Institute of Actuaries of India

Subject CS2B – Risk Modelling and Survival Analysis (Paper B)

November 2019 Examination

INDICATIVE SOLUTION

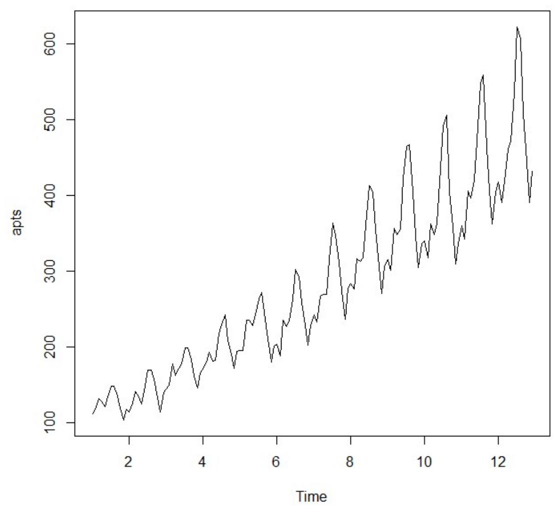
Introduction

The indicative solution has been written by the Examiners with the aim of helping candidates. The solutions given are only indicative. It is realized that there could be other points as valid answers and examiner have given credit for any alternative approach or interpretation which they consider to be reasonable.

Solution 1:

i)

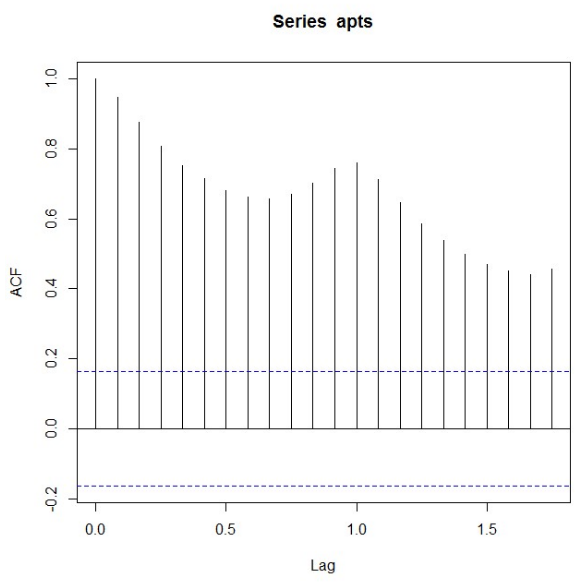
apts <- ts(AirPassengers, frequency=12)
plot(apts)



[2]

ii)

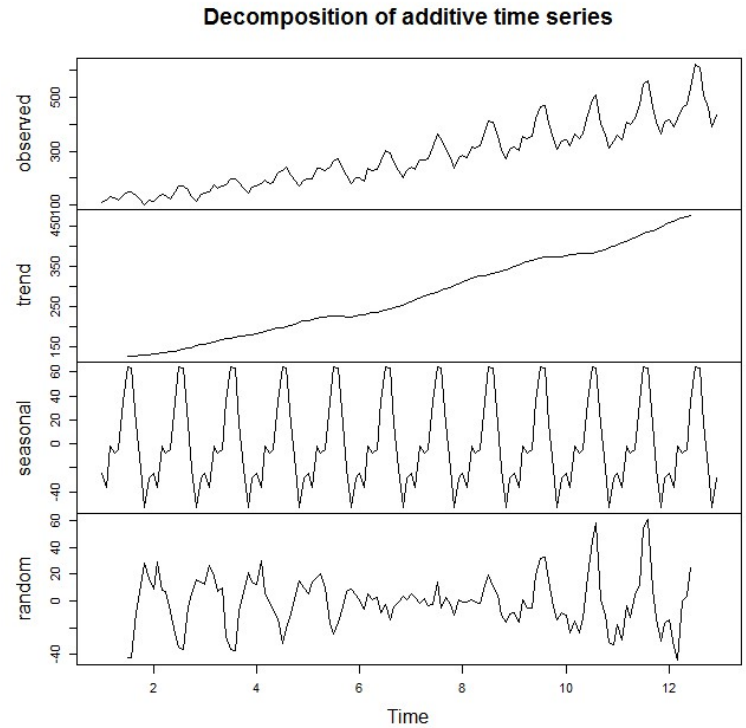
acf(apts)



[2]

iii)

f <- decompose(apts)
plot(f)



[3]

iv)

