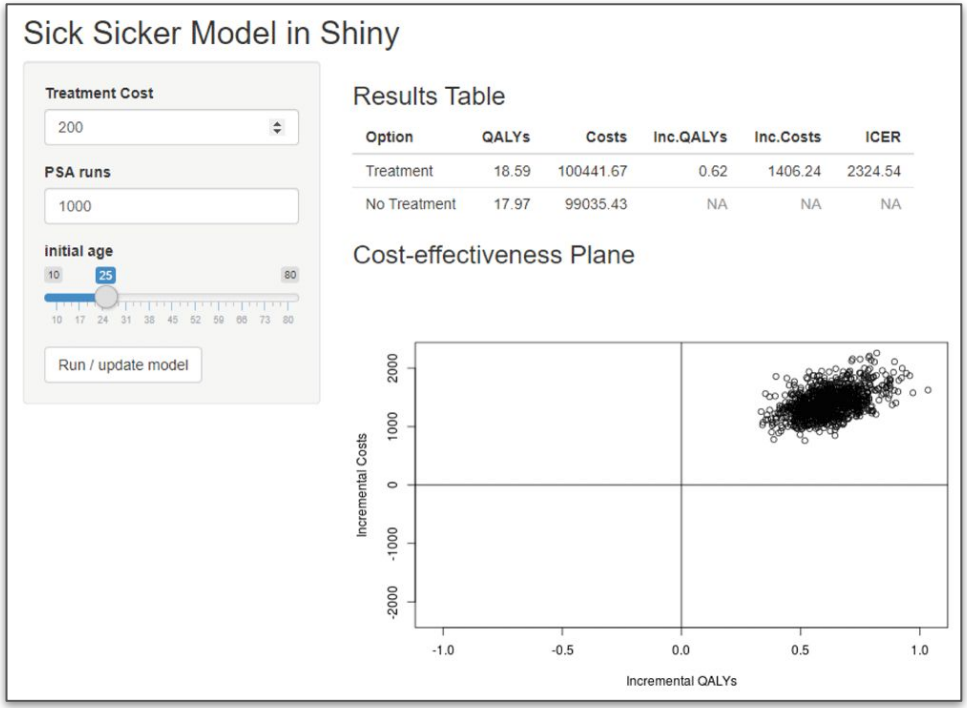
ShinyApp function



Shiny App

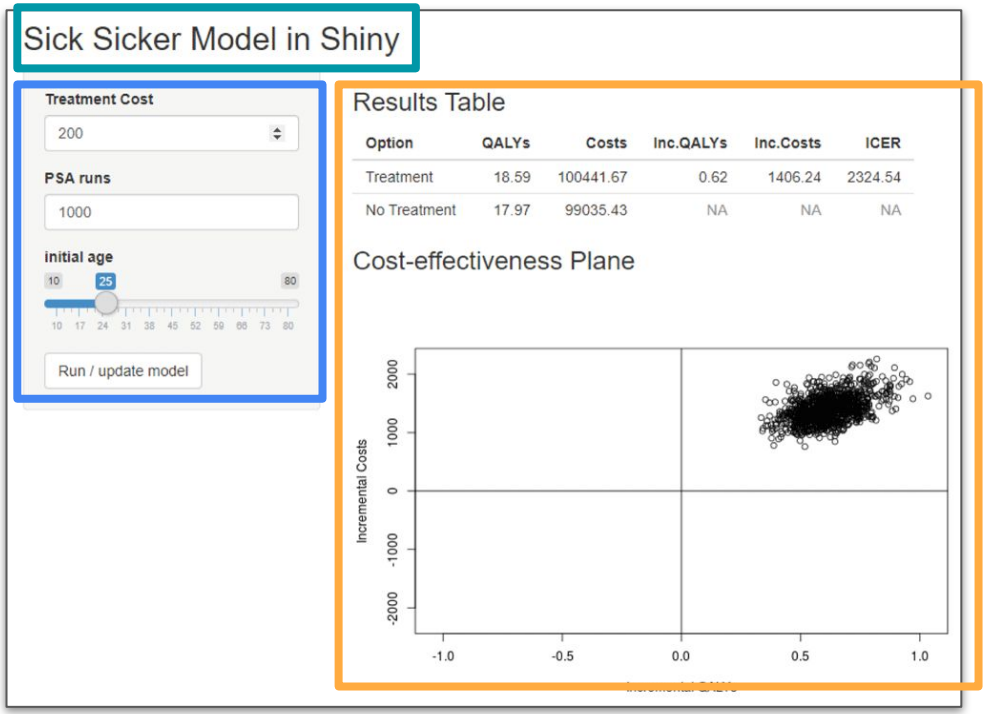
```
ui <- fluidPage (     # creates empty page
  # title of app
  titlePanel( "Sick Sicker Model in Shiny" ),
  # layout is a sidebar-layout
  sidebarLayout(
 # open sidebar panel
     < SIDEBAR PANEL CODE >
# open main panel
     < MAIN PANEL CODE >
       )# close sidebarlayout
) # close UI fluidpage


server <- function(input, output) {
 observeEvent (input$run_model,  # WHEN BUTTON PRESSED
                ignoreNULL = FALSE,  {
        # Run  model function with Shiny inputs
         df_model_res =  f_wrapper(
                c_Trt = input$ SI_c_Trt,
