HTA2

```
#install.packages("shiny")
library(shiny)
```

#ui

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

  # title of app
  "Markov Model_HTA working group",
  tabPanel("severity distribution",
       numericInput(inputId = "SI_d_severity_MA",
                    label = "Severity distribution_MA",
                    value = 0.25,
                    min = 0,
                    max = 1),

       numericInput(inputId = "SI_d_severity_MB",
                    label = "Severity distribution_MB",
                    value = 0.5,
                    min = 0,
                    max = 1),

       numericInput(inputId = "SI_d_severity_S",
                    label = "Severity distribution_S",
                    value = 0.25,
                    min = 0,
                    max = 1)    ),


  # layout is a sidebar-layout
  # open sidebar panel

      # input type numeric

   tabPanel("country" ,
      radioButtons(inputId ="radio", label = "Country",
                   choices = list("Base case" = 1, "Netherlands" = 2, "France"=3),selected = 1)),

   tabPanel("Price of medicine",
      numericInput(inputId = "SI_c_a",
                   label = "Price of study medicine A",
                   value = 2,
                   min = 0,
                   max = 20),
```

```
    numericInput(inputId = "SI_c_b",
                 label = "Price of study medicine B",
                 value = 0.5,
                 min = 0,
                 max = 10)),

   tabPanel("maintain care" ,
          fluidRow(
  column(2,
     numericInput(inputId = "SI_ru_MA_gp",
                 label = "number of GP visit for mild patients per year",
                 value = 3,
                 min = 0,
                 max = 10),

    numericInput(inputId = "SI_ru_MA_spiro",
                 label = "number of spirometry for mild patients per year",
                 value = 3,
                 min = 0,
                 max = 10),

    numericInput(inputId = "SI_ru_MA_beta",
                 label = "number of beta adrenergics for mild patients per year",
                 value = 100,
                 min = 50,
                 max = 150),

    numericInput(inputId = "SI_ru_MA_theoph",
                 label = "number of theophyline for mild patients per year ",
                 value = 365,
                 min = 100,
                 max = 800),

     numericInput(inputId = "SI_ru_MA_steroid",
                 label = "number of steroid for mild patients per year ",
                 value = 365,
                 min = 100,
                 max = 800)

  ),

  column(2,
    numericInput(inputId = "SI_uc_outgp",
                 label = "unit cost of GP visit",
                 value = 25,
                 min = 10,
                 max = 80),

    numericInput(inputId = "SI_uc_spiro",
                 label = "unit cost of spirometry",
                 value = 20,
                 min = 5,
                 max = 60),
```

```
      numericInput(inputId = "SI_uc_beta",
                   label = "unit cost of beta adrenergics",
                   value = 0.6,
                   min = 0.1,
                   max = 6),

      numericInput(inputId = "SI_uc_theoph",
                   label = "unit cost of theophyline ",
                   value = 0.5,
                   min = 0.1,
                   max = 5),

       numericInput(inputId = "SI_uc_steroid",
                   label = "unit cost of steroid",
                   value = 1,
                   min = 0.1,
                   max = 10),

      numericInput(inputId = "SI_uc_outspecialist",
                   label = "unit cost of specialist visit",
                   value = 60,
                   min = 10,
                   max = 120)
      ),

column(2,
       numericInput(inputId = "SI_ru_MB_gp",
                   label = "number of GP visit for moderate patients per year",
                   value = 3,
                   min = 0,
                   max = 10),

      numericInput(inputId = "SI_ru_MB_spiro",
                   label = "number of spirometry for moderate patients per year",
                   value = 3,
                   min = 0,
                   max = 10),

      numericInput(inputId = "SI_ru_MB_beta",
                   label = "number of beta adrenergics for moderate patients per year",
                   value = 150,
                   min = 200,
                   max = 100),

      numericInput(inputId = "SI_ru_MB_theoph",
                   label = "number of theophyline for moderate patients per year ",
                   value = 365,
                   min = 100,
                   max = 800),

       numericInput(inputId = "SI_ru_MB_steroid",
                   label = "number of steroid for moderate patients per year ",
                   value = 365,
```

```
                  min = 100,
                  max = 800)
    ),
column(2,



   numericInput(inputId = "SI_ru_S_gp",
                  label = "number of GP visit for servere patients per year",
                  value = 4,
                  min = 0,
                  max = 10),

     numericInput(inputId = "SI_ru_S_spiro",
                  label = "number of spirometry for servere patients per year",
                  value = 4,
                  min = 0,
                  max = 10),

     numericInput(inputId = "SI_ru_S_beta",
                  label = "number of beta adrenergics for servere patients per year",
                  value = 150,
                  min = 200,
                  max = 100),

     numericInput(inputId = "SI_ru_S_theoph",
                  label = "number of theophyline for servere patients per year ",
                  value = 365,
                  min = 100,
                  max = 800),

      numericInput(inputId = "SI_ru_S_steroid",
                  label = "number of steroid for servere patients per year ",
                  value = 365,
                  min = 100,
                  max = 800)
   ),

column(2,
   numericInput(inputId = "SI_ru_E_h_icu",
                  label = "number of ICU stay for exacerbation ",
                  value = 0,
                  min = 0,
                  max = 5),

      numericInput(inputId = "SI_ru_E_h_noicu",
                  label = "number of non ICU stay for exacerbation ",
                  value = 6,
                  min = 0,
                  max = 10),

      numericInput(inputId = "SI_ru_E_h_er",
                  label = "number of ER visit for exacerbation ",
```