#Fit data
fit <- arima(AirPassengers, order=c(1,0,0), list(order=c(2,1,0), period=12))
fit

```
> fit

Call:
arima(x = AirPassengers, order = c(1, 0, 0), seasonal = list(order = c(2,
1,
    0), period = 12))

Coefficients:
         ar1     sar1    sar2
      0.9458  -0.1333  0.0821
s.e.  0.0284   0.1035  0.1078

sigma^2 estimated as 143.1:  log likelihood = -516.18,  aic = 1040.37
```

[3]

v)

fore <- predict(fit, n.ahead=24)

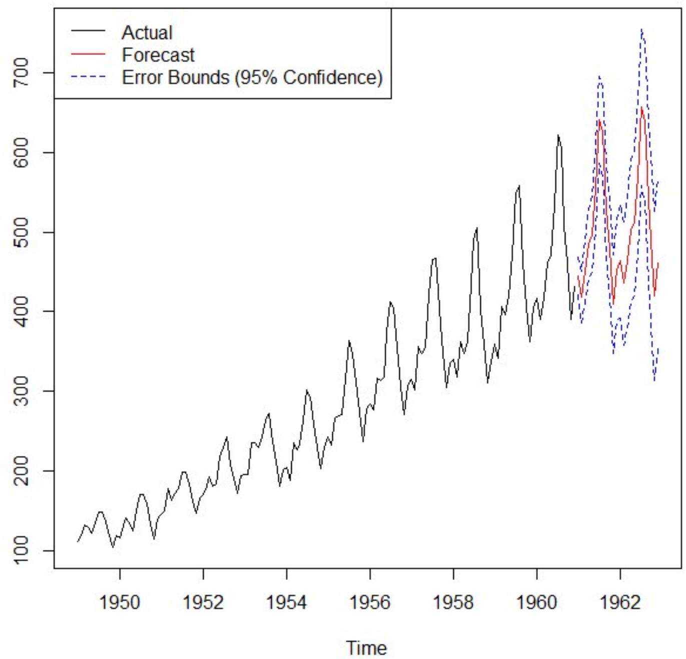
[2]

vi)

U <- fore$pred + 2*fore$se
L <- fore$pred - 2*fore$se

[3]

ts.plot(AirPassengers, fore$pred, U, L, col=c(1,2,4,4), lty = c(1,1,2,2))
legend("topleft", c("Actual", "Forecast", "Error Bounds (95% Confidence)"),
col=c(1,2,4), lty=c(1,1,2))

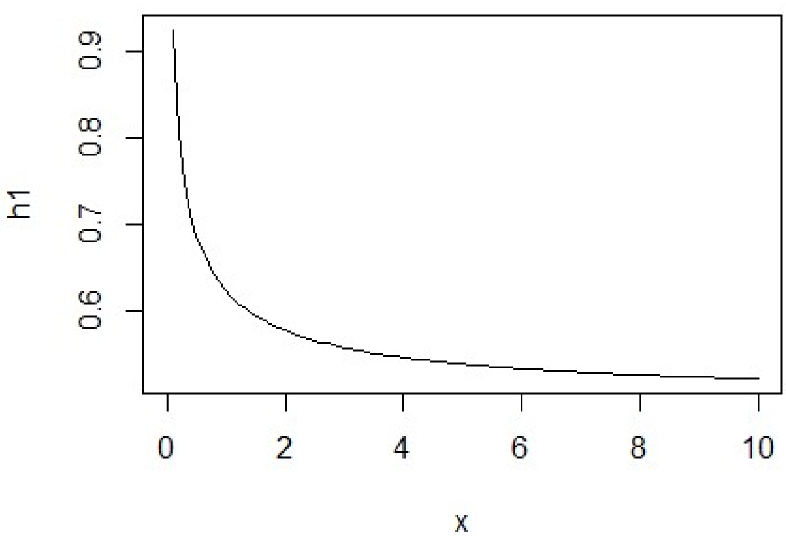


[5]
[8]
[20 Marks]

Solution 2:

i)

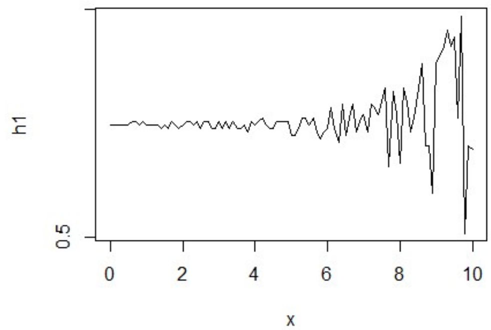
h1<-function(x) {dgamma(x,0.75,0.3)/(1-pgamma(x,0.75,0.3))}
plot.function(h1,0,10) [2]



