Institute of Actuaries of India

Subject CS2-Paper B – Risk Modelling and Survival Analysis

June 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:

```
set.seed(2000)
n<-rpois(5000,500)
for(j in 1:5000) {y<-rgamma(n[j],250,0.5);z<-pmin(y,350);Agg_Claim[j]<-sum(z)}
Agg_Claim
```

```
> Agg_Claim
 [4995] 165200 176400 175700 177100 175700 171850
```

[4 Marks]

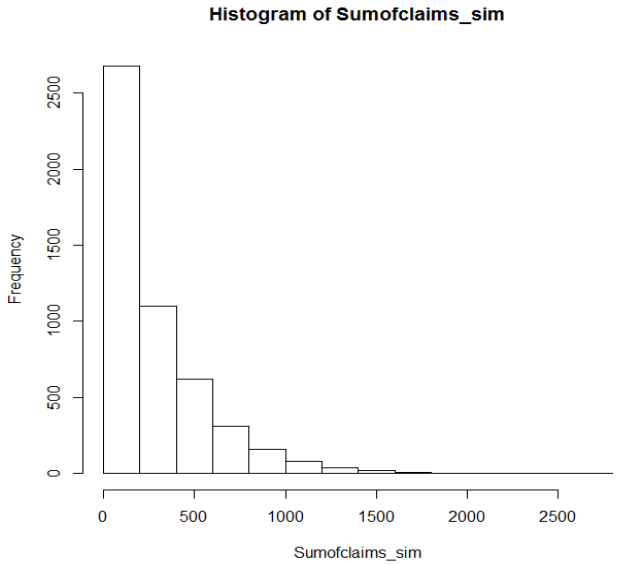
Solution 2:

i)
```
set.seed(2000)
rate1 <- 0.006
q1 <- 0.002
n1 <- 300
rate2 <- 0.007
q2 <- 0.001
n2 <- 500
rate3 <- 0.006
q3 <- 0.003
n3 <- 200
Sumofclaims_sim <- numeric(5000)

Total_policies = n1+n2+n3

for (j in 1:5000){
  x<-numeric(Total_policies)
  for (i in 1:Total_policies)
  {if (i<= n1){rate=rate1} else{if(i<=n1+n2){rate=rate2}else{rate=rate3}}
   x[i] <- rexp(1,rate=rate)}

  death<-numeric(1000)
  for (i in 1:Total_policies)
  {if (i<= n1){prob=q1} else{if(i<=n1+n2){prob=q2}else{prob=q3}}
   death[i] <- rbinom(1,size=1,prob)}

  sim_claim <- x*death
  sim_claim<-sum(sim_claim)
  Sumofclaims_sim[j] <- sim_claim
}
Sumofclaims_sim
```

```
[4981]  48.0509705 180.6917655 387.8498483 158.5486544  79.6699608 718.4442927
[4987] 260.0237473  58.7869147   0.0000000 824.1810537 121.5688946 393.4891939
[4993]  34.2667969   0.0000000   0.0000000 323.9416166   0.0000000  47.5719195
[4999] 400.2364413 155.3775077
```

ii)
```
summary(Sumofclaims_sim)
hist(Sumofclaims_sim)
```

```
> summary(Sumofclaims_sim)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   29.03  175.60  260.88  393.86 2664.99
```



[Note the numbers will change with change in seed]
The claims seem to be highly skewed with Q1 equal to 29.03, Q2 equal to 260.88 and Q3 equal to 393.86. The highest value is 2664.99.

The distribution seems to follow exponential distribution, which is consistent with underlying distribution of claims.

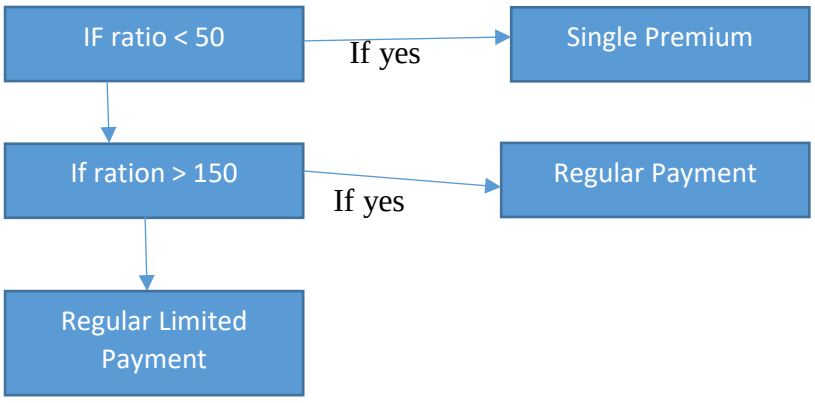
[17 Marks]

Solution 3:

i) For the same sum assured, the single premium will be much higher than regular premium, therefore the ratio of sum assured to first year premium will be smaller for single premium than regular premium. The ratio for limited premium policies will be in between single premium and regular premium.