```
                  value = 1,
                  min = 0,
                  max = 5)
  ),

column(2,
  numericInput(inputId = "SI_ru_SE_h_icu",
                  label = "number of ICU stay for severe exacerbation ",
                  value = 3,
                  min = 0,
                  max = 7),

     numericInput(inputId = "SI_ru_SE_h_noicu",
                  label = "number of non ICU stay for severe exacerbation ",
                  value = 8,
                  min = 2,
                  max = 20),

     numericInput(inputId = "SI_ru_SE_h_er",
                  label = "number of ER visit for severe exacerbation ",
                  value = 1,
                  min = 0,
                  max = 5)
  ))),

  tabPanel("Treatment for exacerbation" ,
      fluidRow(
  column(4,
     numericInput(inputId = "SI_uc_h_icu",
                  label = "unit cost of ICU stay ",
                  value = 1100,
                  min = 500,
                  max = 2500),

     numericInput(inputId = "SI_uc_h_noicu",
                  label = "unit cost of non ICU stay  ",
                  value = 250,
                  min = 120,
                  max = 600),

     numericInput(inputId = "SI_uc_h_er",
                  label = "unit cost of ER visit ",
                  value = 90,
                  min = 30,
                  max = 200),

     numericInput(inputId = "SI_ru_E_o_pulmon",
                  label = "number of specialist visit for exacerbation ",
                  value = 1,
                  min = 0,
                  max = 3),

     numericInput(inputId = "SI_ru_E_o_gp",
```

```
                label = "number of gp visit for exacerbation ",
                value = 1.5,
                min = 0,
                max = 5),

    numericInput(inputId = "SI_ru_E_o_anti",
                label = "number of antibiotic use for exacerbation ",
                value = 10,
                min = 0,
                max = 30),

    numericInput(inputId = "SI_ru_E_o_syssteroid",
                label = "number of systematic steroid use for exacerbation ",
                value = 15,
                min = 5,
                max = 50),

    numericInput(inputId = "SI_ru_E_o_inhsteroid",
                label = "number of inhale steroid use for exacerbation ",
                value = 40,
                min = 5,
                max = 100),

    numericInput(inputId = "SI_ru_E_o_theoph",
                label = "number of theophyline use for exacerbation ",
                value = 40,
                min = 5,
                max = 200)
    ),

  column(4,
    numericInput(inputId = "SI_ru_SE_o_pulmon",
                label = "number of specialist visit for servere exacerbation ",
                value = 1.7,
                min = 0,
                max = 5),

    numericInput(inputId = "SI_ru_SE_o_gp",
                label = "number of gp visit for servere exacerbation ",
                value = 1.6,
                min = 0,
                max = 6),

    numericInput(inputId = "SI_ru_SE_o_anti",
                label = "number of antibiotic use for servere exacerbation ",
                value = 16,
                min = 0,
                max = 40),

    numericInput(inputId = "SI_ru_SE_o_syssteroid",
                label = "number of systematic steroid use for servere exacerbation ",
                value = 30,
                min = 5,
```

```
                max = 90),

    numericInput(inputId = "SI_ru_SE_o_inhsteroid",
                label = "number of inhale steroid use for servere exacerbation ",
                value = 35,
                min = 5,
                max = 120),

    numericInput(inputId = "SI_ru_SE_o_theoph",
                label = "number of theophyline use for servere exacerbation ",
                value = 35,
                min = 5,
                max = 200),

    numericInput(inputId = "SI_ru_SE_o_oxy",
                label = "number of oxygen use for servere exacerbation ",
                value = 1,
                min = 0,
                max = 5),

    numericInput(inputId = "SI_uc_o_pulmon",
                label = "unit cost of specialist visit for (servere) exacerbation",
                value = 70,
                min = 20,
                max = 200)
    ),

  column(4,
    numericInput(inputId = "SI_uc_o_gp",
                label = "unit cost of gp visit for (servere) exacerbation ",
                value = 25,
                min = 10,
                max = 70),

    numericInput(inputId = "SI_uc_o_anti",
                label = "unit cost of antibiotics ",
                value = 5,
                min = 0.1,
                max = 20),

    numericInput(inputId = "SI_uc_o_syssteroid",
                label = "unit cost of systematic steroid ",
                value = 0.9,
                min = 0.1,
                max = 9),

    numericInput(inputId = "SI_uc_o_inhsteroid",
                label = "unit cost of inhalation steroid ",
                value = 0.5,
                min = 0.1,
                max = 5),

    numericInput(inputId = "SI_uc_o_theoph",
```

```
                    label = "unit cost of theophyline ",
                    value = 0.3,
                    min = 0.1,
                    max = 5),

       numericInput(inputId = "SI_uc_o_oxy",
                    label = "unit cost of oxygen ",
                    value = 235,
                    min = 50,
                    max = 600)
       ))),
 tabPanel(
   "select results to show" ,
      radioButtons(inputId ="radio1", label = "Result to show",
                    choices = list("patient time in disease status" = 1, "cost of medicine" = 2,"cost of




