**Acturial Statistics**

**1.Calculation of present value and accumulated value under compound interest.**

Q.1: In compound interest system, prepare the table for calculating present value and accumulatedvalue of Re. 1 for different effective rate of interest and different period (take is 0.01, 0.02 0.15 and 1,2100 Based on above prepared table, answer the following questions

1. if a person deposit Rs. 1500/- at effective rate of interest 7%neaccumulated value at the end of the 17 years.
2. if a person will get Rs. 20000/- after 25 years at effective rate of interest 8% then calculate amount to be deposited today. (c) If a person deposited Rs. 25000/- for 20 years and at effective rate of interest during first 5 years 5% for next 10 years is 4% and remaining period it was 6%, then calculate its value after 20 years.

**Ans**

Accumulated Value:

year=seq(1,100,1)

n=length(year);

int=seq(0.01,0.15,0.01);

n\_i=length(int);

x=matrix(0,nrow=n,ncol=n\_i)

for (i in 1:n){

for(j in 1:n\_i) {

x[i, j]=(1+int[j])^(year[i])

}

} options(max.print=9999)

dimnames(x)=list(row.names=year,col.names=int)

x=round(x,2);x

#Present Value:

year=seq(1,100,1)

n=length(year);

int=seq(0.01,0.15,0.01);

x=matrix(0,nrow=n, ncol=n\_i)

for (i in 1:n){

for(j in 1:n\_i) {

x[I,j]=(1+int[])^(-year[i])

}

}options(max.print=9999)

dimnames(x)=list(row.names-year,col.names=int)

x=round(x,6);x

**Que 2**

In compound interest system, for each m prepare separate tables for calculating present value and accumulated value of Re. 1 for different nominal rate of interest and different period (take 1 (m) 0.01, 0.02 0.15 and n 1,2,100 and m 2, 4,12). Based on above table prepared, give answers the

following questions (a) If a person deposit Rs. 1000/- at nominal rate of interest 8% for 15 years then find its

accumulated value

(b) If a person will get Rs. 10000/- after 10 years where rate of interest 5% convertible semiannually then calculate its present value.

(c) if a person deposit Rs. 10000/- for 5 years with rate of interest 4% convertible quarterly for first 2 years and for the remaining period rate of interest is 6% convertible monthly, then calculate its accumulated value at end of the 5 year

**Ans**

year=seq(1,100,1)

n=length(year);

int-seq(0.01,0.15,0.01);

n\_i=length(int);

x=matrix(0,nrow=n,ncol=n\_i)

for (i in 1:n){

for(j in 1:n\_i) {

#x(i,j)=(1+(int[j] /2))^(year[i]\*2)

#x[i, j]=(1+(int[j] /4))^(year[i]\*4)

x[ij]=(1+(int[j]/12))^(year[i]\*12)

}

}

options(max.print-9999)

dimnames(x)=list(row.names=year,col.names=int)

x=round(x,6);x

**PLOTING OF UTILITY FUNCTION**

alpha=0.03 w=100

U1=-exp(-alpha\*(w-1))

for (w in 1:100)

{

( U1/w) exp(-alpha \*(w-1))}

graphics.off();par("mar"); par(mar=c[1,1,1,1)); par(mfrow=c(4,1));

plot(U1,main= "Exponential Utility function")

#1)=====Fractional Utility function

gamma=0.45;

U2=w^gamma;

for (w in 1:100) {

U2[w]=w^gamma;

}

plot(U2,main= "Fractional Utility function");

#iii)=====Quadratic Utility function===========

a=0.525; b=1/(2\*a);

W= seq(0.01,b,0.01)

U3-rep(0,length(W))

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

U3[i]=W[i]-a\*W[i]\*W[i];

}

plot(U3,main= "Quadratic Utility function");

#iv)=====Logarithmic Utility function

W1=seq(0.0.1:95);

U4=rep(0,length(W1))

k=10; c=5,

for (i in 1:length(W1)){

U4[i]=k\* log(W1[i])+c;

}

plot(U4,main "Logarithmic Utility function")

**Distribution of Total claim amount.**

**Q1**

X1=10000, with probability p=1/4

0,with probability p=3/4

Program: n=100;p=0.25

for (i in 1:length(n]]

x=matrix(c(rbinom(n[i]\*n[i],1,0.25)),n[0].nl));

y-apply(x,1,sum

par(mfrow=c(2,1))

qqnorm(y) hist(y)

**Construction of Life tables**

Q2

X=c(0,1,2,3,4,5,6,7,8,9);

q=c(0.1,0.19,0.16,0.14,0.08,0.26,0.5,0.6,0.8,1);

p-1-q;

w=length(p);

l1=120;

I=c(11,2.w);

for(i in 2:w){

l[i]=l[i-1]\*p[i-1];

}

I=round(1,2)

de= l\*q;

de=round(de,2)

L=c(1:(w-1),0.5\*l [w]);

for(i in 1:(w-1)){

l[i]=0.5\* (l[i]+l[i+1])

}

L=round(L,2)

T=c(1:(w-1),L[w])];

for(i in 1:(w-1)){

T[w-i]=T[w-i+1]+L[w-i]

}

T=round(T,2)

ex=T/l;

ex=round(ex,2);

y=data.frame(x,q,p,l,de,L,T,ex);y

**Estimate of Net Single (Actuarial) Premium based on random generated future life times**

**Que 1**: If future life time of an individual having age 20 years (T (201) has uniform distribution over (0, 80). Draw the random sample of size 1000 from uniform (0, 80), suppose these observations are future life times of 1000 persons having current age 20 years and i-0.08. Based on these observed future life data estimate the following:

Net single premium of whole life insurance, 10 year-term life insurance, 10 year-term pure endowment life insurance and 10 year-term endowment life insurance when benefit is payable at the moment of death. Also compare your estimated values with true values.