It is observed that the ratio is above 150 means the policies are Regular Premium Policies and if the ratio is below 50 means the policies are Single Premium Policies. If the ratio falls between 50 and 150 indicates that the policies are Regular Limited Payment Polices.

Binary Decision Tree:



ii) Sum Assured are usually in lakhs/Crores and the Premium are usually quoted in thousands/lakhs. So, if (Lakhs, thousands) are used as scales then the SA would totally dominate the calculations and the premiums would effectively be ignored. However, if the SA and premium are converted to 100s and 1000s respectively, then SA/Premium ratio provides a numerically similar scale for similar type of policies.

iii) The following R code explain the binary decision tree:

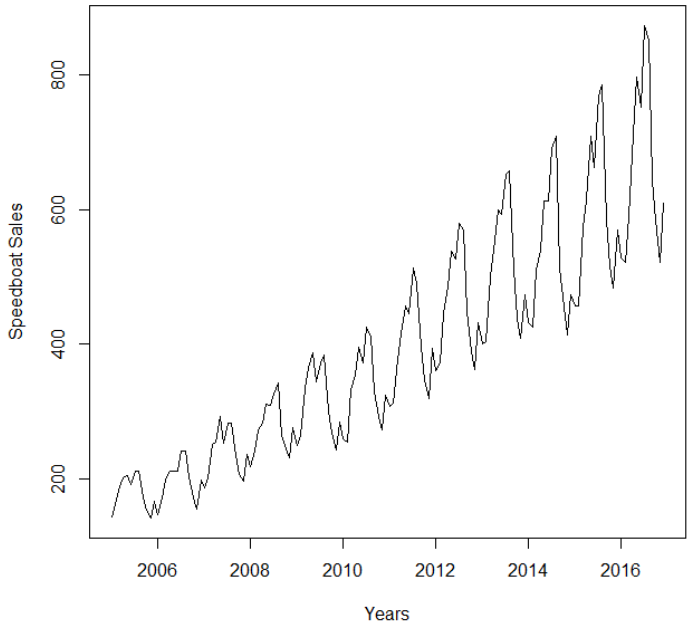
```
SA<-5000000
> FP<-250000
> Ratio<-(SA/100)/(FP/1000)
> guess<-ifelse(Ratio<50,"Single Premium",ifelse(Ratio>150,"Regular Premium","Regular Limited Premium"))
> guess
> guess
[1] "Regular Premium"
```

[10 Marks]

Solution 4:

i)

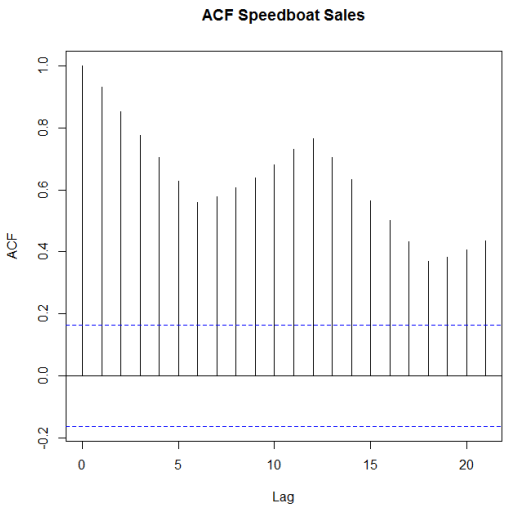
```
data = read.csv(PATH/Speedboat-Sales.csv')
data = ts(data[,2],start = c(2005,1),frequency = 12)
plot(data, xlab='Years', ylab = 'Speedboat Sales')
```

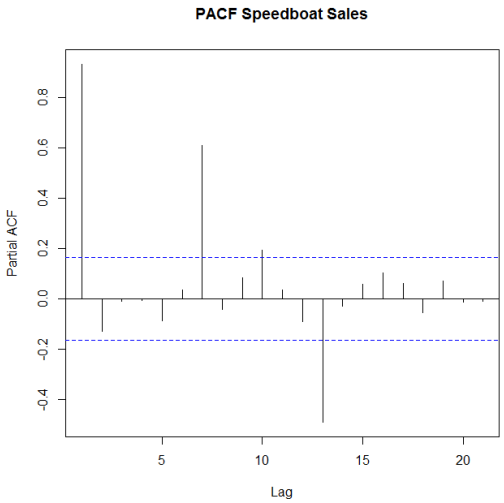


Clearly the above chart has an upward trend for speedboat sales and there is also a seasonal component.