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


      # close sidebarPanel

    # open main panel
    mainPanel(
      tabsetPanel(
      # heading (results table)

      # tableOutput id = icer_table, from server
      tabPanel("Result Table",tableOutput(outputId = "result_table")),
      tabPanel("Transition Probability",fluidRow(tableOutput(outputId="prob_table_trtA_first"),tableOut
      tabPanel("Probability_Exacerbation", tableOutput(outputId="table_P_ex")),
      tabPanel("Probability_Severe_Exacerbation", tableOutput(outputId="table_P_ex_s"))


      )

    ) # close mainpanel


) # close UI fluidpage
```

```
## Warning: Navigation containers expect a collection of `bslib::nav()`/
## `shiny::tabPanel()`s and/or `bslib::nav_menu()`/`shiny::navbarMenu()`s. Consider
## using `header` or `footer` if you wish to place content above (or below) every
## panel's contents.
```

Shiny server function —-

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

  # when action button pressed ...
 datasetinput<- reactive({

                    d_MA <- input$SI_d_severity_MA
                    d_MB <- input$SI_d_severity_MB
                    d_S <- input$SI_d_severity_S
                    country <-input$radio

                    ppp<- {if (country== 2){ppp<-1.2}
                    else   if (country== 3){ppp<-0.8}
                    else {ppp<-1}}

                    show_results<- input$radio1

                    show_vars<-{if(show_results==1){show_vars<- c("Time_in_Mild","Time_in_Moderate","Time
                           else if(show_results==2){show_vars<- c("Cost_Drug")}
                           else if(show_results==3){show_vars<- c("Cost_maintain")}
                           else if(show_results==4){show_vars<- c("Cost_Hospitalization_exacerbation","C
                           else{show_vars<- c("Exacerbation","Severe_exacerbation","Total_Costs","Avoid_
                      }

                    c_a <- input$SI_c_a
                    c_b <- input$SI_c_b
                    ru_MA_gp <- input$SI_ru_MA_gp
                    ru_MA_spiro <- input$SI_ru_MA_spiro
                    ru_MA_infvac <- 1
                    ru_MA_beta <- input$SI_ru_MA_beta
                    ru_MA_theoph <- input$SI_ru_MA_theoph
                    ru_MA_steroid <- input$SI_ru_MA_steroid
                    pp_MA_gp <- 1
                    pp_MA_spiro <-1
                    pp_MA_infvac <-1
                    pp_MA_beta <- 0.5
                    pp_MA_theoph <-0.5
                    pp_MA_steroid <-0.2
                    uc_outgp <- input$SI_uc_outgp
                    uc_spiro <- input$SI_uc_spiro
                    uc_infvac <- 15
                    uc_beta <- input$SI_uc_beta
                    uc_theoph <- input$SI_uc_theoph
                    uc_steroid <- input$SI_uc_steroid

                    ru_MB_gp <- input$SI_ru_MB_gp
                    ru_MB_spiro <- input$SI_ru_MB_spiro
                    ru_MB_infvac <- 1
                    ru_MB_beta <- input$SI_ru_MB_beta
                    ru_MB_theoph <- input$SI_ru_MB_theoph
                    ru_MB_steroid <- input$SI_ru_MB_steroid
                    pp_MB_gp <- 1
```

```
pp_MB_spiro <-1
pp_MB_infvac <-1
pp_MB_beta <- 0.6
pp_MB_theoph <-0.6
pp_MB_steroid <-0.4
uc_outspecialist <- input$SI_uc_outspecialist


ru_S_gp <- input$SI_ru_S_gp
ru_S_spiro <- input$SI_ru_S_spiro
ru_S_infvac <- 1
ru_S_beta <- input$SI_ru_S_beta
ru_S_theoph <- input$SI_ru_S_theoph
ru_S_steroid <- input$SI_ru_S_steroid
pp_S_gp <- 1
pp_S_spiro <-1
pp_S_infvac <-1
pp_S_beta <- 0.7
pp_S_theoph <-0.7
pp_S_steroid <-0.7


