

```

#This program was coded in R by Nam Lethanh based on original Fortran code developed earlier
by Tsuda Yoshitane during his time studying master at the laboratory of Prof. Kiyoshi
Kobayashi, Graduate School of Engineering, Kyoto University.
(http://psa2.kuciv.kyoto-u.ac.jp/)
#   for Educational purpose
#   Created date:   August, 2011 @Osaka University, Osaka, Japan
#   Modified date:  13rd, March, 2012 @IBI, ETHZ, Zurich, Switzerland
#   Version:        1.0
#   Users of the program are at their own risks, please kindly cite the owners of the
program and the cited paper.
#       ARIGATOU for using and citing my work.
#   Required reading material:
#   2)  Yoshitane TSUDA·Kiyoyuki KAITO·Kazuya AOKI·Kiyoshi KOBAYASHI, "ESTIMATING MARKOVIAN
TRANSITION PROBABILITIES FOR BRIDGE DETERIORATION FORECASTING", J.Struct. Mech. Earthquake
Eng., JSCE, No.801/I-73,pp.69-82, 2005.10. (Downloadable from
https://sites.google.com/site/namkyodai/publications)

#####INPUT PART#####
jmax=5 # Range of condition states
theta<-c(0.4,0.27,0.19,0.15,0) # Hazard rate of each condition state.
z=1 # Please select the interval or elapsed time in Markov processs
totalyear=30 # Total number of years used to compute the state probabilities over time.
#####
source("mtp.R") #The source code of function to estimate Markov transition probability
mtp=mtp(jmax,z,theta) #Markov transition probability.

RMD<-matrix(double(1),nrow=1,ncol=(jmax-1))# duration of each condition state
CRMD<-matrix(double(1),nrow=1,ncol=(jmax)) #cummulative life time of condition state
#subroutine to compute the life expectancy based on hazard rate.
for (i in 1:(jmax-1)){
RMD[i]=1/theta[i] #in years
}
#subroutine to compute the cummulative life expectancy for plotting
for (i in 1:(jmax)){
if (i==1){
CRMD[i]<-0
} else {
CRMD[i]=CRMD[i-1]+RMD[i-1]
}
}
#par(mfrow=c(1,2),mar=rep(1, 4))
plot.new() #plot new graph if needed
plot(CRMD[,],c(1:jmax),xlim=c(0,CRMD[jmax]),ylim=rev(range(c(1,jmax))),xlab="",ylab="",type=
"p")
predict(smooth.spline(CRMD[,],c(1:jmax),df=4.5)) #function to smooth the curve, df is the
smoothness level.
lines(predict(smooth.spline(CRMD[,],c(1:jmax),df=4.5),x=seq(0,CRMD[jmax],length=100)),col=
"red",lwd=2)
title(main="Deterioration curve", col.main="red", font.main=4)
title(xlab="Time (years)", col.lab=rgb(0,0.5,0))
title(ylab="Condition states", col.lab=rgb(0,0.5,0))

csstate<-matrix(double(1),nrow=totalyear,ncol=jmax) #state probability in year
for (t in 1:totalyear){
if (t==1) {
for (i in 1:jmax){
if (i==1) {

```

```

csstate[t,i]<-1
} else {
csstate[t,i]<-0
}
}
} else {
csstate[t,]<-csstate[(t-1),]*%mtp
}
}

#a function to plot the stacked shaded areas over time
plot.new()
stackedPlot <- function(data, time=NULL, col=1:length(data),...) {
  if (is.null(time)) {
    time <- 1:length(data[[1]])
  }
  plot(0,0, xlim = range(time), ylim = c(0,max(rowSums(data))), t="n" , ...)
  for (i in length(data):1) {
    #the summup to the current collumn
    prep.data <- rowSums(data[1:i]);
    # The polygon must have his first and last point on the zero line
    prep.y <- c(0, prep.data,0)
    prep.x <- c(time[1], time, time[length(time)] )
    polygon(prep.x, prep.y, col=col[i], border = NA);
  }
}
}
}

#-----
#define evolution of conditison states over time into data frame
csstate=csstate*100 #turning probability distribution of condition states to percentage
csstate <- data.frame(csstate)

#draw the graph
#stackedPlot(csstate,col = c("blanchedalmond","blue1","coral",507,525),xlab="Time
(years)",ylab="Percentage(%)" ) #this is used when user want to define his/her own favourite
for each condition state.
stackedPlot(csstate,col = colours(),xlab="Time (years)",ylab="Percentage(%)" ) #this is used
for defaul colour
title(main="Condition state distribution", col.main="red", font.main=4)
#legend("topright", inset=0.05, title="Condition
states",col=c("blanchedalmond","blue1","coral",507,525),lty=2,lwd=15,c("1","2","3","4","5"),bg
="azure2") #this is used to define favourite colour that corresponding to colour of shaded
area of each condition states.
legend("topright", inset=0.05, title="Condition states",col=colours(),lty=2,lwd=15,legend=c(1
:jmax),bg="azure2") #this is used to have default colours.

#THE END