                n_age_init = input$ SI_n_age_init,
                n_sim = input$ SI_n_sim
                )

        #-- CREATE COST EFFECTIVENESS TABLE -- #
         output$ SO_icer_table <- renderTable({
                ICER_TABLE_FUNCTION(df_model_res )
                }) # table plot end.

        #-- CREATE COST EFFECTIVENESS PLANE -- #
         output$ SO_CE_plane <- renderPlot({
                CE_PLANE_FUNCTION(df_model_res)
                }) # renderplot end
  })
}

# Running the App ----
shinyApp(ui, server)
```

UI code



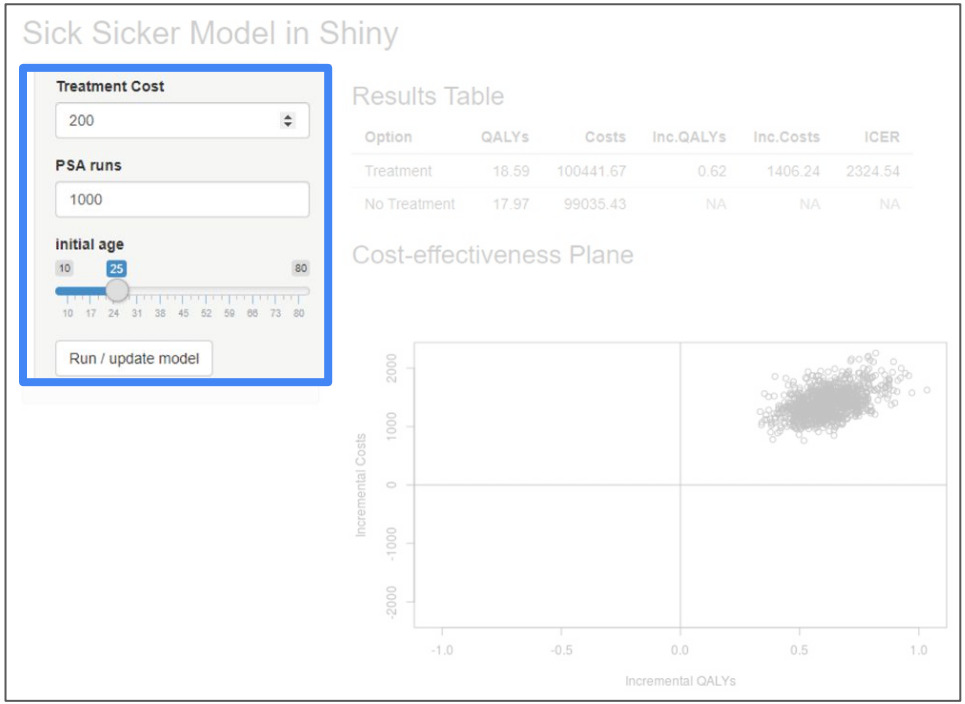

```
ui <- fluidPage (     # creates empty page

  # 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
```