ru_E_h_icu <- input$SI_ru_E_h_icu
ru_E_h_noicu <- input$SI_ru_E_h_noicu
ru_E_h_er <- input$SI_ru_E_h_er
pp_E_h_icu <- 0
pp_E_h_noicu <- 0.1
pp_E_h_er <- 0.05
uc_h_icu <- input$SI_uc_h_icu
uc_h_noicu <- input$SI_uc_h_noicu
uc_h_er <- input$SI_uc_h_er

ru_E_o_pulmon <- input$SI_ru_E_o_pulmon
ru_E_o_gp <- input$SI_ru_E_o_gp
ru_E_o_anti <- input$SI_ru_E_o_anti
ru_E_o_syssteroid <- input$SI_ru_E_o_syssteroid
ru_E_o_inhsteroid <- input$SI_ru_E_o_inhsteroid
ru_E_o_theoph <- input$SI_ru_E_o_theoph
pp_E_o_pulmon <- 0.35
pp_E_o_gp <- 0.4
pp_E_o_anti <- 0.7
pp_E_o_syssteroid <- 0.5
pp_E_o_inhsteroid <- 0.15
pp_E_o_theoph <- 0.1
uc_o_pulmon <- input$SI_uc_o_pulmon
uc_o_gp <- input$SI_uc_o_gp
uc_o_anti <- input$SI_uc_o_anti
uc_o_syssteroid <- input$SI_uc_o_syssteroid
uc_o_inhsteroid <- input$SI_uc_o_inhsteroid
uc_o_theoph <-input$SI_uc_o_theoph

ru_SE_h_icu <- input$SI_ru_SE_h_icu
ru_SE_h_noicu <- input$SI_ru_SE_h_noicu
```

```
                    ru_SE_h_er <- input$SI_ru_SE_h_er
                    pp_SE_h_icu <- 0.1
                    pp_SE_h_noicu <- 0.75
                    pp_SE_h_er <- 0.25
                    uc_h_icu <- input$SI_uc_h_icu
                    uc_h_noicu <- input$SI_uc_h_noicu
                    uc_h_er <- input$SI_uc_h_er

                    ru_SE_o_pulmon <- input$SI_ru_SE_o_pulmon
                    ru_SE_o_gp <- input$SI_ru_SE_o_gp
                    ru_SE_o_anti <- input$SI_ru_SE_o_anti
                    ru_SE_o_syssteroid <- input$SI_ru_SE_o_syssteroid
                    ru_SE_o_inhsteroid <- input$SI_ru_SE_o_inhsteroid
                    ru_SE_o_theoph <- input$SI_ru_SE_o_theoph
                    ru_SE_o_oxy <- input$SI_ru_SE_o_oxy
                    pp_SE_o_pulmon <- 0.5
                    pp_SE_o_gp <- 0.4
                    pp_SE_o_anti <- 0.75
                    pp_SE_o_syssteroid <- 0.8
                    pp_SE_o_inhsteroid <- 0.2
                    pp_SE_o_theoph <- 0.3
                    pp_SE_o_oxy <- 0.2

                    uc_o_oxy <- input$SI_uc_o_oxy


 # 1 month per cycle
n_cycles <- 12                          # number of cycles
v_names_cycles  <- paste("cycle", 0:n_cycles)    # cycle names
v_names_states  <- c("MA", "MB", "S")   # state names
n_states        <- length(v_names_states)        # number of health states
# Strategy names
v_names_str     <- c("Treatment A",
                     "Treatment B")
n_str           <- length(v_names_str)           # number of strategies

#transition probability in trt A
#from MA
p_a_MA_MB <- c(0.09, rep(0.04, (n_cycles-1)))
p_a_MA_S <- c(rep(0.01,(n_cycles)))
p_a_MA_MA <- c(0.9, rep(0.95,(n_cycles-1)))

#from MB
p_a_MB_MA <- c(0.25, rep(0.025,(n_cycles-1)))
p_a_MB_S <- c(0.05, rep(0.025,(n_cycles-1)))
p_a_MB_MB <- 1-(p_a_MB_MA+
             p_a_MB_S)

#from S
p_a_S_MA <- c( rep(0.01,(n_cycles)))
p_a_S_MB <-c(0.34, rep(0.04, (n_cycles-1)))
p_a_S_S <- 1-(p_a_S_MA+
           p_a_S_MB)
```

```
#transition probability in trt B
p_b_MA_MB <- c(0.24 ,rep(0.09, (n_cycles-1)))
p_b_MA_S <- c(rep(0.01, (n_cycles)))
p_b_MA_MA <- 1-(p_b_MA_MB+
             p_b_MA_S)
#from MB
p_b_MB_MA <- c(0.1,rep(0.01,(n_cycles-1)))
p_b_MB_S <- c(0.05, rep(0.04,(n_cycles-1)))
p_b_MB_MB <- 1-(p_b_MB_MA+
             p_b_MB_S)

#from S
p_b_S_MA <- c(rep(0.01,(n_cycles)))
p_b_S_MB <-c(0.19, rep(0.02,(n_cycles-1)))
p_b_S_S <- 1-(p_b_S_MA+
             p_b_S_MB)


#probability to ex
#treatment a prob. to ex in trt A

p_a_MA <- 0.05
p_a_MB <- 0.08
p_a_S <- 0.11

#treatment b prob. to ex

p_b_MA <- 0.08
p_b_MB <- 0.1
p_b_S <- 0.12

# prob. of ex to be servere in trt a
p_a_MA_s <- 0.1
p_a_MB_s <- 0.15