l=0.08;x=20;m-1000; n=10;

T=runif(m,0,80);

#Net single Premium for whole life insurance-

z1=(1+i)^(-T);

NSP1=mean(z1);

cat('Net Single Premium for whole life insurance is:")

print(NSP1)

#Net single Premium for 10-year term life insurance

z2-((1+i)^(-T))\* (T<=n);

NSP2=mean(z2);

cat('Net Single Premium for 10-year term life insurance is:")

print(NSP2)

#Net single Premium for 10-year term Pure endowment life insurance-

z3=((1+i)^(-T))\*(T>n);

NSP3=mean(z3);

cat('Net Single Premium for 10-year term pure endowment life insurance is:")

print(NSP3)

#-Net single Premium for 10-year term endowment life insurance-

z4=((1+1)^(-T))\*(T<=n)+((1+i)^(-n))(T>n);

NSP4=mean(24)

cat("Net Single Premium for 10-year term endowment life insurance is:")

print(NSP4)

**Calculation of present values of Annuities**

**Q1 Program**

library("mc2d")

library("lifecontingencies")

year=seq(1,100,1)

int=seq(0.01,0 15,0.01);

n\_i=length(int)

x=matrix(0,nrow=n,ncol=n\_i)

n=length(year)

for (i in 1:n)(

for in 1:n\_i) (

x[i,j]=annuity(int[j],year[i],m=0,k=1,type= "immediate")

}

) dimnames(x)=list(row.names=2000+year,col.names=int);

X

**Que 3**

Q3: Prepared separate tables for present value of m immediate annuity certain for different nominal

rate of interest and different period (Take i (m) 0.01, 0.02,.0.15, n=1,2,100, m-2 or 4 or 12 or

365) and give the answer to the following questions:

(a) If a person wants to get Rs.500/- at end of each quarter up to 12 years at nominal rate of interest 9%,

Year=seq(1,100,1)

n=length(year)

int1=(1+(int/4))^4-1;

int-seq/0.01,0.15,0.01);

n\_i=length(int1)

x=matrix(0,nrow-n,ncol=n\_i)

for (i in 1:n){

for (j in 1:n\_i){

( x[ij]-annuity(intl.year[i],m-0,k-4,type="immediate")

}

}

dimnames(x)=list(row.names=2000+year,col.names=int)

x

**Repayment of loan amount**

Q1: Prepare tables showing the loan outstanding, interest paid and repayment of principal for successive installments of uniform(equal installments) payment, under a loan amount of Rs. 15,00,000/ to be repaid by 15 level uniform annual payment at end of year and effective rate of interest be 0.10 and 0.12

library("fecontingencies")

i=-0.1;

L=1500000;

n=15;

x=annuity(I,n,m=0,k=1,type="immediate")

iny=rep(0,n)

table=data.frame(installment=L/x,Int\_paid=iny,prin\_Paid-iny,outstand-iny):

table[1,2]=L\*i

table(1,3)=table[1,1]- table [1,2] table [1,4]=L-table[1,3]

for (j in 2:n) (

table[j,2]=table[j-1,4]\*i

table [j,3]=table[j,1]-table[j,2] table [j,4]=table[j-1,4]-table[j,3]

}

table[n,4]=round(table[n,4])

View(table)

**Calculation of Ypearly premium values of whole life insurance**

Construct the table for calculation of yearly premium of whole life insurance by using the analytical laws of mortality where benefit will be paid at end of year of death and mode of the premium will be given at beginning of each year, take integer ages x running from 0 to 100. Also calculate continuous premium of whole life insurance and mthl ypremium of whole life insurance.

(a) Gompertz law

H. =BC,S(x) = exp(-m(C-1)), where 8>0,C>1,x20,m= log(C)

W =100,

B=0.00005;

C=1.096478,

A =0.0007,

del=0.05;

Int-exp(del)-1;m=B/log(C);

v=1/(1+int);d=1-v;

temp=rep(0,w+1);

dat=data.frame(Age=seq(0,100,1),Sx=temp,Px=temp,a\_x=temp, Ax=temp,P=temp, Abar=temp ,a\_xbar=temp, PAbar-temp)

dat$Sx=exp(-m \*(C^dat$Age-1));

dat$Px[1:w]=dat$Sx[2: (w+1)]/dat$Sx[1:w]

dat$a\_x[w]=1

for (i in w:2){

dat$a\_x[i-1]=1+dat$Px[i-1]\*v\*dat$a\_x[i]

}

dat$Ax=1-d \*dat$a\_x

dat$P= dat$Ax/dat$a\_x

dat$Abar=int \*dat$Ax/del

dat$Abar[w+1]=1

dat$a\_xbar=(1-dat$Abar)/del

dat$PAbar= dat$Abar/dat$a\_xbar

View(dat)

Dat

**TIME SERIES**

**Que 2**

A] gaussian white noise

#Simulation Of Gaussian Noise and White Noise- n-100; s1-1;s2-2;83-5;s4-8;

z1=rnorm(n,0,s1); z2=rnorm(h,0,s2);

z3 rnorm(n,0,s3); z4=rnorm(n,0,s4);

par(mfcol=c(4,1));

plot(z1,main="WN(0,1)", ylim=c(-20,20),type="l");

plot(z2,main="WN(0,4)",ylim-c(-20,20),type="I");

plot(z3,main="WN(0,25)",ylim-c(-20,20),type="I");

plot(z4,main="WN(0,64)",ylim-c(-20,20),type="I");

b]white noise

#---Simulation Binary Process and random walk---

n=500; p=0.5;

x1=rbinom(n,1,p);

x=-(x1==0)+x1;

par(mfcol=c(4,1));

par(mar=c(1,1,1,1));

plot(x,main= "Simulated Bonary process");

acf(x,n-1);

s=cumsum(x);

plot(s,main = "Simulation Of Random Walk")

acf(s,n-1);