ii)

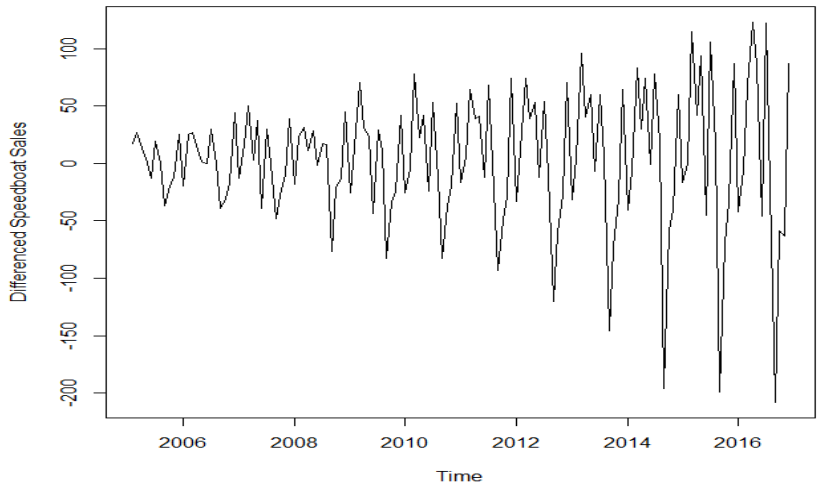
```
par(mfrow = c(1,2))
acf(ts(data),main='ACF Speedboat Sales')
pacf(ts(data),main='PACF Speedboat Sales')
```

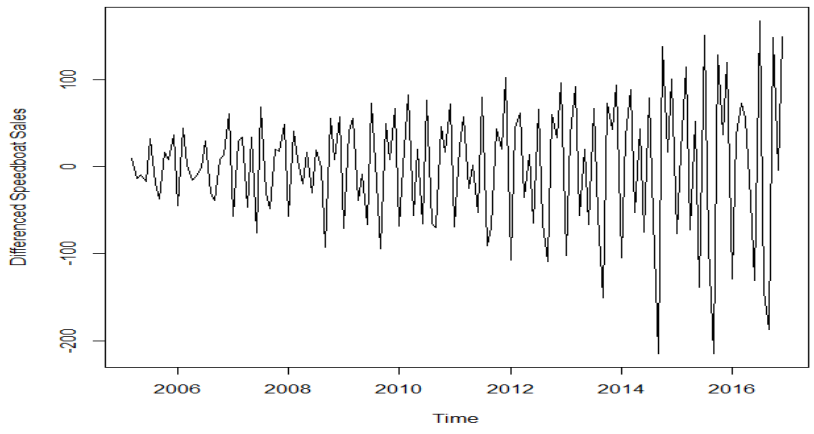




iii) *par(mfrow = c(1,1))*
diff_data = diff(data)
plot(diff_data,ylab='Differenced Speedboat Sales')

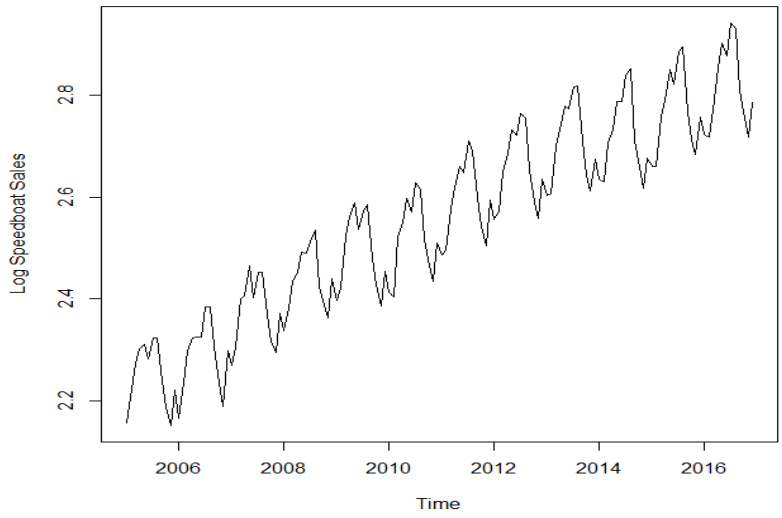
diff_data_2 = diff(diff_data)
plot(diff_data_2,ylab='Differenced Speedboat Sales')





