**#MATRIX MULTIPLICATION FUNCTION IN R(remember to change diagnol matrix as the number u want, it 5 here)**

matmul= function(p,n) {

newmat=diag(**5**)

for (i in 1:n) {

newmat=newmat%\*%p

}

newmat

}

**#CALCULATE PREMIUM AFTER GIVEN N YEARS STARTING AT A SPECIFIC POINT START**

m=matrix(m,nrow = 4,ncol = 4)

m

discount=c(0,0.15,0.25,0.3)

position=c(1,0,0,0) #(CHANGE THIS FOR DIFFERENT POSITION)

premium=0

prem=function(p,n,start,discount){

for (i in 1:n) {

premium=premium+750\*sum(start\*(1-discount))

start=start%\*%p

}

premium

}

prem(m,5,position,discount)

**#lambda function for caluclation mean of waiting times (change index accordingly)**

lambda=function(x){

lambda\_motor=mean(x[,4][2:length(x[,4])])

lambda\_motor

}

**#expected values calculation in waitiing time(modify whole function according to question, time, length,etc)**

expected\_chisqaure\_waiting=function(test.data,lambda,largest.time){

n=sum(test.data[,1])

test.data=cbind(test.data,0)

test.data[1,2]=n\*(1-exp(-lambda\*0.5))

for (i in 2:largest.time) {

test.data[i,2]=n\*(exp(-lambda\*(i-1.5))-exp(-lambda\*(i-0.5)))

}

test.data=test.data[1:(largest.time),]

test.data[(largest.time),2]=n-sum(test.data[,2])

test.data[(largest.time),1]=n-sum(test.data[,1])

test.data=cbind(test.data,0)

for (j in 1:(largest.time)) {

test.data[j,3]=((test.data[j,1]-test.data[j,2])^2)/test.data[j,2]

}

test.data

}

**#triplets test(4 arguments are lines wise- triples, doubles,doubles(with last tern as it is),singles, possible triples(no of unique triplets possible)**

triplets.test=function(trips,doube,doub,sing,poss.trips) {

for (i in 1:length(trips[,1])) {

t=as.character(poss.trips[i,1])

xy=strtrim(t,2)

yz=substr(t,2,3)

y=substr(t,2,2)

n=as.numeric(doube[xy,1])

x=as.numeric(doub[yz,1])

y=as.numeric(sing[y,1])

trips[t,2]=n\*x/y

}

trips=cbind(trips,0)

for (i in 1:length(trips)) {

trips[,3]=((trips[,1]-trips[,2])^2)/trips[,2]

}

sum(trips[,3])

}

**#dof in triplets test**

dof=length(trips[,1])-length(doub[,1])+length(sing[,1])-1

**#time homogenous markov prob calculation**

a=matrix(c(-0.14,0.18,0.013,0.12,-0.23,0.05,0.02,0.05,-0.063),3,3)

i=diag(3)

ph=i+a\*(1/12)

ph

m=diag(3)

for (i in 1:120) {

m=m%\*%ph

}

m

**#time inhomogenous markov generatore matrix functikobn**

murb=function(x){

a=0.05+0.01\*x

a

}

murx=function(x){

a=0.001+0.0002\*x

a

}

mubr=function(x){

a=0.6-0.005\*x

a

}

mubx=function(x){

a=0.004+0.001\*x

a

}

muxr=function(x){

a=0.05-0.0003\*x

a

}

muxb=function(x){

a=0.9-0.006\*x

a

}

murr=function(x){

a=-(murb(x)+murx(x))

a

}

mubb=function(x){

a=-(mubr(x)+mubx(x))

a

}

muxx=function(x){

a=-(muxr(x)+muxb(x))

a

}

gen.matrix=function(x){

a=matrix(c(murr(x),mubr(x),muxr(x),murb(x),mubb(x),muxb(x),murx(x),mubx(x),muxx(x)),3,3)

a

}

s=gen.matrix(25)

**#for time imhomogenous prob calculation please check video answers, and answer accordingly please**

**#calculate curtate lifetime using fucntions**

expe=function(x){

e=0

npx=1

for (i in 1:(119-x)) {

px=(1-qx(x+i-1))

npx=npx\*px

e=e+npx

}

e

}

**#for proportional hazards model, pleaase see exam question, there was nothing new to enter here as notes**

**#expose to risk, calculating 70 age last biorthday etr**

funerel=read.csv("FuneralData.csv")

funerel

max of

dob,

doe,

start of investigation

min of

dod,

doexit,

date of ending of investiagtion

dob=funerel[,2]+70

dob

head(funerel)

doe=funerel[,3]

d0=2013.000

d=pmax(dob,doe,d0)

d=as.matrix(d)

d

head(funerel)

doex=funerel$BIRTH+71

d1=2017.999

dod=funerel$DEATH

dod[is.na(dod)]=d1

a=pmin(doex,d1,dod)

a=as.matrix(a)

head(a)

enddate=pmax(d,a) (this is done to remove neagtive elemnts, as let say someone is getting 70 on2020.000 , sinve we had taken max of all 3, its start date will become 2020.000, and since we have taken min of all 3, its end date will become 2017.999, taking diif of these 2 will give – values, therefore, we took max of start and end date to make these val;ue to be 0)

enddate

d

enddate

temp=enddate-d

head(temp)

sum(temp)