**Que 3**

Obtain the realization of the following time series (X1,11,2-300)

a) were X, U cos(@r)+ V sin(01), @el-,aland U-Normal(0,1),V

-√3V,,WDere V,-Uniform(-1.1)

n=50;

U=rnorm(n,0,1);

V=sqrt(3)\*runif(n,-1,1);

f=1/6;

t=cbind(seq(1,n,1));

X=U\*cos(2\*pi\*f\*t)+V\*sin(2\*pi\*f\*t);

Y =cos(2\*pi\*f\*t)+sin(2\*pi\*f\*t);

# Plotting part

par(mfcol=c(2,1));

plot(X,type="I");

plot(Y,type="I");

**TOPIC 2: ESTIMATING THE TREND COMPONENET WITH LINEAR FILTERS**

**Topic:2 Estimating the trend component with linear filters,**

1. Consider the following time series data on 60 consecutive observations

53.5 53 53.2 52.5 53.4 56.5 65.3 70.7 66.9 58.2 55.3 53.4

52.1 51.5 51.5 52.4 53.3 52.2 55.5 64.2 69.6 58.5 55.3 53.6

52.3 51.5 51.7 51.5 57.1 63.6 68.8 68.9 60.1 55.6 53.9

53.3 53.1 53.5 53.5 53.9 57.1 64.7 69.4 70.3 62.657.6

54.8 54.2 54.6 54.3 54.8 58.1 68.1 73.3 75.5 66.4

(a) Observe the time series plot of the above data and estimate the trend component of

using 5-term moving average filter.

(b) Plot the time series data overlaid with the estimated trend component. Also obtain the

residual plot.

(c) Also estimate the trend component of it using 3-term moving average filter and 7- term moving average filter. And compare the three estimates of the trend component

library(matlab);

x=c(write all data as question not this

53.5 53 53.2 52.5 53.4 56.5 65.3 70.7 66.9 58.2 55.3 53.4

52.1 51.5 51.5 52.4 53.3 52.2 55.5 64.2 69.6 58.5 55.3 53.6

52.3 51.5 51.7 51.5 57.1 63.6 68.8 68.9 60.1 55.6 53.9

53.3 53.1 53.5 53.5 53.9 57.1 64.7 69.4 70.3 62.657.6

54.8 54.2 54.6 54.3 54.8 58.1 68.1 73.3 75.5 66.4

)

n=length(x);

q=1:

#Smoothing with moving average-

f=ones (2\*q+1,1)/(2\*q+1);

m=zeros(n,1);

m[1:q]=x[1:q];

for (i in (q+ 1):(n-q)){

m[i]=t(f)%\*%x[(i-q):(i+q)]

}

m[(n-q+1):n)=x[(n-q+1):n]

#Accuracy measure

MAD =mean (abs(x-m));

MSD=mean((t(x-m)}%\*%(x-m));

MAPE= mean (abs((x-m)/x)).

cat("MAD", MAD, "\n")

cat("MSD", MSD, "\n")

cat("MAPE", MAPE,"\n")

cat (The first 10 smotthed values are)

print(m[1:10]);

#-Plot-

plot(x,type="I", col="red")

par(new=TRUE)

plot(m,type="I", col="green")

Topic.1: Linear Programming Problems

1. Maximize z=3x, +2x2, subject to. 3x + x 59, 2x, +4x2 ≤20, 3x, +2x212,

Here we 'Ipsolve' package for Linear Programming

**Q.1)** Program: Rstudio

>f=c(3,2)

> A-matrix(c(3,1,2,4,3,2),ncol-2,byrow=TRUE)

> dir=c("<=", "<=", "<=") > B=c(9,20,12)

> sol=lp("max",f,A,dir,B,compute.sens=TRUE)

> sol$status; solSsolution; solSobjval

#status: 0 solution: [2 3] ObjVal: 12

1. Minimize z = x + x2, subject to: 2x1 + x2 ≥8, 3x1 +7x2 2 12, X1, X220.

Q.2) Program: Rstudio

>f=c(1,1)

> A=matrix(c(2,1,3,7),ncol-2,byrow-=TRUE)

> dir=c(">=","=")

> B-c(8,12)

>sol=lp("min",f,A,dir,B,compute.sens=TRUE)

> sol$status; solSsolution; solSobjval

#status: 0 solution: [4 0] ObjVal: 4

**3.**Maximize z=3x, +5x₂-2x, +10x4,

subject to -5x, +10x₂-3x, +6x, ≤5

-211+5x2-1, +7x58 11, 150, 12, 120 (**Different signs of xi variables**)

**4.**Minimize z=-3x1-12, subject to: x2+2x2 <5

7x, +3x2-5x, ≤20, 11, 12, 20 **(Degenerate Solutio**

Q3) Program: Rstudio

>f-c(3,5,-2,10)

>A matrix(c(-5,10,3,6-2,5,-1,7,1,0,0,0,0,0,1,0),ncol-4,byrow=TRUE)

> dir=c("<=", "<<=","<=","<=")

> B=c(5,8,0,0) >sol-ip("max",f,A,dir,B,compute.sens TRUE)

> solSstatus; solSsolution; solSobjval

#status: 0 solution: [0.000 0.000 0.000 0.833] Obj Val: 8.333

**Q.4)** Program: Rstudio

> E=c(-3,-1,0)

> A=matrix(c(1,2,0,1,1,-1,7,3,-5),ncol=3,byrow=TRUE)

> dir=c("<=", "<=", "<=")

> B=c(5,2,20) > sol-lp("min",f,A,dir,B,compute.sens=TRUE)

> sol$status; sol$solution; sol$objval

# sol$status: 0 solSsolution: [503] solSobjval -15

**6.** Minimize z=-2x1-x2,

subject to: x1 + x2 ≥5,

x1+x2≤4, x1,

x220. (**Infeasible Solution**)

**Q.6**) Program: Rstudio

> f=c(-2,-1) > A=matrix(c(1,1,1,1),ncol=2,byrow=TRUE)

> dir=c(">=","<=")

> B=c(5,4)

> sol-lp("min",f,A,dir,B,compute.sens=TRUE)

> sol$status

# status: 2

: **Topic.2: Integer Programming**

Problem Solve the following IPP.