UI code



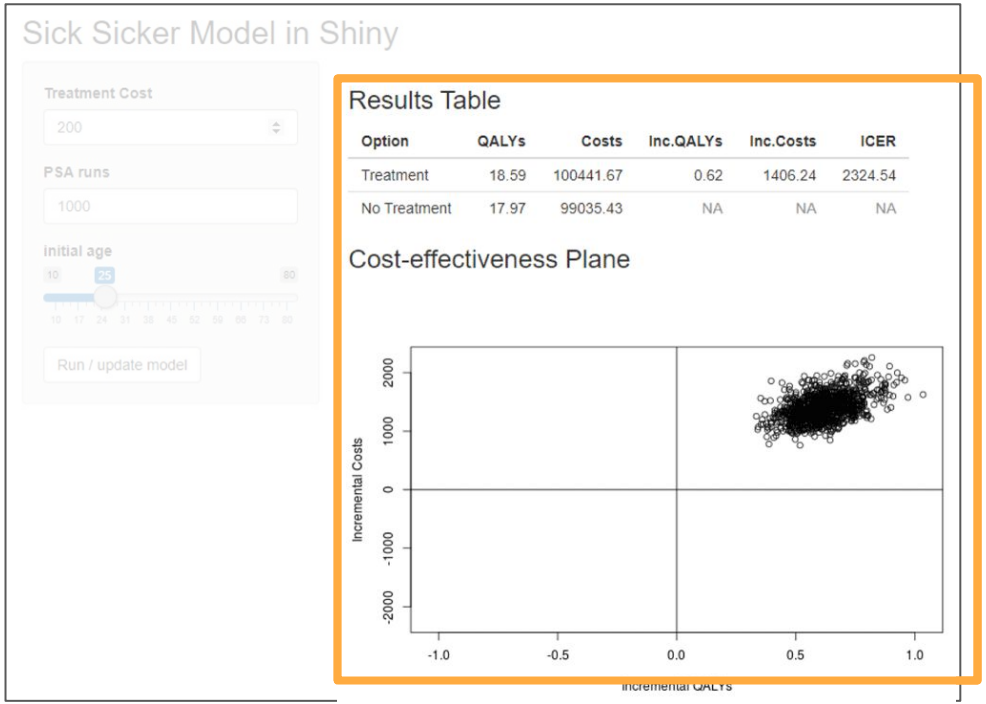

```
ui <- fluidPage (     # creates empty page

  # 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



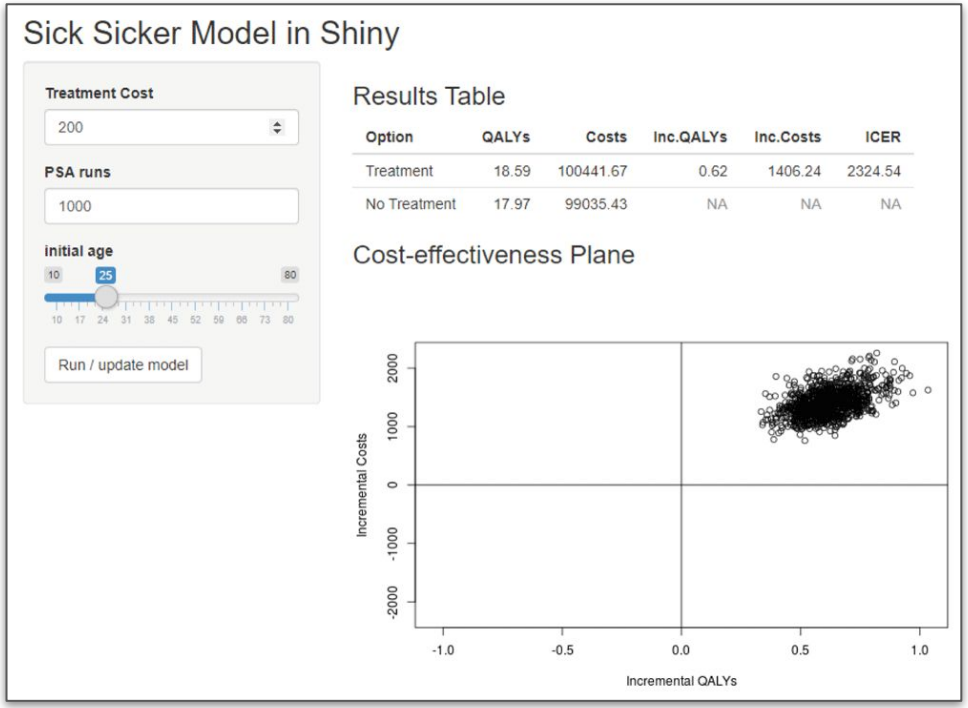