p_a_S_s <- 0.2

# prob. of ex to be servere in trt b
p_b_MA_s <- 0.15
p_b_MB_s <- 0.20
p_b_S_s <- 0.25

# maintain cost in trtA per cycle
c_E_h <- ((ru_E_h_icu*pp_E_h_icu*uc_h_icu)+(ru_E_h_noicu*pp_E_h_noicu*uc_h_noicu)+(ru_E_h_er*pp_E_h_er*
c_E_o <- ((ru_E_o_pulmon*pp_E_o_pulmon*uc_o_pulmon)+(ru_E_o_gp*pp_E_o_gp*uc_o_gp)+(ru_E_o_anti*pp_E_o_a

c_SE_h <-((ru_SE_h_icu*pp_SE_h_icu*uc_h_icu)+(ru_SE_h_noicu*pp_SE_h_noicu*uc_h_noicu)+(ru_SE_h_er*pp_SE

c_SE_o <-((ru_SE_o_pulmon*pp_SE_o_pulmon*uc_o_pulmon)+(ru_SE_o_gp*pp_SE_o_gp*uc_o_gp)+(ru_SE_o_anti*pp_

c_MA <-( ((ru_MA_gp*pp_MA_gp*uc_outgp)+(ru_MA_spiro*pp_MA_spiro*uc_spiro)+(ru_MA_infvac*pp_MA_infvac*uc

c_MB <- (((ru_MB_gp*pp_MB_gp*uc_outspecialist)+(ru_MB_spiro*pp_MB_spiro*uc_spiro)+(ru_MB_infvac*pp_MB_i
```

```
c_S <- (((ru_S_gp*pp_S_gp*uc_outspecialist)+(ru_S_spiro*pp_S_spiro*uc_spiro)+(ru_S_infvac*pp_S_infvac*u

#drug cost

c_drug_trtA <- c_a*30.5*n_cycles
c_drug_trtB <- c_b*30.5*n_cycles

#hospitalized cost for ex_trtA
hc_ex_MA_trtA <- ((p_a_MA*(1-p_a_MA_s))*(c_E_h))
hc_ex_MB_trtA <- ((p_a_MB*(1-p_a_MB_s))*(c_E_h))
hc_ex_S_trtA <- ((p_a_S*(1-p_a_S_s))*(c_E_h))

#other cost for ex_trtA
oc_ex_MA_trtA <- ((p_a_MA*(1-p_a_MA_s))*(c_E_o))
oc_ex_MB_trtA <- ((p_a_MB*(1-p_a_MB_s))*(c_E_o))
oc_ex_S_trtA <- ((p_a_S*(1-p_a_S_s))*(c_E_o))

#hospitalized cost for severe ex_trtA
hc_exs_MA_trtA <- ((p_a_MA*(p_a_MA_s))*(c_SE_h))
hc_exs_MB_trtA <- ((p_a_MB*(p_a_MB_s))*(c_SE_h))
hc_exs_S_trtA <- ((p_a_S*(p_a_S_s))*(c_SE_h))

#other cost for severe ex_trtA
oc_exs_MA_trtA <- ((p_a_MA*(p_a_MA_s))*(c_SE_o))
oc_exs_MB_trtA <- ((p_a_MB*(p_a_MB_s))*(c_SE_o))
oc_exs_S_trtA  <- ((p_a_S*(p_a_S_s))*(c_SE_o))

#hospitalized cost for ex_trtB
hc_ex_MA_trtB <- ((p_b_MA*(1-p_b_MA_s))*(c_E_h))
hc_ex_MB_trtB <- ((p_b_MB*(1-p_b_MB_s))*(c_E_h))
hc_ex_S_trtB <- ((p_b_S*(1-p_b_S_s))*(c_E_h))

#other cost for ex_trtB
oc_ex_MA_trtB <- ((p_b_MA*(1-p_b_MA_s))*(c_E_o))
oc_ex_MB_trtB <- ((p_b_MB*(1-p_b_MB_s))*(c_E_o))
oc_ex_S_trtB <- ((p_b_S*(1-p_b_S_s))*(c_E_o))

#hospitalized cost for severe ex_trtB
hc_exs_MA_trtB <- ((p_b_MA*(p_b_MA_s))*(c_SE_h))
hc_exs_MB_trtB <- ((p_b_MB*(p_b_MB_s))*(c_SE_h))
hc_exs_S_trtB <- ((p_b_S*(p_b_S_s))*(c_SE_h))

#other cost for severe ex_trtB
oc_exs_MA_trtB <- ((p_b_MA*(p_b_MA_s))*(c_SE_o))
oc_exs_MB_trtB <- ((p_b_MB*(p_b_MB_s))*(c_SE_o))
oc_exs_S_trtB <- ((p_b_S*(p_b_S_s))*(c_SE_o))


#cost in trtA per cycle
c_totalMA_trtA <- (c_a*30.5)+
                  ((p_a_MA*(1-p_a_MA_s))*(c_E_h+c_E_o))+
                  ((p_a_MA*(p_a_MA_s))*(c_SE_h+c_SE_o))+
                  (c_MA/12)
```

```r
c_totalMB_trtA <- (c_a*30.5)+
                  ((p_a_MB*(1-p_a_MB_s))*(c_E_h+c_E_o))+
                  ((p_a_MB*(p_a_MB_s))*(c_SE_h+c_SE_o))+
                  (c_MB/12)
c_totalS_trtA <- (c_a*30.5)+
                  ((p_a_S*(1-p_a_S_s))*(c_E_h+c_E_o))+
                  ((p_a_S*(p_a_S_s))*(c_SE_h+c_SE_o))+
                  (c_S/12)

#cost in trtB per cycle

c_totalMA_trtB <- (c_b*30.5)+
                  ((p_b_MA*(1-p_b_MA_s))*(c_E_h+c_E_o))+
                  ((p_b_MA*(p_b_MA_s))*(c_SE_h+c_SE_o))+
                  (c_MA/12)

c_totalMB_trtB <- (c_b*30.5)+
                  ((p_b_MB*(1-p_b_MB_s))*(c_E_h+c_E_o))+
                  ((p_b_MB*(p_b_MB_s))*(c_SE_h+c_SE_o))+
                  (c_MB/12)

c_totalS_trtB <- (c_b*30.5)+
                  ((p_b_S*(1-p_b_S_s))*(c_E_h+c_E_o))+
                  ((p_b_S*(p_b_S_s))*(c_SE_h+c_SE_o))+
                  (c_S/12)

## 04.1 Cohort trace
## Initial state vector
# All starting healthy
v_s_init <- c("MA" = d_MA, "MB" = d_MB, "S" = d_S)
v_s_init

## Initialize cohort trace for cSTM for all strategies
m_M_trtA <- matrix(0,