1. Pure integer programming problem Minimize Z=-3x, -4x, subject to 3x, +2x2 58 X1 +4x2 $10 X1, X220 and are integers.

Here we '**Ipsolve'** package for Integer Programming

**Q.1)** Program :Rstudio

>[-c(-3,-4)

> A=matrix(c(3,2,1,4),ncol=2,byrow=TRUE)

> dir=c("<=","<") > B=c(8,10)

> sol-lp("min",f,A,dir,B,compute.sens-TRUE,all.int=TRUE)

> sol$status; solSsolution; sol$objval # sol$status: 0 sol$solution: [12] sol$objval -11

**2. Binary** Integer programming problem:

Maximize z = 18x, +14xy +8xg +41,

subject to 15x1 +12x2 +7x+4x+xg 5 37

X1, X2, X3, X4, X = 0,1

Q.2) Program: Rstudio

>f=c(18,14,8,4,0)

> A-matrix(c(15,12,7,4,1),ncol=5,byrow=TRUE)

> dir=c("<=")

> B=c(37)

> sol=lp("max",f,A,dir,B,compute.sens-TRUE,all.bin=TRUE)

> sol$status; solSsolution; solSobjval

#sol$status: 0 solSsolution: [1 1 1 0 0] solSobjval 40

1. **Mixed integer** programming problem:

Minimize z=x -3x2 subject to -x+2x ≤11/2 I, I, 20 and 12 is integers

Q3)Program: Rstudio

> fc(1,-3)

>A matrix(c(1,1,-1,2),ncol-2,byrow-TRUE)

> dir=c("<<"," > B-c(5,11/2)

>sol-lp("min",f,A,dir,B,compute.sens-TRUE,int.vec-c(2))

>solSstatus solssolution; solSobival

#solSstatus: 0 solSsolution: [0.5 3] solSobival -8.5

**Topic.3: Quadratic Programming Problem**

1. Solve the following QPP.

Z-10x-25x₂+10x+x+4xx,

Subject to: X1+2x2 $10 X1 +32 59,

Here we **'quadprog** package for Quadratic Programming

Q.1) Program: Rstudio

> Dmat-matrix(c(20,4,4,2),ncol-2,byrow=TRUE)

> dvec c(10,25)

> Amat-matrix(c(-1,-1,1,0,-2,-1,0,1),ncol=4,byrow=TRUE)

> bvec=c(-10,-9,0,0)

>m-solve.QP(Dmat,dvec,Amat,bvec)

> m$sol; mSvalue #m$sol:[0 5] mSvalue: -100

1. Solve the following QPP.

Maximize Z = (x+1)+(x-2)'

Subject to: X1 52 1,51, X1, X2 20

Q.2) Program: Rstudio

> Dmat-matrix(c(2,-1,-1,4),ncol-2,byrow=TRUE)

> dvec=c(1,1)

> Amat-matrix(c(-2,-1),ncol-1,byrow=TRUE)

>bvec=-1

> m-solve.QP(Dmat,dvec,Amat,bvec)

> mSsol; m$value

#m$sol:[0.3636364 0.2727273]

m$value: -0.4545455

**Single and double exponential Smoothing.**

**1.** Consider the following data on number of employees (in thousands) in metal industry measured each month over five years.

43.4 44.4 44.B 43.1 42.6 43.0 44.4 40.1 423 46.2 42.8 42.4 436 43.1 42.8 42.0 42.4 42.8 42.4 1-12 44.2 44.3 42.2 13-24 41.8 25-36 42.5 42.6 37-48 45.0 45.5 49-60 49.6 49.9 444 43.2 44.8 44.3 43.1 44.7 48.3 42.9 46.8 45.2 50.0 445 48.3 49.1 45 0 50.5 51.2 47.5 50.7 48.9 44.9 49.4 50.3 45.2 50.0 49.2 48.1 49.6 50.7 50.9 50.7

(a) Obtain the time series plot and observe that do not have clear trend and seasonal pattern,

(b) Plot the time series data overlaid with the estimated trend component using single exponential smoothing. Run the program for different values of smoothing parameter between 0.1 and 0.4.

(c) Estimate the number of employee for the next five years using the optimum smoothing parameter. (d) Analyze the residuals using appropriate tests.

**2**. Consider the data employ.mtw in Minitab. To predict employment over six months in a sement of the metals industry use both single and double exponential smoothing as there is no clear trend or seasonal pattern in the data.

1. Predict the employment for next 6 months for both methods.
2. Compare the fit by double exponential smoothing method with that from single exponential smoothing through calculating MAD, MSD and MAPE
3. Plot the series along with predicted observations.