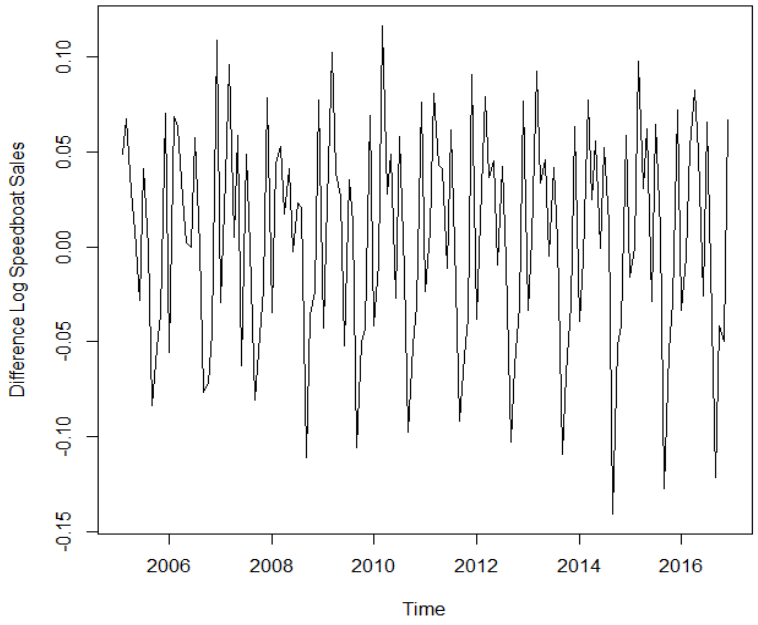
The above series (d=1) is not stationary on variance i.e. variation in the plot is increasing as we move towards the right of the chart. We need to make the series stationary on variance to produce reliable forecasts through ARIMA models.

iv) *Log_data = log10(data)*
plot(Log_data,ylab='Log Speedboat Sales')

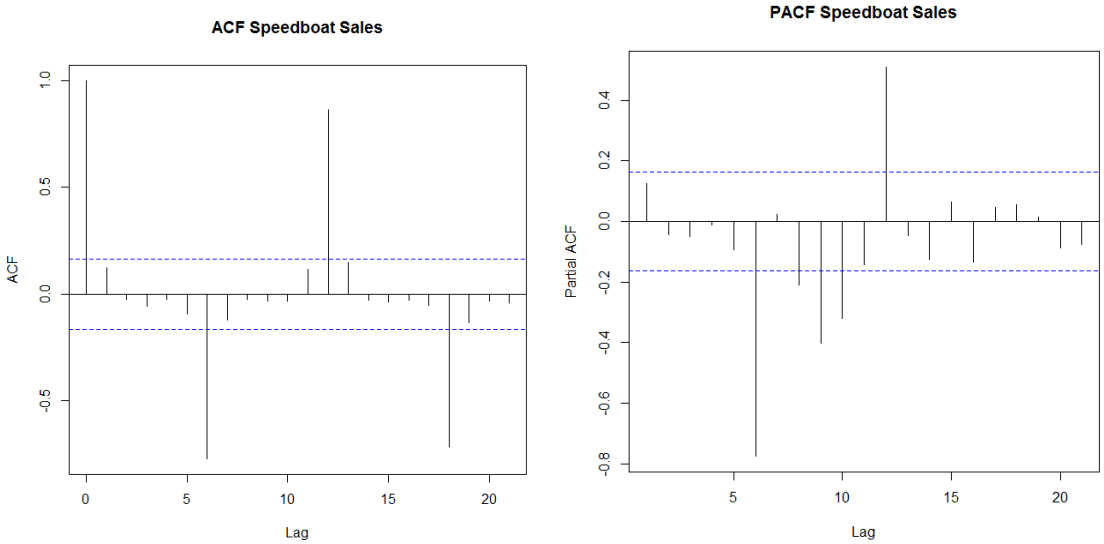


v) *Diff_log_data = diff(Log_data)*

plot(Diff_log_data,ylab='Difference Log Speedboat Sales')



vi) *par(mfrow = c(1,2))*
acf(ts(Diff_log_data),main='ACF Speedboat Sales')
pacf(ts(Diff_log_data),main='PACF Speedboat Sales')



Since, there are enough spikes in the plots outside the insignificant zone (dotted horizontal lines) we can conclude that the residuals are not random. This implies that there is juice or information available in residuals to be extracted by AR and MA models. Also, there is a seasonal component available in the residuals at the lag 12 (represented by spikes at lag 12). This makes sense since we are analyzing monthly data that tends to have seasonality of 12 months because of patterns in tractor sales.