                   nrow = (n_cycles + 1), ncol = n_states,
                   dimnames = list(v_names_cycles, v_names_states))
# Store the initial state vector in the first row of the cohort trace
m_M_trtA[1, ] <- v_s_init
## Initialize cohort traces
m_M_trtB <- m_M_trtA  # structure and initial states remain the same

## 04.2 Transition probability array
## Initialize transition probability array
# all transitions to a non-death state are assumed to be conditional on survival
a_P_trtA <- array(0,  # Create 3-D array
                  dim = c(n_states, n_states, n_cycles),
                  dimnames = list(v_names_states, v_names_states,
                                  v_names_cycles[-length(v_names_cycles)])) # name the dimensions of the

a_P_trtB <- array(0,  # Create 3-D array
                  dim = c(n_states, n_states, n_cycles),
                  dimnames = list(v_names_states, v_names_states,
                                  v_names_cycles[-length(v_names_cycles)])) # name the dimensions of the
```

```r
## fill the matrix with probs
## trt A
# from MA
a_P_trtA["MA", "MA",] <- p_a_MA_MA
a_P_trtA["MA", "MB",] <- p_a_MA_MB
a_P_trtA["MA", "S",]  <- p_a_MA_S

# from MB
a_P_trtA["MB", "MA",] <- p_a_MB_MA
a_P_trtA["MB", "MB",] <- p_a_MB_MB
a_P_trtA["MB", "S",]  <- p_a_MB_S

# from S
a_P_trtA["S", "MA",] <- p_a_S_MA
a_P_trtA["S", "MB",] <- p_a_S_MB
a_P_trtA["S", "S",]  <- p_a_S_S

## trt B
# from MA
a_P_trtB["MA", "MA",] <- p_b_MA_MA
a_P_trtB["MA", "MB",] <- p_b_MA_MB
a_P_trtB["MA", "S",]  <- p_b_MA_S

# from MB
a_P_trtB["MB", "MA",] <- p_b_MB_MA
a_P_trtB["MB", "MB",] <- p_b_MB_MB
a_P_trtB["MB", "S",]  <- p_b_MB_S

# from S
a_P_trtB["S", "MA",] <- p_b_S_MA
a_P_trtB["S", "MB",] <- p_b_S_MB
a_P_trtB["S", "S",]  <- p_b_S_S


# 05 Run Markov model

for (t in 1:n_cycles){  # loop through the number of cycles
  # estimate the cohort trace for cycle t + 1 using the t-th matrix from the probability array

  m_M_trtA[t + 1, ] <- m_M_trtA[t, ] %*% a_P_trtA[, , t]
  m_M_trtB[t + 1, ] <- m_M_trtB[t, ] %*% a_P_trtB[, , t]
}


df_prob_trtA_first<-data.frame(a_P_trtA[,,1])
df_prob_trtA_after<-data.frame(a_P_trtA[,,2])
df_prob_trtB_first<-data.frame(a_P_trtB[,,1])
df_prob_trtB_after<-data.frame(a_P_trtB[,,2])

df_prob_ex<-data.frame(p_a_MA, p_a_MB, p_a_S)
df_prob_ex_s<- data.frame(p_a_MA_s, p_a_MB_s, p_a_S_s)
```

```r
# print the first few lines of the matrix

## 07.1  Costs and exacerbation nr

# per cycle
# calculate expected costs by multiplying cohort trace with the cost vector for the different health st

v_tc_trtA <- m_M_trtA %*% c(c_totalMA_trtA, c_totalMB_trtA, c_totalS_trtA)
v_tc_trtB <- m_M_trtB %*% c(c_totalMA_trtB, c_totalMB_trtB, c_totalS_trtB)

v_hc_ex_trtA <- m_M_trtA %*% c(hc_ex_MA_trtA, hc_ex_MB_trtA, hc_ex_S_trtA)
v_hc_ex_trtB <- m_M_trtB %*% c(hc_ex_MA_trtB, hc_ex_MB_trtB, hc_ex_S_trtB)

v_oc_ex_trtA <- m_M_trtA %*% c(oc_ex_MA_trtA, oc_ex_MB_trtA, oc_ex_S_trtA)
v_oc_ex_trtB <- m_M_trtB %*% c(oc_ex_MA_trtB, oc_ex_MB_trtB, oc_ex_S_trtB)

v_hc_exs_trtA <- m_M_trtA %*% c(hc_exs_MA_trtA, hc_exs_MB_trtA, hc_exs_S_trtA)
v_hc_exs_trtB <- m_M_trtB %*% c(hc_exs_MA_trtB, hc_exs_MB_trtB, hc_exs_S_trtB)

v_oc_exs_trtA <- m_M_trtA %*% c(oc_exs_MA_trtA, oc_exs_MB_trtA, oc_exs_S_trtA)
v_oc_exs_trtB <- m_M_trtB %*% c(oc_exs_MA_trtB, oc_exs_MB_trtB, oc_exs_S_trtB)

v_mc_trtA <- m_M_trtA %*% c(c_MA, c_MB, c_S)
v_mc_trtB <- m_M_trtB %*% c(c_MA, c_MB, c_S)

# per cycle
# calculate expected exacerbation number
v_exs_trtA <- m_M_trtA %*% c(p_a_MA*p_a_MA_s, p_a_MB*p_a_MB_s, p_a_S*p_a_S_s)
v_exs_trtB <- m_M_trtB %*% c(p_b_MA*p_b_MA_s, p_b_MB*p_b_MB_s, p_b_S*p_b_S_s)