[1]
[3]

ii)

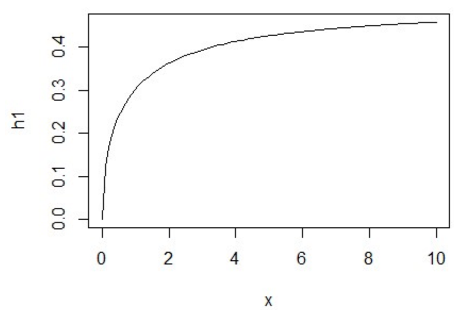
```
h2<-function(x) {dgamma(x,1,0.3)/(1-pgamma(x,1,0.3))}
plot.function(h2,0,10)
```



[1]

iii)

```
h3<-function(x) {dgamma(x,1.5,0.3)/(1-pgamma(x,1.5,0.3))}
plot.function(h3,0,10)
```



[1]

iv)

If $\alpha < 1$, it is a decreasing function of x and thus indicating a heavier tail than the exponential distribution. If $\alpha > 1$, it is a increasing function of x and thus indicating a lighter tail than the exponential distribution. If $\alpha = 1$ then the function is relatively constant in most part of the projection.

[2]

[7 Marks]

Solution 3:

i)

```
B=0.00000729
C=1.128

gmu<-function(x){
 Mu<-B*C^x
 Mu
}
```

[2]

```
qx<-function(x){
 1-exp(-gmu(x+1/2))
}
```

[2]

```
x<-55
```

```
ex<-0
npx<-1
for(i in 1:(100-x)){
  px=1-qx(x+i-1)
  npx=npx*px
  ex<-npx+ex}
ex
```

```
> ex
[1] 21.71408
```

[3]

[7 Marks]

Solution 4:

i)

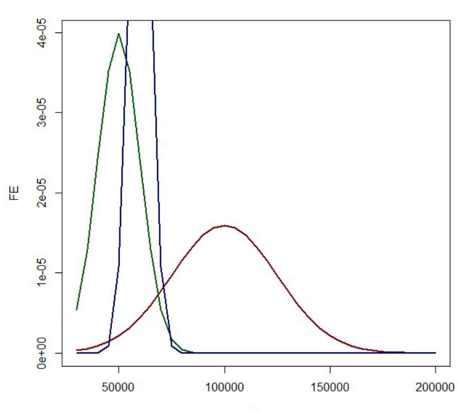
```
y<-seq(30000,200000,5000)
y

FE<-dnorm(y,50000,10000)
FPT<-dnorm(y,100000,25000)
FUL<-dnorm(y,60000,5000)

plot(y,FE,typ="l",col="darkgreen",lwd=2)
 lines(y,FPT,col="darkred",lwd=2)
 lines(y,FUL,col="darkblue",lwd=2)
```

```
> y
 [1]  30000  35000  40000  45000  50000  55000  60000  65000  70000  75000
80000
[12]  85000  90000  95000 100000 105000 110000 115000 120000 125000 130000
135000
[23] 140000 145000 150000 155000 160000 165000 170000 175000 180000 185000
190000
[34] 195000 200000
```

[1]



[4]

[5]

ii)

```
PE<-0.8
PPT<-0.05
PUL<-0.15
PxE<-dnorm(75000,50000,10000)
PxPT<-dnorm(75000,100000,25000)
```

PxUL<-dnorm(75000,60000,5000)
*P1<-PE*PxE / (PE*PxE+PPT*PxPT+PUL*PxUL)*
*P2<-PUL*PxUL / (PE*PxE+PPT*PxPT+PUL*PxUL)*

P1
P2

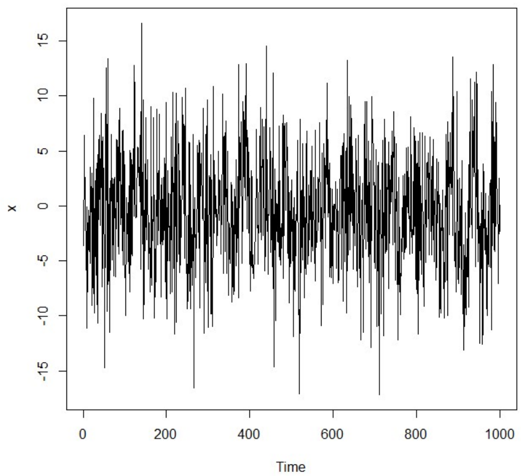
[3]

```
> P1
[1] 0.6944786
> P2
[1] 0.06584688
```