vii) *ARIMAfit = auto.arima(Log_data , approximation=FALSE,trace=FALSE)*
summary(ARIMAfit)

```
Series: Log_data
ARIMA(0,1,1)(0,1,1)[12]

Coefficients:
          ma1     sma1
      -0.3978  -0.5550
s.e.   0.0894   0.0729

sigma^2 estimated as 0.0002518:  log likelihood=355.75
AIC=-705.5   AICc=-705.31   BIC=-696.88

Training set error measures:
                     ME       RMSE        MAE        MPE      MAPE      MASE
Training set 0.0002955105 0.01501924 0.01127437 0.01048748 0.4425516 0.2167046
                    ACF1
Training set 0.01183187
```

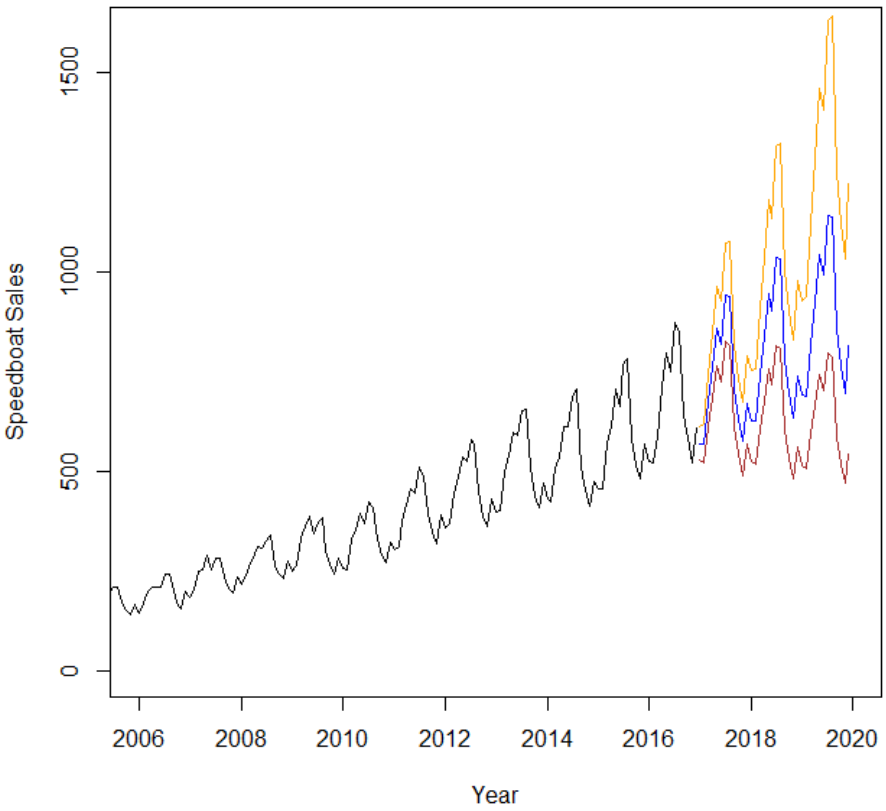
The best fit model is selected based on Akaike Information Criterion (AIC), and Bayesian Information Criterion (BIC) values. The idea model to choose who has minimum AIC and BIC values.

As expected, our model has I (or integrated) component equal to 1. This represents differencing of order 1. There is additional differencing of lag 12 in the above best fit model. Moreover, the best fit model has MA value of order 1. Also, there is seasonal MA with lag 12 of order 1.