**Q.1)**Program:

**#Single Exponential Smoothing-----** x=c(44.2,44.3,44.4,43.4,42.8,44.3,44.4,44.8,44.4,43.1,42.6,42.4,42.2,41.8, 40.1,42,42.4,43.1,42.4,43.1,43.2,42.8,43,42.8,42.5,42.6,42.3,42 9.43.6, 44.7,44.5,45,44.8,44.9,45.2,45.2,45,45.5,46.2,46.8,47.5,48.3,48.3,49.1, 48.9,49.4,50,50,49.6,49.9,49.6,50.7,50.7,50.9,50.5,51.2,50.7,50.3,49.2,48.1)

n-length(x); a=0.5;

#--Smoothig-

m-zeros(n,1); x\_hat-zeros(n,1);

m[1]=x[1];

for (t in 2:n)

{

m[t]-a x[t]+(1-a)\*m[t-1];

cat(The first 10 smotthed values are:")

print(m[1:10]);

#Plot

- plot(m,type="I", col="green")

#Accuracy measures—

MAD-mean(abs(x-m));

MSD-mean(t((x-m))%%(x-m));

MAPE=mean(abs((x-m)/x));

cat("MAD", MAD, "\n")

cat("MSD", MSD, "\n")

cat("MAPE", MAPE, "\n")

**Q.2)** Program: Rstudio

#Double Exponential Smoothing-

library(matlab); x-c(44.2,44.3,44.4,434,42.8,44.3,444,44.8,444,43.1,42.6,42.4,42.2,418, 40.1,42,424,43.1,42.4,43.1,43.2,42.8,43,42.8,42.5,42.6,423,42.9,43.6, 44.7,44.5,45,44.8,44.9,45.2,45 2,45,45.5,46.2.46.8,475,48.3,483,49.1, 48.9,49.4,50,50,49.6,49.9,49.6,50.7,50.7,50.9,50.5,51.2,50.7,50.3,49.2,48.1)

n=length(x);

a=0.3; r=0.03;

#Smoothig-

m=zeros(n,1); b=zeros(n, 1); x\_hat zeros(n,1);

m[1]=x[1];

b[1]=x[2]-x[1];

x\_hat[1]=x[1];

for (t in 2:0){ m[t]-a x[1]+(1-a)\*(m[t-1)+b[t-1])

b[t]r (m[t]-m[t-1])+(1+r)\*b[t-1]

x\_hat[t]-m[t-1]+b[t-1]

}

cat('The first 10 smotthed values are:')

print(x\_hat[1:10]);

Plot

plot(x, type="i", col="red")

par(new TRUE)

#Plot

- plot(m,type="I", col="green")

#Accuracy measures—

MAD-mean(abs(x-m));

MSD-mean(t((x-m))%%(x-m));

MAPE=mean(abs((x-m)/x));

cat("MAD", MAD, "\n")

cat("MSD", MSD, "\n")

cat("MAPE", MAPE, "\n")

**Topic:4 Simulation of stationary processes.**

1. Simulate the stationary MA(1) process and plot the sample ACVF and ACF

**2**. Simulate the stationary AR(2) process X-0.3X-1 -0.3X-2-2

**3**. Simulate the stationary AR(1) process and plot the sample ACVF and ACF

**4.** Check the stationarity, causality and invertibility of the following processes.

(a) all roots are outside the unit disk

(b) all roots are inside the unit disk

(c) All roots are on the unit circle

(d) Two mots are outside and one root inside the unit disk

(e) One root is outside, one root is inside the unit disk and one roof on the unit circle.

**Q.1)** Program: Rstudio

par(mfrow=c(2,1))

plot(arima.sim(list(order=c(0,0,1),ma=0.5),n=100),ylab="x", main (expression(MA(1)theta-0.5))) plot(arima.sim(list(order=c(0,0,1),man- 0.5),n=100),ylab="x", main=(expression(MA(1)theta=-0.5)))

**Q.2)** Program: Rstudio

z=c(1,-0.3,0.3)

(a=polyroot(z)[1])

arg Arg(a)/(2\*pi)

1/arg

set.seed(90210)

ar2-arima.sim(list(order-c(2,0,0),ar-c(1.5,-0.75)),n=144)

plot(1:144/12,ar2, type="l",xlab="time")

abline(v 0:12,lty "dotted",lwd=2)

ACF-ARMA acf(ar-c(1.5,-0.75), ma=0,50)

plot(ACF,type="h",xlab="lag")

abline(h=0)

**Q.3) Program: Matlab**

**Q.4)** Program: Rstudio

p=2; q=2;

arp=c(-0.88,-1.9,1);

r\_ar-abs(polyroot(arp))

maq=c(0.7,0.2,1);

r\_ma-abs(polyroot(maq))

are',r\_ar);

cat(' roots of MA polynomial are',r\_ma);

if(abs(r\_ar)>1){

cat('causal stationary')

}else if (sum(r\_ar0){ cat('process is non causal stationary')

}else{

cat('non stationary')

}

if (abs(r\_ma)>1){

cat('invertible')

}else{

cat('non invertible')

}

**Topic:5 Prediction of stationary processes, plotting Periodogram and spectral density function**.

**1.** Simulate the stationary AR(2) process Xt 0.7X1-1+0.10Xt-2+Zt. Estimate the parameters 1 and by using Yule-Walker equations.

1. Plot the spectral density function of white noise,AR(I) and MA(1) process for various parameters.

**Q.1)** Program Rstudio

#Esimation Of Ar parameters using Yule-Walker est-

z=c(1,0.7,0.10)

(a=polyroot(z)[1])

arg-Arg(a)/(2\*pi)

1/arg

set.seed(90210) ar2-arima.sim(list(order=c(2,0,0),ar=c(0.7,0.1)),n=144)

par(mfrow=c(3,1))

par(mar c(1,1,1,1))

plot(1.144/12,ar2,type="I",main="Time series plot for AR2", xlab="Time(one unit=12 points)") abline(v=0:12,lty="dotted",lwd=2)

ACFARMAacf(ar-c(0.7,0.1),ma-0,50)

plot(ACF type="h"main="ACF of AR Process",xlab-"lag")

abline(h=0)

#--Yule Walker Estimation-

ar2 yw ar yw(ar2,order=2)

ar2 ywSx mean

ar2 ywSar

ar2\_yw$var pred

rec.pr predict(ar2 yw,n ahead=24)