[4]
[7]
[12 Marks]

Solution 5:

i)
set.seed(90210) # so you can reproduce these results
*x = 2*cos(2*pi*1:1000/50 + .6*pi) + rnorm(1000,0,5)*
plot.ts(x)



[3]

ii)
*z1 = cos(2*pi*1:1000/50)*
*z2 = sin(2*pi*1:1000/50)*

[3]

iii)
summary(fit <- lm(x~0+z1+z2)) # zero to exclude the intercept

```
> summary(fit <- lm(x~0+z1+z2)) # zero to exclude the intercept

Call:
lm(formula = x ~ 0 + z1 + z2)

Residuals:
     Min       1Q   Median       3Q      Max 
-16.0671  -3.9817  -0.4643   3.0976  15.7080 

Coefficients:
   Estimate Std. Error t value Pr(>|t|)    
z1  -0.6784     0.2278  -2.978  0.00297 ** 
z2  -2.2071     0.2278  -9.690  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.093 on 998 degrees of freedom
```

```
Multiple R-squared:  0.09336,	Adjusted R-squared:  0.09154 
F-statistic: 51.38 on 2 and 998 DF,  p-value: < 2.2e-16
```

[4]

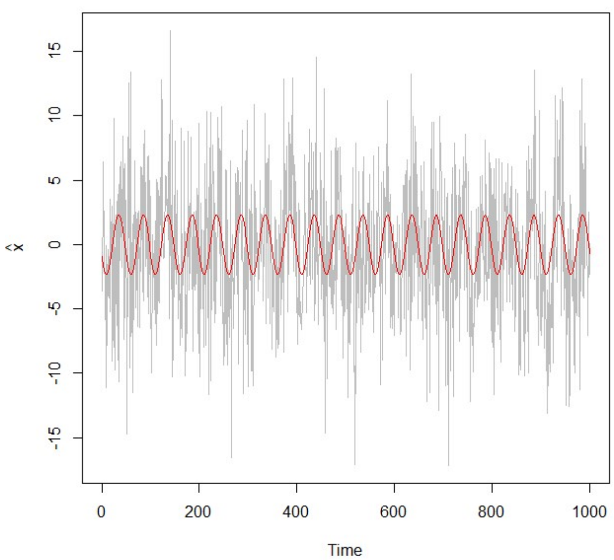
beta1 = -0.6784
beta2 = -2.2071
phi = atan(-beta2/beta1)
phi
A = beta1/cos(phi)
A

```
> phi
[1] -1.27259
```

```
> A
[1] -2.309008
```

[3]
[7]

iv)
plot.ts(x, col=8, ylab=expression(hat(x)))
lines(fitted(fit), col=2)



[3]
[16 Marks]

<u>Solution 6</u>:

i)
*H2S<-function(x){ 0.007*x}*
*H2D<-function(x){ 0.001*x}*
S2H<-function(x){ 0.006(100-x)}*
S2D<-function(x){ 0.002(100-x)}* [3]

transmat<-function(x){
M<-matrix(0,nrow=3,ncol=3)
M[1,1]<-1-H2S(x)-H2D(x)
M[1,2]<-H2S(x)
M[1,3]<-H2D(x)
M[2,1]<-S2H(x)
M[2,2]<-1-S2H(x)-S2D(x)
M[2,3]<-S2D(x)

```
 M[3,1]<-0
 M[3,2]<-0
 M[3,3]<-1
 M
}
```

[3]

```
n=30
B<-c(1,0,0)
for (i in 1:4){
 B=B%*%transmat(n+i-1)}
B
```

```
> B
          [,1]      [,2]      [,3]
[1,] 0.5451076 0.2608138 0.1940787
```

Hence the probability of Healthy person aged 30 will be in Sick state after 4 years is 0.2608138.

[3]

[9]

ii)

```
n=25
C<-c(0,1,0)
for (i in 1:7){
 C=C%*%transmat(n+i-1)}
C
```

```
> C
          [,1]      [,2]      [,3]
[1,] 0.3946025 0.1720589 0.4333386
```

Hence the probability of sick person aged 25 will be in Death state after 7 years is 0.4333386.