```
sidebarPanel( # open sidebar panel

    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/update 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
```

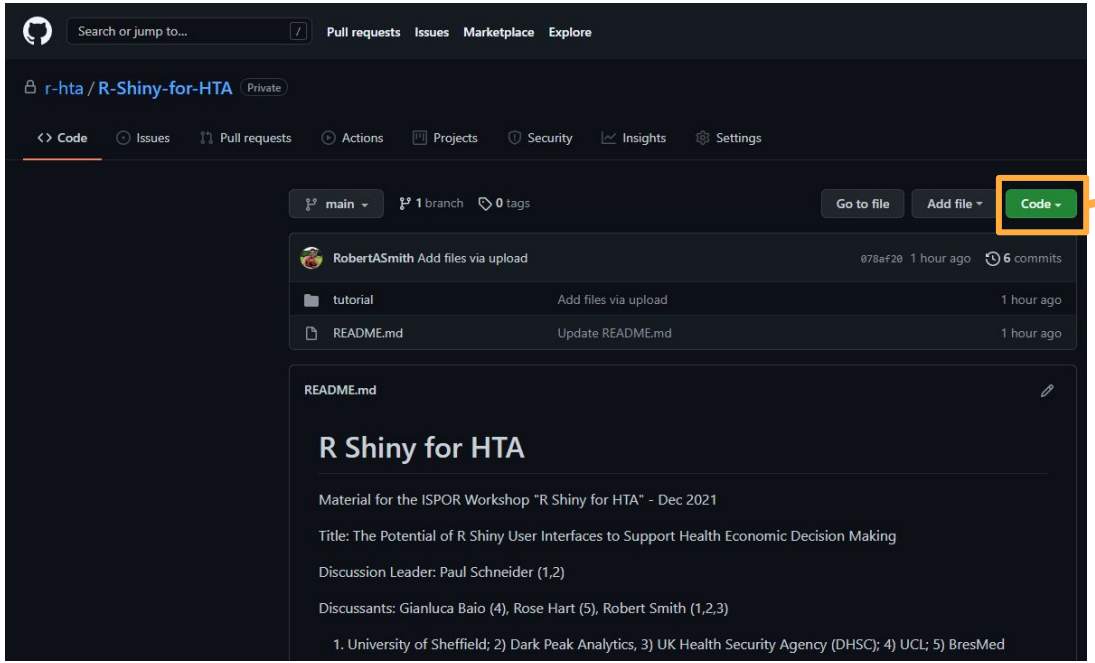
Server Code

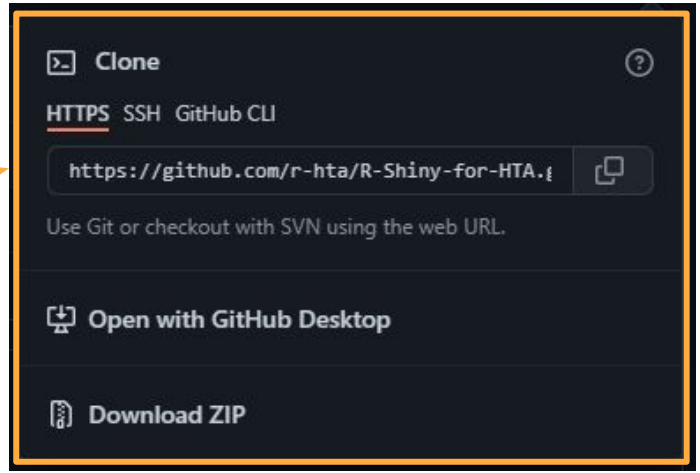