U-rec pr$pred+rec pr$se

L-rec prSpred-rec pr$se

Minx= min(ar2,L);maxx=max(ar2,U);

ts.plot(ar2,rec.pr$pred,main="Time Series Plot",xlim-c(1,n),ylim c(minx,maxx)) lines(rec.pr$pred,col="red" type="o")

lines(U,col="blue",Ity="dashed")

lines(L,col="blue",lty="dashed")

**Q.2)** Program: Rstudio

s=2; theta=0.8; phi-0.9,

par(mfrow-c(3,1))

-WN(0,s^2)—

fl=function(s) (s^2)/(2\*pi);

plot. function(1,0,pi)

MA(1)

f2-function(lamda) ((s^2)(1+2 theta\*cos(lamda)+theta^2)/(2\*pi).

plot.function(12,0,pi)

AR(1)

F3=-function(lamda) ((s^2)(1/(1-2 phi\*cos(lamda)+phi^2)))/(2\*pi)

plot.function(f3,0,pi)

**13.1 Time series plot and realization of various time series**.

**1.** Obtain the realization of the following.

(a) Gaussian white noise.

(b) White noise.

(c) Binary process

(X,, = 1,2,..., 500] for which P(X, 1) = P(X) = -1) = 1/2. [i.i.d. noise).

1. Simple symmetric random walk (S,,t=1,2,.,200) where S, X, t = 1,2,... and (X) is = i.i.d. noise.

Program: --- **Simulation Of Gaussian Noise and White Noise----**

zirnorm(n,0,1);

z2 rnorm(n,0,2); z3 rnorm(n,V,83);

z4-rnorm(n,0,s4);

par (mfcol=c(4,1))

plot (z1,main 'WN(0,1)" ylim-c(-20 20),type="1")

plot (z2,main 'WN(0,4)" ylim=c(-20, 20),type="1") =

plot (23,main 'WN(0,25) ", ylim=c(-20,20),type="1") =

plot (z4,main 'WN(0, 64)", ylim-c(-20,20),type="1') =

Simulation Einary Process and random walk--- n=500: Trinom(n,1,p);

par (nfcol-c4,1)) ;

par (nar-c(1,1,1,1)) plot (x,main "Similated Bcnary process"); 5-cumsum(x);

plot (s,main= "Eimulation Cf Random Walk")

1. Obtain the realization of the following time series (X,, = 1,2,300) (a) where X, = U cos (01) + V sin (0), 0 € -л,л and U- Normal(0,1), V = √3V where V₁~ Uniform (-1.1).

**Program:**

1=50;

Urncrm(n, 0, 1);

V=sqrt(3)

rinif(n,-1,1);

f=1/6 : t=cbind(seq(1,n,1)): I=\*cos(2\*pi\*t)+V\*s in (2\*pi\*f\*t);

Y=cos (2\*pi\*f \*t)+sin(2\*pi\*f \*t);

#Plotting part

plot (X,type-"1")

par (mfcol-c(2,1)

plot (Y,type="1")

**13.2 Estimating the trend component with linear filters**.

1. Consider the following time series data on 60 consecutive observations

53.5 53 53.2 52.5 53.4 56.5 65.3 70.7 66.9 58.2 553 53.4 52.1 52.3 51.5 51.5 52.4 53.3 55.5 64.2 65.6 69.3 58.5 55.3 55.6 53.6 51.5 51.7 515 52.2 57.1 63.6 68.8 68.9 60.1 53.9 53.3 53.1 53.5 53.5 53.9 57.1 64.7 69.4 70.3 62.6 57.9 55.8 54.8 54.2 54.6 | 54.3 54.8 58.1 68.1 73.3 75.5 66.4 60.5 57.7

(**a)** Observe the time series plot of the above data and estimate the trend component of it using 5-term moving average filter.

**(b**) Plot the time series data overlaid with the estimated trend component. Also obtain the resid- ual plot.

**(c**) Also estimate the trend component of it using 3-term moving average filter and 7-term mov- ing average filter. And compare the three estimates of the trend component.

program:

Simulate the stationary

AR(2) process

X = 0.7X-1 +0.10X1-2+Zr.

Estimate the parameters i and 2 by using Yule-Walker equations.

Program:

#-----Esimation of Ar parameters using Yule-Walker

est- (a-polyroot (z) [1])

arg-Arg (a)/(2+pi) 1/arg set.seed(90210)

ar2=arma.eim(list (order=c(2,0,0),

ar=c(0.7,0,1)),n=144)

par (mfrov=c(3,1))

par (mar=c(1,1,1,1)))

plot (1 144/12, ar 2, type '1",main= Time series plot fcr AR2", = xlab="Tine (one unit-12 points)")

abline v 0:12, lt y="dott ed', lwd=2)

ACF=ARMAacf (arc (0.7,0.1) ma=0,50)

plot (ACF, type="h", main="A CF of AR Process",xlab="Lag")

abline(h=0)

#Ytle Walk or Eetination--- ar2\_7w-ar.yv (ar2,order-2) ar2\_7w8x.me an ar2\_yw3 ar ar2\_7wSvar.pred rec. or predict (arzyw.n.ahead=24) U-res pr pred+rec. pr$se L-res.pr$pred-rec. pr$se minx-min (ar 2,L);

maxx =max( ar 2,U); ts.plot (arz,rec, pr$pred main="Time Series Plot ",

xin=c(1,n),

71im=c(minx, maxx)) lines(rec pr pred.col="red" type="o") lines(col="blue",1ty="dashed") lines(L,col="blue",lty="dashed")

Outpat Coefficients: 1 0.7233

1. Plot the spectral density function of white noise ABCT) and MA(1) process for various parameters.

Program:

theta-0.8,

par (mfrov=c(3, 1)))

#- VN (0,82)- phi-0.9, 5-2 fi-function (s) (s 21/12 pi);

plot.function(f1,3,pi)

#- ---MA(1).