```

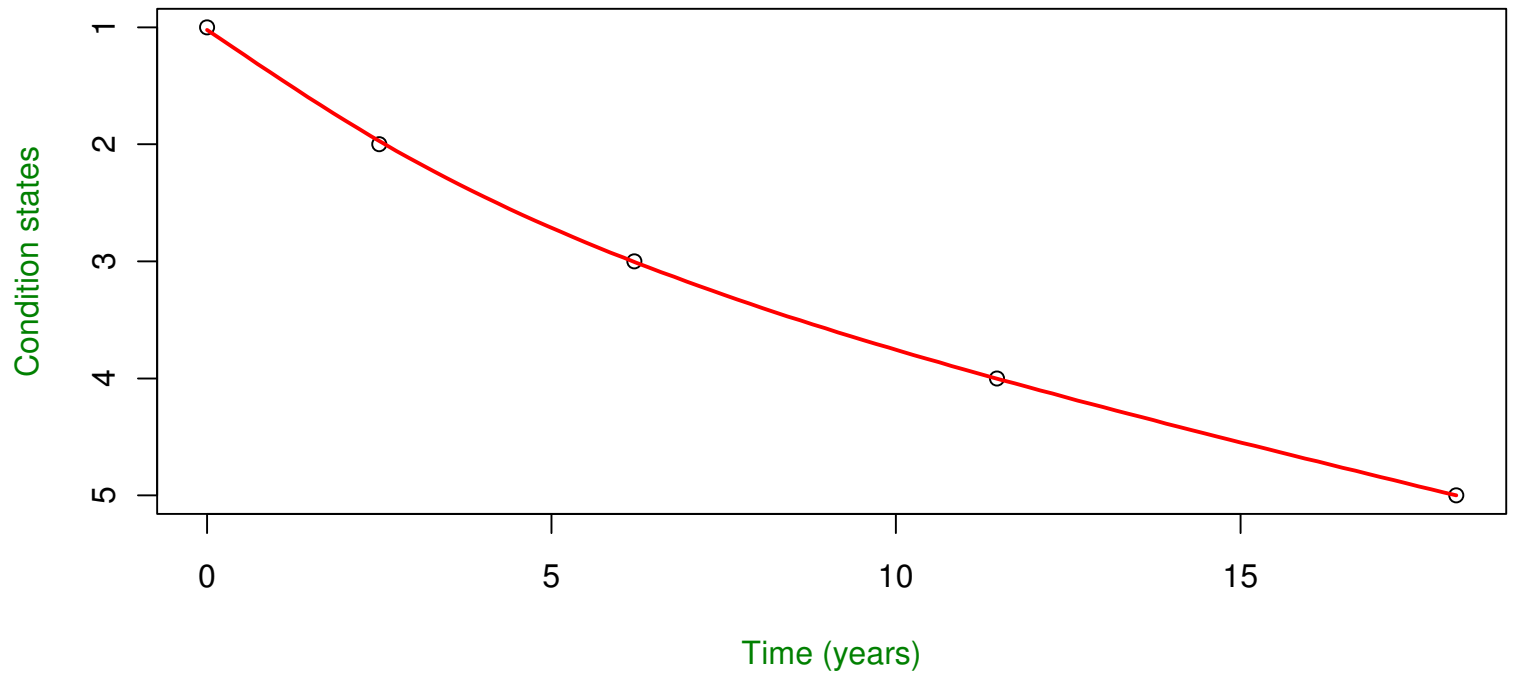
```

mtp<-function(jmax,z,theta){
thetasa<-matrix(double(1),nrow=jmax,ncol=jmax)
probb<-matrix(double(1),jmax,jmax)
thetasa<-matrix(double(1),nrow=jmax,ncol=jmax)
#####
for (i in 1:jmax){
for (j in 1:jmax){
thetasa[i,j]=theta[i]-theta[j]
}
}
#####

for (i in 1:jmax){
for (j in 1:jmax){
probl=0.0
reserve<-1.0
for (k in i:(j-1)){
if (j<=i) {
reserve=1
} else {
reserve=reserve*theta[k]
}
}
#####
if (i>j){
probb[i,j]=0.0
} else {
for (k in i:j){
prod11=1.0
#####
for (e in i:j){
if(e !=k) {
prod11=thetasa[e,k]*prod11
}
}
#####
probl=probl+exp(-theta[k]*z)/prod11
}
#####
probl<-probl*reserve
probb[i,j]=probl
}
}
}
return(probb)
}
#####

```

Deterioration curve



Condition state distribution