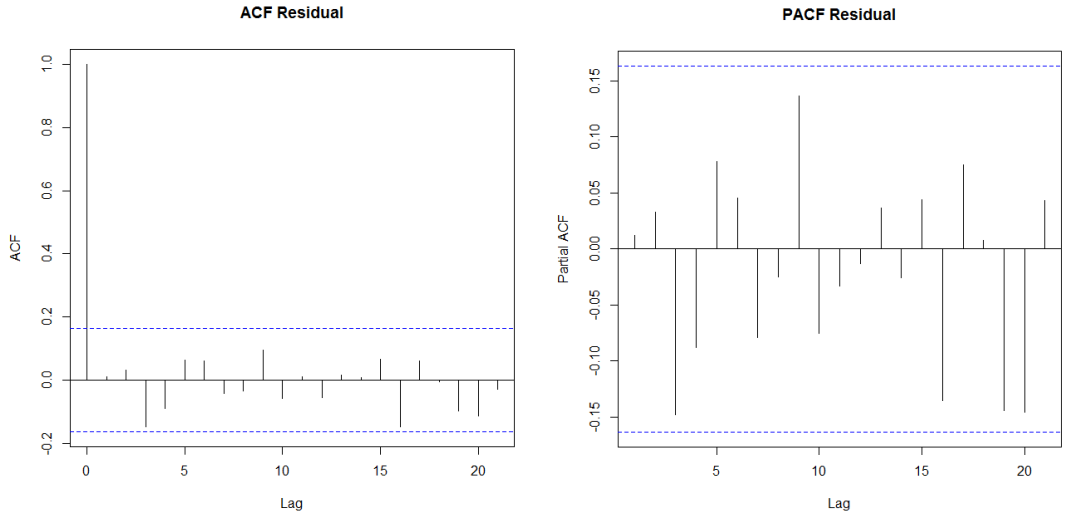
viii) *par(mfrow = c(1,1))*
pred = predict(ARIMAfit, n.ahead = 36)
pred
plot(data,type='l',xlim=c(2006,2020),ylim=c(1,1600),xlab = 'Year',ylab = 'Speedboat Sales')
lines(10^(pred$pred),col='blue')
lines(10^(pred$pred+2*pred$se),col='orange')
lines(10^(pred$pred-2*pred$se),col='Brown')

```
$`pred`
```

```
         Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep
2017 2.757190 2.755749 2.828882 2.882418 2.934635 2.914012 2.973922 2.972476 2.850108
2018 2.799138 2.797696 2.870829 2.924366 2.976582 2.955959 3.015869 3.014424 2.892055
2019 2.841085 2.839643 2.912776 2.966313 3.018530 2.997907 3.057816 3.056371 2.934002
         Oct      Nov      Dec
2017 2.800220 2.759985 2.827989
2018 2.842168 2.801933 2.869937
2019 2.884115 2.843880 2.911884

$se
          Jan        Feb        Mar        Apr        May        Jun        Jul
2017 0.01586846 0.01852403 0.02084398 0.02293039 0.02484220 0.02661703 0.02828070
2018 0.03914877 0.04151302 0.04374969 0.04587744 0.04791080 0.04986130 0.05173832
2019 0.06376506 0.06627392 0.06869122 0.07102629 0.07328700 0.07548003 0.07761111
          Aug        Sep        Oct        Nov        Dec
2017 0.02985180 0.03134424 0.03276878 0.03413392 0.03544653
2018 0.05354959 0.05530156 0.05699971 0.05864872 0.06025261
2019 0.07968523 0.08170671 0.08367937 0.08560658 0.08749136
```



ix) *par(mfrow=c(1,2))*
acf(ts(ARIMAfit$residuals),main='ACF Residual')
pacf(ts(ARIMAfit$residuals),main='PACF Residual')



Since there are no spikes outside the insignificant zone for both ACF and PACF plots we can conclude that residuals are random with no information or juice in them. Hence our ARIMA model is working fine.

[27 Marks]

Solution 5:

i) *#Create a matrix with the transition probabilities.*

```
Matrix2<-
matrix(c(0,0.5,0.5,0,0,0,0,1/4,0,1/4,0,1/4,0,1/4,1/4,1/4,0,1/4,0,1/4,0,0,0,0.5,0,0.5,0,0,0,1/3,0,1/3,0,
1/3,0,0,0,1/3,0,1/3,0,1/3,0,0.5,0,0,0,0.5,0),byrow=TRUE,nrow=7)
Matrix2

sum(Matrix2[1,])
sum(Matrix2[2,])
sum(Matrix2[3,])
sum(Matrix2[4,])
sum(Matrix2[5,])
sum(Matrix2[6,])
sum(Matrix2[7,])
```

```
     [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
[1,] 0.00 0.5000000 0.5000000 0.0000000 0.0000000 0.0000000 0.0000000
[2,] 0.25 0.0000000 0.2500000 0.0000000 0.2500000 0.0000000 0.2500000
[3,] 0.25 0.2500000 0.0000000 0.2500000 0.0000000 0.2500000 0.0000000
[4,] 0.00 0.0000000 0.5000000 0.0000000 0.5000000 0.0000000 0.0000000
[5,] 0.00 0.3333333 0.0000000 0.3333333 0.0000000 0.3333333 0.0000000
[6,] 0.00 0.0000000 0.3333333 0.0000000 0.3333333 0.0000000 0.3333333
[7,] 0.00 0.5000000 0.0000000 0.0000000 0.0000000 0.5000000 0.0000000
```

#Create a vector with state names

```
statesmatrix1<-c("A","B","C","D","E","F","G")
statesmatrix1
```

```
> statesmatrix1
[1] "A" "B" "C" "D" "E" "F" "G"
```