f2 function (lamda) (s2)+(1+2\*tieta\*ccs (landa) +theta 2))/(2\*pi);

plot function (f2.3.pi #- -AE(1) f3-function (lami a) (e-1)+(1/(1-2 phi coe (lamda) +phi)))/(2\*pi)

vn Lorm(n,0,321; SW-Accs (2\*pi\*t\*f0) +B+sin(2\*pi\*t\*f();

x=y-nean (y);

1-c(rop (1,q))

if (1%%2==0)

fq-(1-2)/2helse {q=(1-1)/2

} = ((1: q))/n;

for(i in 1: q) a[1]=2\*sum(x\*08 (2\*pi\*w [1] \*t))/n [i]=2\*sum(x\*sin(2\*pi\*w [i] \*t))/n: [i]n (a[i]%%a [-] b[i]%+%b[i])/2;

} req\_freq=w[p=max(p)];

period=1/req freq: cat (Actual freq', 0);

cat(Required freq' req\_freq); cat(actual period 1/0);

cat(Required per-oc' period);

cat('total variat:01',sum(x+1));

cat(\n\t sum of intensities',

sun (p))

Jutput: Actual freq 0.435 actual period: 2.2988E1 Required freq: 0.4333333 Required period: 2.307692 total variation: 422, E21 sun of intensities 422.2079

r\_ma abs(pclyr oot (maq)))

#roots of NA polynomial

cat(' roots of AR polynomial are',r\_ar );

cat(' roots of MA polynomial are', na);

if (abs(rar)>1

) { cat('causal stationary') J

else if

(eum (r\_ar<1)>0) [

cat (process is non causal stationary') Felsef cat('non stationary) } if (abs (r\_ma) >1)

{ cat('invertible')

}

Else

{ cat('non invertible')

}

Outpat roots of AR polyaonial are

0.385103 2.285103 roots of MA polynomial are 0.83566 0.83666

process 18 non causa. stationary non invertible

**13.5 Prediction of stationary processes,**

**plotting Periodogram and spectral density function**

. 1. Obtain the periodogram of the data.

Program:

#per

lod og am n-21); tt((1));

rm(list=1s(all=TRUE))

A=1; B=1; f0-87/200;

s2=1;

par(nfrow=c(2,1!)

plot(arima.sin(list (order=c(0,0,1),

ma=0.5), n=100),y lab="x",

main=(expression (MA(1)theta=0.5)))

plot(arima.sin(list (order=c(0,0,1),

ma=-0.5),

n=100),

ylab="x",

main=(expression (MA(1) theta ==-0.5))}

**Simulate the stationary AR(2) process X=-0.3X-1 +0.3X-2+Z1.**

Program:

z=c(1, -0,3,0.3) (a-polyroot(z) [1])

arg-Arg (a)/(2+pi) 1/arg set.seed(9021))

ar 2-arima, sim(11st (order=c(2,0,0),

ar=c(1.5,-0 75)), n=144) plot (1:144/12,

ar 2, type="1",xlab="time")

ablire (v=0:12, lty="cott ed",1wd=2)

ACF-ARMAacf (ar=c (1.5,-0.75), ma=0,50)

plot (ACF,type="h",xlab="lag")

ablire(h=0)

1. **Check the stationarity, causality and invertibility** of the following processes.

(a) all roots are outside the unit disk.

(b) all roots are inside the unit disk.

(c) All roots are on the unit circle.

(d) Two roots are outside and one root inside the unit disk.

(e) One mor is outside one root is inside the unit disk and one root on the unit circle

**Program:**

arp=c(-0.88, -19,1);

maq=c(0.7,0.2,1);

# coefficients of ar and ma polynomial T

ar-ats (polyroot (arp))

#roots of ar polynomial

vn Torm (n,0,321;

if (1%%2==0) 3W-Accs (2\*pi\*t\*f0) +B+sin(2\*pi\*t\*fC);

x=y-nean (y); 1-c(rop (1,q))

for(i in 1: q) fq-(1-2)/2h

else {q=(1-1)/2} =((1: q))/n;

a[1]=2+sum(x+cos (2\*pi\* [1] \*t))/n [i]=2 sum(x\*sin(2\*pi\*w [i] \*t))/n;

[i]n (a[i]%%a [-] b[i]%+%b[i])/2;

}

req\_freq=w[p =max(p)];

period=1/req freq:

cat (Actual freq',f0);

cat(Required freq' req\_freq);

cat(actual poriod 1/0);

cat(Required per.oc' period);

cat(total variatio1',sum(x+1));

cat \n\t sum of intensities', sun (p))

**Jutput:**

Actual freq 0.435

actual period: 2.2988E1

Required freq: 0.4333333

Required period: 2.307692

total variation: 422, E21

sum of intensities 422.2079

**Survival analysis;**

**Que 1]** The following are the time in days between successive serious earthquakes worldwide. An earthquake is included in the data set if its magnitude was at least 7.5 on Richter scale, or if over 1000 people were killed. Recording starts on 16-th of December 1902 and ends on 14-th March 1997. There were 63 earthquakes recorded altogether, and so 62 waiting times.