[3]

[12 Marks]

Solution 7:

i)

```
patients<-c(1:15)
time<-c(6,6,12,18,27,27,30,36,39,39,54,57,60,60,60)
censtat<-c(1,0,1,0,1,0,0,0,1,0,0,1,0,0,0)
data1<-data.frame(patients,time,censtat)
data1
library(survival)
model1<-survfit(Surv(data1$time,data1$censtat)~1)
data<-summary(model1)
data
```

```
> data
Call: survfit(formula = Surv(data1$time, data1$censtat) ~ 1)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    6     15       1    0.933  0.0644        0.815            1
   12     13       1    0.862  0.0911        0.700            1
   27     11       1    0.783  0.1115        0.593            1
   39      7       1    0.671  0.1409        0.445            1
   57      4       1    0.503  0.1797        0.250            1
```

[8]

ii)

```
tj=c(6,12,27,39,57)
dj=c(1,1,1,1,1)
```

nj=c(15,13,11,7,4)
1-dj/nj

Lambda=cumsum(dj/nj)
SNA<-exp(-Lambda)
SNA

```
> 1-dj/nj
[1] 0.9333333 0.9230769 0.9090909 0.8571429 0.7500000

> SNA
[1] 0.9355070 0.8662431 0.7909672 0.6856719 0.5340018
```

[5]

iii)
survdata<-data.frame(tj,nj,dj,data$surv,SNA)
survdata

```
> survdata
  tj nj dj data.surv       SNA
1  6 15  1 0.9333333 0.9355070
2 12 13  1 0.8615385 0.8662431
3 27 11  1 0.7832168 0.7909672
4 39  7  1 0.6713287 0.6856719
5 57  4  1 0.5034965 0.5340018
```

[3]
[16 Marks]

Solution 8:

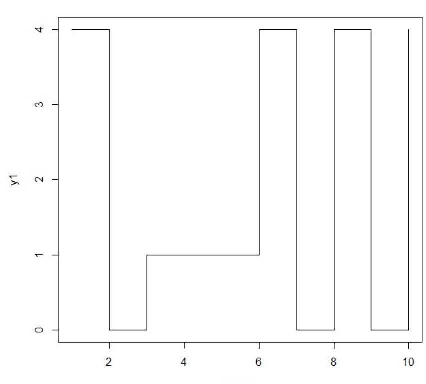
i)
set.seed(101010)
*x1 = 2*rbinom(11, 1, .5) - 1 # simulated sequence of coin tosses*
*x2 = 2*rbinom(101, 1, .5) - 1*
*x3 = 2*rbinom(1001, 1, .5) - 1*

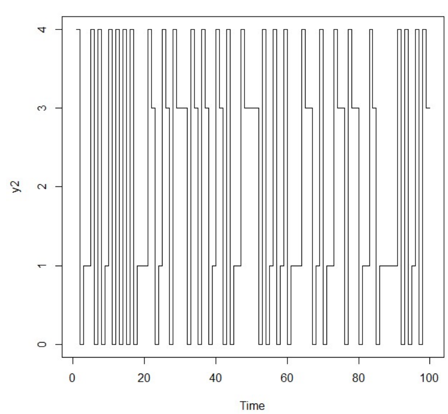
[2]

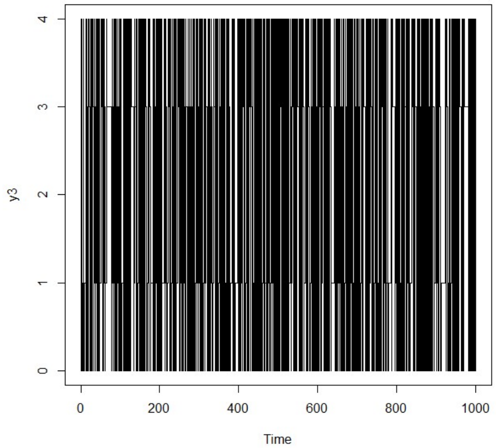
y1 = 2 + filter(x1, sides=1, filter=c(1.5,-.5))[-1]
y2 = 2 + filter(x2, sides=1, filter=c(1.5,-.5))[-1]
y3 = 2 + filter(x3, sides=1, filter=c(1.5,-.5))[-1]

[3]

plot.ts(y1, type='s')
plot.ts(y2, type='s')
plot.ts(y3, type='s')







[3]

[8]

ii)

c(mean(y1), mean(y2),mean(y3)) # the sample mean
c(var(y1), var(y2),var(y3)) # the variance

```
> c(mean(y1), mean(y2),mean(y3)) # the sample mean
[1] 1.900 1.970 1.975
> c(var(y1), var(y2),var(y3)) # the variance
[1] 3.433333 2.615253 2.582958
```

As the sample size increases, volatility reduces. [2]

[10 Marks]

****