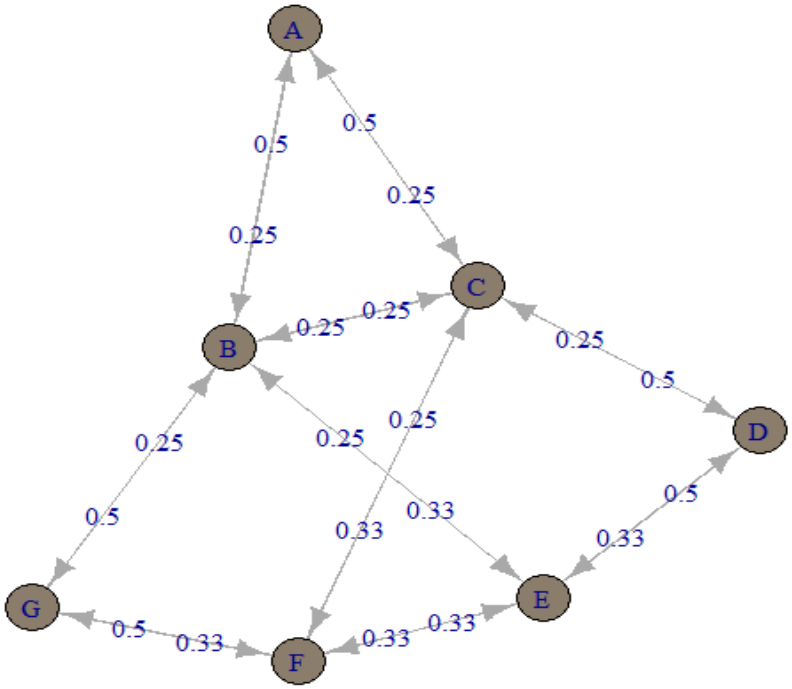
#Create the transition matrix

```
transitionmat1<-
  new("markovchain",states=statesmatrix1,transitionMatrix=Matrix2)
```

transitionmat1

markovchain::plot(transitionmat1)

```
> transitionmat1
Unnamed Markov chain
 A  7 - dimensional discrete Markov Chain defined by the following states:
 A, B, C, D, E, F, G
 The transition matrix  (by rows)  is defined as follows:
     A         B         C         D         E         F         G
A 0.00 0.5000000 0.5000000 0.0000000 0.0000000 0.0000000 0.0000000
B 0.25 0.0000000 0.2500000 0.0000000 0.2500000 0.0000000 0.2500000
C 0.25 0.2500000 0.0000000 0.2500000 0.0000000 0.2500000 0.0000000
D 0.00 0.0000000 0.5000000 0.0000000 0.5000000 0.0000000 0.0000000
E 0.00 0.3333333 0.0000000 0.3333333 0.0000000 0.3333333 0.0000000
F 0.00 0.0000000 0.3333333 0.0000000 0.3333333 0.0000000 0.3333333
G 0.00 0.5000000 0.0000000 0.0000000 0.0000000 0.5000000 0.0000000
```



ii) *# What are the absorbing states?*
markovchain::absorbingStates(transitionmat1)

 # Find the steady state or stationary distribution.
markovchain::steadyStates(transitionmat1)

```
> markovchain::steadyStates(transitionmat1)
      A   B   C   D    E    F   G
[1,] 0.1 0.2 0.2 0.1 0.15 0.15 0.1
```

iii) *# Time to reach steady state if particle starts from A*

```
n=0
starting=c(1,0,0,0,0,0,0)
while(starting%*%Matrix2!=starting){
 starting=starting%*%Matrix2
 print(starting)
 n=n+1
 print(n)
}
```

```
 [1] 54
     [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,]  0.1  0.2  0.2  0.1 0.15 0.15  0.1
[1] 55
There were 50 or more warnings (use warnings() to see the first 50)
```

 # Time to reach steady state if particle starts from B

```
m=0
starting=c(0,1,0,0,0,0,0)
while(starting%*%Matrix2!=starting){
 starting=starting%*%Matrix2
 print(starting)
 m=m+1
 print(m)
}
```

```
[1,]  0.1 0.2007407 0.1992593 0.1004772 0.1492593 0.1507407 0.09952283
[1] 54
There were 50 or more warnings (use warnings() to see the first 50)
```

 # Time to reach steady state if particle starts from C

```
p=0
starting=c(0,0,1,0,0,0,0)
while(starting%*%Matrix2!=starting){
 starting=starting%*%Matrix2
 print(starting)
 p=p+1
 print(p))
}
```

```
[1,]  0.1 0.1993928 0.2006072 0.09960879 0.1506072 0.1493928 0.1003912
[1] 56
There were 50 or more warnings (use warnings() to see the first 50)
```

[17 Marks]