v_ex_trtA <- m_M_trtA %*% c(p_a_MA, p_a_MB, p_a_S)
v_ex_trtB <- m_M_trtB %*% c(p_b_MA, p_b_MB, p_b_S)

## total time in disease status
v_a_MA<- m_M_trtA[,"MA"]
v_time_underdis_MA_a <- sum(v_a_MA)

v_a_MB<- m_M_trtA[,"MB"]
v_time_underdis_MB_a <- sum(v_a_MB)

v_a_S<- m_M_trtA[,"S"]
v_time_underdis_S_a <- sum(v_a_S)

v_b_MA<- m_M_trtB[,"MA"]
v_time_underdis_MA_b <- sum(v_b_MA)

v_b_MB<- m_M_trtB[,"MB"]
v_time_underdis_MB_b <- sum(v_b_MB)

v_b_S<- m_M_trtB[,"S"]
v_time_underdis_S_b <- sum(v_b_S)

df_months_in_disease_state_at_12month <-data.frame(
```

```
                    Option =  c("Treatment A","Treatment B"),

                    Time_in_Mild  =  c(v_time_underdis_MA_a,
                              v_time_underdis_MA_b),
                    Time_in_Moderate  =  c(v_time_underdis_MB_a,
                                  v_time_underdis_MB_b),
                    Time_in_Severe = c(v_time_underdis_S_a,
                              v_time_underdis_S_b)
                )

## 06.2.1 total exacerbation nr (u)

v_u_trtA <- sum(v_ex_trtA)
v_u_trtB <- sum(v_ex_trtB)

## 06.2.1 total severe exacerbation nr (u)
v_su_trtA <- sum(v_exs_trtA)
v_su_trtB <- sum(v_exs_trtB)

## 06.2.1 total cost (c)

v_c_trtA <- sum(v_tc_trtA)
v_c_trtB <- sum(v_tc_trtB)

#total hospitalization cost ex

v_thc_trtA<- sum(v_hc_ex_trtA)
v_thc_trtB<- sum(v_hc_ex_trtB)

#total other cost ex

v_toc_trtA<- sum(v_oc_ex_trtA)
v_toc_trtB<- sum(v_oc_ex_trtB)

#total hospitalization cost ex s

v_thcs_trtA<- sum(v_hc_exs_trtA)
v_thcs_trtB<- sum(v_hc_exs_trtB)

#total other cost ex s

v_tocs_trtA<- sum(v_oc_exs_trtA)
v_tocs_trtB<- sum(v_oc_exs_trtB)

#total maintain cost

v_tmc_trtA <- sum(v_mc_trtA)
v_tmc_trtB <- sum(v_mc_trtB)

c_ex_avoid <- (v_c_trtA-v_c_trtB)/(v_u_trtB-v_u_trtA)

Time_in_Mild  <-c(v_time_underdis_MA_a,v_time_underdis_MA_b)
Time_in_Moderate  <- c(v_time_underdis_MB_a,
```

```
                      v_time_underdis_MB_b)
Time_in_Severe <- c(v_time_underdis_S_a,
                    v_time_underdis_S_b)
Exacerbation  <-  c(v_u_trtA,v_u_trtB)
Severe_exacerbation <- c(v_su_trtA,v_su_trtB)
Cost_Drug <- c(c_drug_trtA,c_drug_trtB)
Cost_Hospitalization_exacerbation <- c(v_thc_trtA,v_thc_trtB)
Cost_Other_exacerbation <- c(v_toc_trtA,v_toc_trtB)
Cost_Hospitalization_severe_exacerbation <- c(v_thcs_trtA,v_thcs_trtB)
Cost_Other_severe_exacerbation <- c(v_tocs_trtA,v_tocs_trtB)
Cost_maintain <- c(v_tmc_trtA,v_tmc_trtB)
Total_Costs  <-  c(v_c_trtA,v_c_trtB)
Avoid_Exacerbation <- c(-(v_u_trtA -v_u_trtB),NA)
Inc.Costs = c(v_c_trtA - v_c_trtB,NA)
Addintional_costs_per_exacerbation_avoid = c(c_ex_avoid, NA)

dt_res_all <-data.table(Time_in_Mild,Time_in_Moderate ,Time_in_Severe,Exacerbation,Severe_exacerbation,
                        Cost_Drug,Cost_Hospitalization_exacerbation,Cost_Other_exacerbation,
                        Cost_Hospitalization_severe_exacerbation,Cost_Other_severe_exacerbation,
                        Cost_maintain,Total_Costs,Avoid_Exacerbation,
                        Inc.Costs,Addintional_costs_per_exacerbation_avoid)

df_res_table <- dt_res_all[ # create dataframe

                    # row.names  =  c("Treatment A","Treatment B"),
                    ,..show_vars
                     ]




 })




                # renderTable continuously updates table
output$result_table <- renderTable({
  if (input$run_model>0){isolate(datasetinput())}
})


n_cycles <- 12                          # number of cycles
v_names_cycles  <- paste("cycle", 0:n_cycles)    # cycle names
v_names_states  <- c("MA", "MB", "S")   # state names
n_states        <- length(v_names_states)        # number of health states