**#calculate mortality for 70 age last birthday**

birth=funerel$BIRTH[!is.na(funerel$DEATH) & funerel$DEATH>2013.000 & funerel$DEATH<2018.000]

birth

death=funerel$DEATH[!is.na(funerel$DEATH) & funerel$DEATH>2013.000 & funerel$DEATH<2018.000]

death

a=death-birth

d=sapply(a, floor)

d

sum(d==70)

mu=3/70.444

mu

**#determione no of policies in force on 2013,2014,2015……**

count70=function(x){

d1=x[2]

d2=x[3]

d3=x[4]

age=floor(census-d1)

present=(d2<=census) & (d3>census) | is.na(d3)

sum(ifelse(age==70 & present,1,0))

}

census=2013.000

count70(funerel)

census=2014.000;count70(funerel)

census=2015.000;count70(funerel)

**#graduation test (nothing new just read )(graduated using gompertz law)**

data=read.csv("Graduation2.csv")

data

head(data)

data[,4]=data[,3]/data[,2]

head(data)

gompertz=lm(log(data$CRUDE)~data$AGE)

gompertz

gompertz$coefficients

b=as.numeric(gompertz$coefficients[1])

b

c=as.numeric(gompertz$coefficients[2])

c

b=exp(b)

c=exp(c)

b

c

data$GRADUATED=b\*(c)^data$AGE

head(data)

plot(data$AGE,data$CRUDE)

lines(data$AGE,data$GRADUATED)

diff1=function(v)v[-1]-v[-length(v)]

a=diff1(diff1(diff1(data$GRADUATED)))

head(data$GRADUATED[-1])

head(data$GRADUATED)

head(data$GRADUATED[-3])

a=as.matrix(round(a,5))

a

head(data)

data[,6]=data[,2]\*data[,5]

head(data)

data[,7]=(data[,3]-data[,6])/sqrt(data[,6])

h

data[,8]=data[,7]^2

sum(data[,8])

nrow(data)

qchisq(0.95,49)

**#converting to ts with a specified start date and end date eg given start from jan 2005 and end at 2014 dec**

sales=ts(read.table("sales.txt"),start = c(2005,1),end=c(2014,12),frequency = 12)

**#lag.max is the number of total values**

**#frequency is 12,4,1 for monthly, quarterly, yearlu**

**#functions of burr and pareto**

rpareto <- function(n,a,l){

rp <- l\*((1-runif(n))^(-1/a)-1)

rp

}

dpareto <- function(x,a,l){

a\*(l^a)/((l+x)^(a+1))

}

ppareto <- function(q,a,l){

1-(l/(l+q))^a

}

qpareto <- function(p,a,l){

q <- l\*((1-p)^(-1/a)-1)

q

}

rburr <- function(n,a,l,g){

rp <- (l\*((1-runif(n))^(-1/a)-1))^(1/g)

rp

}

dburr <- function(x,a,l,g){

(a\*g\*l^a)\*x^(g-1)/((l+x^g)^(a+1))

}

pburr <- function(q,a,l,g){

1-(l/(l+q^g))^a

}

qburr <- function(p,a,l,g){

q <- (l\*((1-p)^(-1/a)-1))^(1/g)

q

}

**#for weibull remember all function take y(gamma) as (-1/y)**

**Means if we are given c=5, y=0.5**

**We need to give rweibull/pweibull/dweibull/qweibull(x,y,c^(-1/y))**

**#3 parameter pareto dpareto**

d3pareto <- function(x,a,l,k){

gamma(a+k)\*(l^a)\*x^(k-1)/(gamma(a)\*gamma(k)\*(l+x)^(a+k))

}

**#to find pparetto we need to take cummulative sum of integration of dpareto**

**We can use integrate function R but it requires the argument given must have 1 argument, since d3pareto have 3 arguments we need to convert the function into a function with 1 argument**

d3pareto.v2 <- function(x){

gamma(a+k)\*(l^a)\*x^(k-1)/(gamma(a)\*gamma(k)\*(l+x)^(a+k))

}

**#NowwecanintegratethePDFfrom0toq.Let’ssaywewanttofind(10)PX**

**,whereXfollowsathree‐parameterParetodistributionwith2**

**,4and5k.Firstwesetourparametervalues:**

a<-2; l<-4; k<-5

**#NowweintegratethePDFalongtherequiredrange:**

q<-10

integrate(d3pareto.v2,0,q)

0.451555 with absolute error < 8e-06

**#skewness function**

skew=function(x){

sum((x-(mean(x)))^3)/length(x)

}

**#nlm function, find minimum value that is close to 0**

**Example**

**We need to fing median of a 3 parameter pareto,**

**Now we don’t have qpareto for 3 parameter so we ned to find using integrate function**

**Where we eill take p3pareto$value-0.5=0**

**We need to minimise this relation or say equate to 0 to find valye oif median**

f=function(m){

abs(integrate(d3pareto.v2,0,m)$value-0.5)

}

o=10

nlm(f,o)

**#find mode using simple functions . we just find dlnorm values of a list of values,**

**Max max out of them(since mode is the value that maximises the probablity)**

**Find the value from the list which had that max value**

x=seq(0.000001,2,by=0.000001)

g=dlnorm(x,0,0.5)

g

max(g)

i=which(g==max(g))

i

x[i]

**#nlm function, maximizing parameter value by minimizing negative log sum valueas**

f<-function(param) {

neglnL<- -sum(dexp(claims,param,log=TRUE));

neglnL

}

p<-0.001

f(p)

MLE<-nlm(f,p)

MLE(this vwill give the extimate value for which negative of log likelihood is minimum, min of negative == max of positive)

**#arrrange data quarter wise(extreme value theory**

for (q in 1:4) {

startmonth<-(q-1)\*3+1

endmonth<-q\*3

maxima[maxima[,2] %in% new.order[startmonth:endmonth],4]<-q

}

head(maxima)

**#if we question wants to calculate gev, remember to copy paste it from the course notes khushboo lectures**