Solution 6:

i) *#Define a function for S(x).*

```
Sx<-function(x){
 0.1*(100-x)^(1/2)}
```

#New born surviving till age 65.

```
Sx(65)
```

```
> Sx(65)
[1] 0.591608
```

```
#10 year old surviving till 95
z<-Sx(95)/Sx(10)
z
```

```
> z
[1] 0.2357023
```

```
#New born dying between the age of 65-70 years
x<-Sx(65)-Sx(70)
x
```

```
> x
[1] 0.04388542
```

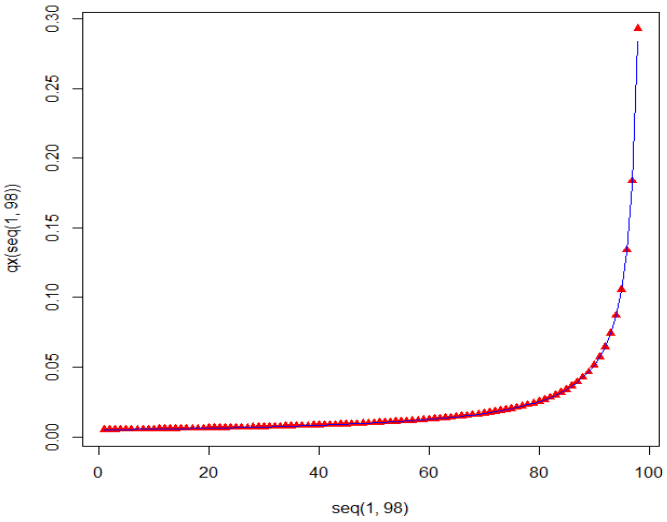
ii)
```
# Calculate qx for each age
age<-c(1,seq(5,95,by=5),seq(96,98,by=1))
qx<-function(x){
  1-(Sx(x+1)/Sx(x))}

mortality.table<-data.frame(age)
mortality.table$q.x<-round(qx(mortality.table$age),10)
mortality.table
```

age	q.x
1	1 0.005063324
2	5 0.005277082
3	10 0.005571074
4	15 0.005899756
5	20 0.006269654
6	25 0.006689038
7	30 0.007168551
8	35 0.007722123
9	40 0.008368348
10	45 0.009132611
11	50 0.010050506
12	55 0.011173535
13	60 0.012579117
14	65 0.014389239
15	70 0.016807920
16	75 0.020204103
17	80 0.025320565
18	85 0.033908217
19	90 0.051316702
20	95 0.105572809
21	96 0.133974596
22	97 0.183503419
23	98 0.292893219

iii)
```
#Define a function for Mu(x).
mu<-function(x){
  1/(2*(100-x))}

# Calculate qx for each age using approximation
qx1<-function(x){
  1-exp(-mu(x+1/2))}

mortality.table$q.x1<-round(qx1(mortality.table$age),10)
mortality.table
```

age	q.x	q.x1
1	1	0.005063324	0.005063280
2	5	0.005277082	0.005277033
3	10	0.005571074	0.005571016
4	15	0.005899756	0.005899688
5	20	0.006269654	0.006269572
6	25	0.006689038	0.006688938
7	30	0.007168551	0.007168428
8	35	0.007722123	0.007721969
9	40	0.008368348	0.008368152
10	45	0.009132611	0.009132356
11	50	0.010050506	0.010050166
12	55	0.011173535	0.011173067
13	60	0.012579117	0.012578449
14	65	0.014389239	0.014388239
15	70	0.016807920	0.016806324
16	75	0.020204103	0.020201326
17	80	0.025320565	0.025315086
18	85	0.033908217	0.033895003
19	90	0.051316702	0.051270520
20	95	0.105572809	0.105160683
21	96	0.133974596	0.133122100
22	97	0.183503419	0.181269247
23	98	0.292893219	0.283468689

iv) *# Ploting qx and qx1 using the line graph.*
plot(seq(1,98),qx(seq(1,98)),pch=17,col="Red")+lines(seq(1,98),qx1(seq(1,98)),col="Blue")



v) *# Commentary on the graph.*
Below is the relationship between the survival function $S_0(x)$ and hazard function μ_x:
$S_0(x)$ = exp[-integral($\mu_x(t)$ dt] .....(1)

The survival function $S_0(x)$ above is defined as:

$$S_0(x) = \frac{1}{10}(100 - x)^{1/2}, \quad \text{for} \quad 0 \le x \le 100.$$

$\delta/\delta\tau$ [log ($S_0(x)$] = μ_x = *1 / 2(100-x)*
Deriving μ_x using the above relationship in (1), we get the same function for μ_x as specified in part (c) and hence creating mortality table using either survival function or hazard function shall give the same result.
There are minor differences in the graph of *qx* and that is purely due to the approximation assumed in part (c)

[25 Marks]

****