```
server <- function(input, output) {

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

       # 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 --#
        output$SO_icer_table <- renderTable({
              ICER_TABLE_FUNCTION(df_model_res)
              }) # table plot end.

       #-- CREATE COST EFFECTIVENESS PLANE --#
        output$SO_CE_plane <- renderPlot({
              CE_PLANE_FUNCTION(df_model_res)
              }) # renderplot end
  })
}
```

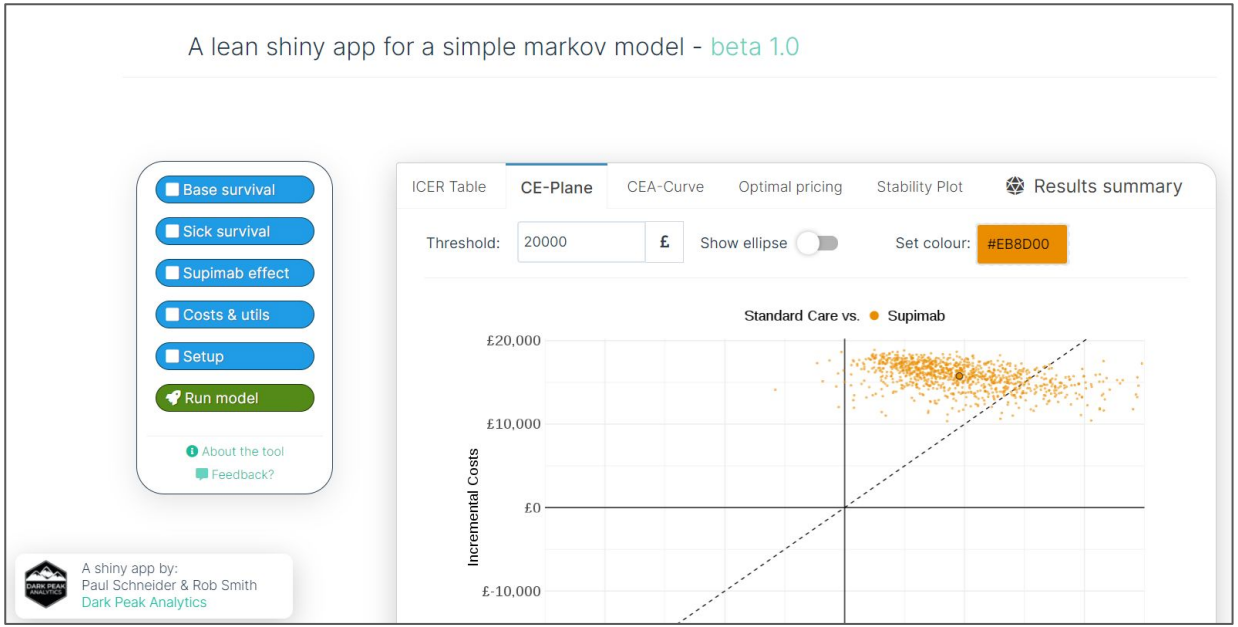
Workshop repository:

https://github.com/r-hta/R-Shiny-for-HTA





More sophisticated app



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

Open source materials

Workshop: https://github.com/r-hta/R-Shiny-for-HTA

Basic shiny app tutorial:
App: https://robertasmith.shinyapps.io/sick_sicker/
Paper: https://wellcomeopenresearch.org/articles/5-69
Code: https://github.com/RobertASmith/paper_makeHEshiny

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