# Strategy names
v_names_str     <- c("Treatment A",
                     "Treatment B")
n_str           <- length(v_names_str)           # number of strategies

#transition probability in trt A
```

```
#from MA
p_a_MA_MB <- c(0.09, rep(0.04, (n_cycles-1)))
p_a_MA_S <- c(rep(0.01,(n_cycles)))
p_a_MA_MA <- c(0.9, rep(0.95,(n_cycles-1)))

#from MB
p_a_MB_MA <- c(0.25, rep(0.025,(n_cycles-1)))
p_a_MB_S <- c(0.05, rep(0.025,(n_cycles-1)))
p_a_MB_MB <- 1-(p_a_MB_MA+
             p_a_MB_S)

#from S
p_a_S_MA <- c( rep(0.01,(n_cycles)))
p_a_S_MB <-c(0.34, rep(0.04, (n_cycles-1)))
p_a_S_S <- 1-(p_a_S_MA+
           p_a_S_MB)

#transition probability in trt B
p_b_MA_MB <- c(0.24 ,rep(0.09, (n_cycles-1)))
p_b_MA_S <- c(rep(0.01, (n_cycles)))
p_b_MA_MA <- 1-(p_b_MA_MB+
             p_b_MA_S)
#from MB
p_b_MB_MA <- c(0.1,rep(0.01,(n_cycles-1)))
p_b_MB_S <- c(0.05, rep(0.04,(n_cycles-1)))
p_b_MB_MB <- 1-(p_b_MB_MA+
             p_b_MB_S)

#from S
p_b_S_MA <- c(rep(0.01,(n_cycles)))
p_b_S_MB <-c(0.19, rep(0.02,(n_cycles-1)))
p_b_S_S <- 1-(p_b_S_MA+
             p_b_S_MB)


#probability to ex
#treatment a prob. to ex in trt A

p_a_MA <- 0.05
p_a_MB <- 0.08
p_a_S <- 0.11

#treatment b prob. to ex

p_b_MA <- 0.08
p_b_MB <- 0.1
p_b_S <- 0.12

# prob. of ex to be servere in trt a
p_a_MA_s <- 0.1
p_a_MB_s <- 0.15
p_a_S_s <- 0.2
```

```r
# prob. of ex to be servere in trt b
p_b_MA_s <- 0.15
p_b_MB_s <- 0.20
p_b_S_s <- 0.25

## 04.2 Transition probability array
## Initialize transition probability array
# all transitions to a non-death state are assumed to be conditional on survival
a_P_trtA <- array(0,  # Create 3-D array
                dim = c(n_states, n_states, n_cycles),
                dimnames = list(v_names_states, v_names_states,
                                v_names_cycles[-length(v_names_cycles)])) # name the dimensions of the


## fill the matrix with probs
## trt A
# from MA
a_P_trtA["MA", "MA",] <- p_a_MA_MA
a_P_trtA["MA", "MB",] <- p_a_MA_MB
a_P_trtA["MA", "S",]  <- p_a_MA_S

# from MB
a_P_trtA["MB", "MA",] <- p_a_MB_MA
a_P_trtA["MB", "MB",] <- p_a_MB_MB
a_P_trtA["MB", "S",]  <- p_a_MB_S

# from S
a_P_trtA["S", "MA",] <- p_a_S_MA
a_P_trtA["S", "MB",] <- p_a_S_MB
a_P_trtA["S", "S",]  <- p_a_S_S


df_prob_trtA_first<-data.frame(a_P_trtA[,,1])
df_prob_trtA_after<-data.frame(a_P_trtA[,,2])

a_P_trtB <- array(0,  # Create 3-D array
                dim = c(n_states, n_states, n_cycles),
                dimnames = list(v_names_states, v_names_states,
                                v_names_cycles[-length(v_names_cycles)])) # name the dimensions of the
## trt B
# from MA
a_P_trtB["MA", "MA",] <- p_b_MA_MA
a_P_trtB["MA", "MB",] <- p_b_MA_MB
a_P_trtB["MA", "S",]  <- p_b_MA_S

# from MB
a_P_trtB["MB", "MA",] <- p_b_MB_MA
a_P_trtB["MB", "MB",] <- p_b_MB_MB
a_P_trtB["MB", "S",]  <- p_b_MB_S

# from S
a_P_trtB["S", "MA",] <- p_b_S_MA
a_P_trtB["S", "MB",] <- p_b_S_MB
```

```
a_P_trtB["S", "S",]  <- p_b_S_S


df_prob_trtB_first<-data.frame(a_P_trtB[,,1])
df_prob_trtB_after<-data.frame(a_P_trtB[,,2])

df_prob_ex<-data.frame(p_a_MA, p_a_MB, p_a_S)
df_prob_ex_s<- data.frame(p_a_MA_s, p_a_MB_s, p_a_S_s)


output$prob_table_trtA_first <- renderTable(df_prob_trtA_first)
output$prob_table_trtA_after <- renderTable(df_prob_trtA_after)
output$prob_table_trtB_first <- renderTable(df_prob_trtB_first)
output$prob_table_trtB_after <- renderTable(df_prob_trtB_after)
output$table_P_ex <- renderTable(df_prob_ex)
output$table_P_ex_s <- renderTable(df_prob_ex_s)




} # Server end
```

```
shinyApp(ui, server)
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed, pleas
```