840,157,145,44, 33,121,150,280, 434, 736, 584, 887, 263,1901, 695, 294, 562,

721, 76, 710, 46, 402, 194, 759,319, 460, 40,1336, 335, 1334, 454, 36, 667,

40, 556, 99, 304, 375, 567, 139, 780, 203, 436, 30, 384, 129, 9, 209, 599, 83,

832, 328, 246, 1617, 638, 937, 735, 38, 365, 92, 82, 220

Assume that the earthquakes occur at random and hence waiting times are exponentially distributed. Obtain

1. Point estimate of scale parameter (λ)
2. Interval estimate of λ for confidence coefficient of 95%
3. Check the assumption of exponentiality using simple graphical methods

**R Code:**

t =c(840,157,145,44,33,121,150,280, 434, 736, 584, 887, 263,1901,695,

294, 562, 721, 76, 710, 46, 402, 194, 759,319, 460, 40,1336,335, 1334,

454, 36, 667, 40, 556, 99, 304, 375, 567, 139, 780, 203,436, 30, 384, 129,

9, 209, 599, 83, 832, 328, 246, 1617, 638,937, 735, 38, 365, 92, 82, 220)

N=length(t) # The number, n, of components of vector t.

n

#[1] 62

estlam=n/sum(t) # Estimate of lambda.

estlam

#[1] 0.002288921

LCL=estlam\*2\*qgamma(.025,shape=n)/(2\*n) # Lower confidence limit (LCL)

LCL

#[1] 0.001754903

UCL<-estlam\*2\*qgamma(.975,shape=n)/(2\*n) # Upper confidence limit (UCL).

UCL

#[1] 0.002892792

#We use graphical method to check exponentiality . We plot empirical and estimated survival curves on the same graph paper. If the two curves are close then the model is appropriate.

d<-c(rep(1,62))

cd<-cumsum(d)

emps<-(n-cd)/n # Empirical survival function.

t<-sort(t)

s<-exp(-(estlam\*t)) #Estimate of the survival function.

plot(t,emps,"o", pch = 1, lwd=2,xlab="Waiting time", ylab="Survival time", main=" survival functions",cex=.7)

points(t,s,"o",pch=2) # Add the plot of empirical survival function.

legend (locator (1), legend = c ("empirical", "estimated exponential", pch = 1:2, cex = 0.7)

**Que 2] Exponential Distribution (Type I censored data)**

Suppose 20 items from an exponential distribution are put on life test and observed for 150 hrs. During the period 15 item fail with the following life times, measured in hrs: 3, 19, 23, 26, 27, 37, 38, 41, 45, 58, 84, 90, 99, 109, 138. Test the hypothesis Ho : — 65 against H\ : \x > 65 at 2.5% level of significance.

t<-c(3,19,23,26,27,37,38,41,45,58,84,90,99,109,138) # Vector of failure times

n<-20 # Sample size.

r<- length (t); r # Number of failures

[1]15

mu0<-65 # Specified value of mean.

alpha<-0.025 # Level of significance.

t0<-150

p0<-1-exp(-t0/mu0) # Specified proportion

p0

[1]0.9005094

estmu<-(sum(t)+(n-r)\*t0)/r # Estimated value of population mean.

estmu

[1]105.8

alpha1<-alpha-exp(-n\*t0/mu0) # Level of significance when zero is included in the critical region.

alpha1 # Print adjusted level of significance

[1]0.025

qnorm((1-alpha1))

[1] 1.959964

u0<-(estmu - mu0)/mu0

u0

[1]0.6276923

nz0<-u0\*(n\*p0)^.5

nz0

[1]2.663826

dz1<-2\*u0\*(1-p0)\*log(1-p0)/p0

dz1

[1] - 0.3200725

dz2<-(1-p0)\*u0 ^2

dz2

[1]0.03919905

z0<-nz0/(1-dz1+dz2) ^.5 # Observed value of test statistic

z0

[1]2.284824

#Conclusion - ZQ > 2i\_Q.. Therefore Ho is rejected and we conclude that the average life is greater than 65 hours.

**Que 3] Computation of Jb statistics:**

x= c(300,650,800,1280,1710,1920,2050,2200,2600,2950,3150,3400,3500,4350,5700,8100)

y=x/2

z= c(x,y)

cr= rank(z)

cr

#[1]2 6 7 12 17 19 20 22 23 25 26 27 28 30 31 32 1 3 4 5 8 9 10 11 13

#[26] 14 15 16 18 21 24 29

n=length(z)

n

#[1] 32

rl=cr[1:(n/2)]

rl

#[1] 2 6 7 12 17 19 20 22 23 25 26 27 28 30 31 32

jb= data.frame(x,y,rl)

jb

#

x y rl

1 300 150 2

2 650 325 6

3 800 400 7

4 1280 640 12

5 1710 855 17

6 1920 960 19

7 2050 1025 20

8 2200 1100 22

9 2600 1300 23

10 2950 1475 25

11 3150 1575 26

12 3400 1700 27

13 3500 1750 28

14 4350 2175 30

15 5700 2850 31

16 8100 4050 32

m= length(x)

estjb=(1/(m\*(m-1)))\*(sum(rl)-m\*(m+3)/2)

estjb

#[1] 0.7291667

#Conclusion- he J1/ 2 test rejects the null - hypothesis of exponentiality at the 5% level of significance

**Que 4] Kaplan- Meier estimate for failures of vanes (Illustration 5.3)**

time<-c(142,149,320,345,560,805,1130,1720,2480,4210,5230,6890);

length(time);

status=c(1,1,1,0,1,1,0,1,0,0,1,1);

length(status);

#[I] 12

ctrl< — data. frame(time,status);

attach(ctrl);

library (survival);

km.ctrl< — survfit(Surv(time,status==1));

summary (km. ctrl);

**Que 5] Non- parametric estimator of survival function**

#R-code for plot empirical survival function

library (survival)

time=c(17.88,28.92,33.0,41.52,42.12,45.60,48.48,51.84,51.96,54.12,55.56,67.80,68.64,68.64,68.88,84.12,93.12,98.64,105.12,105.84,127.92,128.04,173.4)

length(time)

#[1] 23

status=c(rep(1,23)) # 1 indicates complete and zero indicates censored

length(status)

#[1] 23

ball= data.frame(time,status)

attach(ball)

km.ball= survfit(Surv(time,status))

plot(km.ball,conf.int =F,xlab= "time",ylab= "survival function", main="Non-parametric estimator of survival function", cex=.6)