Institute of Actuaries of India

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

November 2023 Examination

INDICATIVE SOLUTION

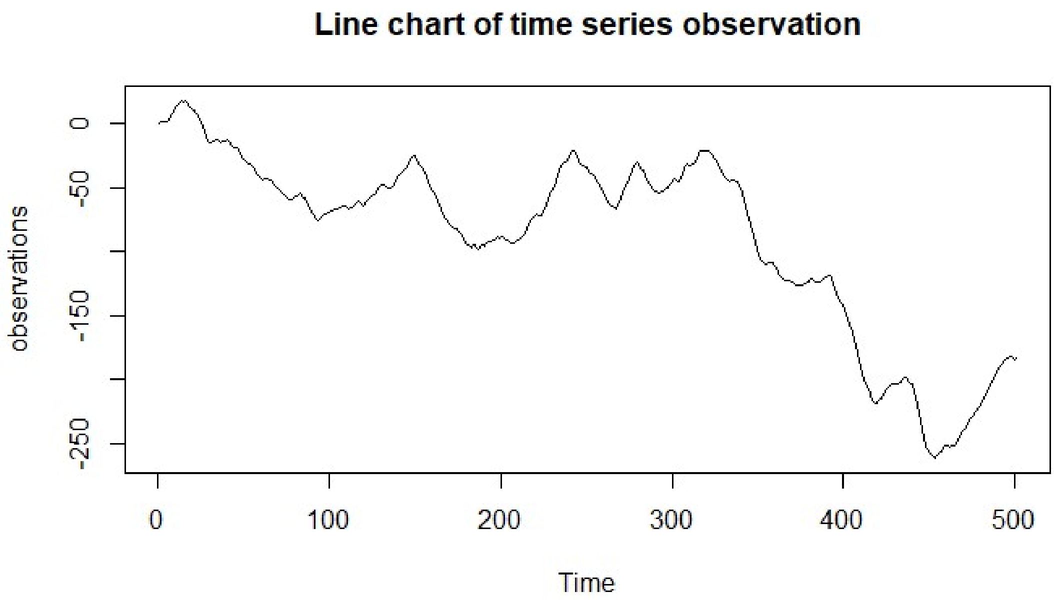
Introduction

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)

```
set.seed(100)

observations <- arima.sim(list(order = c(1,1,1), ar = 0.9, ma = 0.2), n = 50
0)

plot(observations, main = "Line chart of time series observation")
```

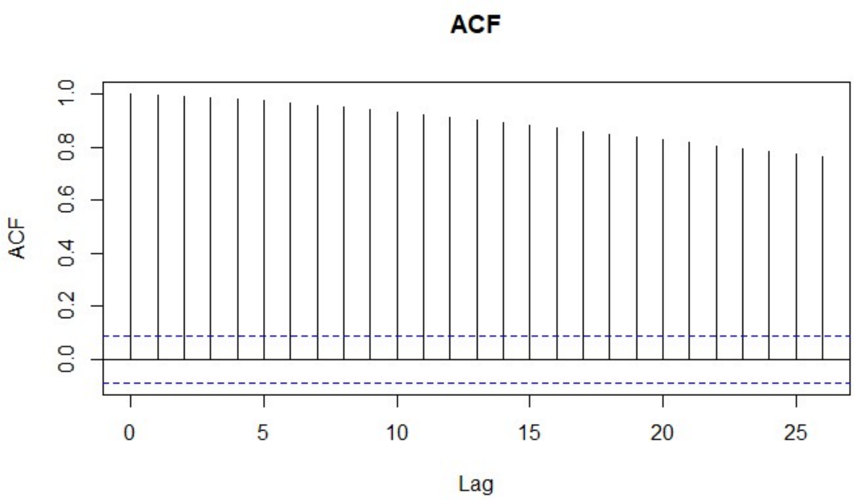


[3]

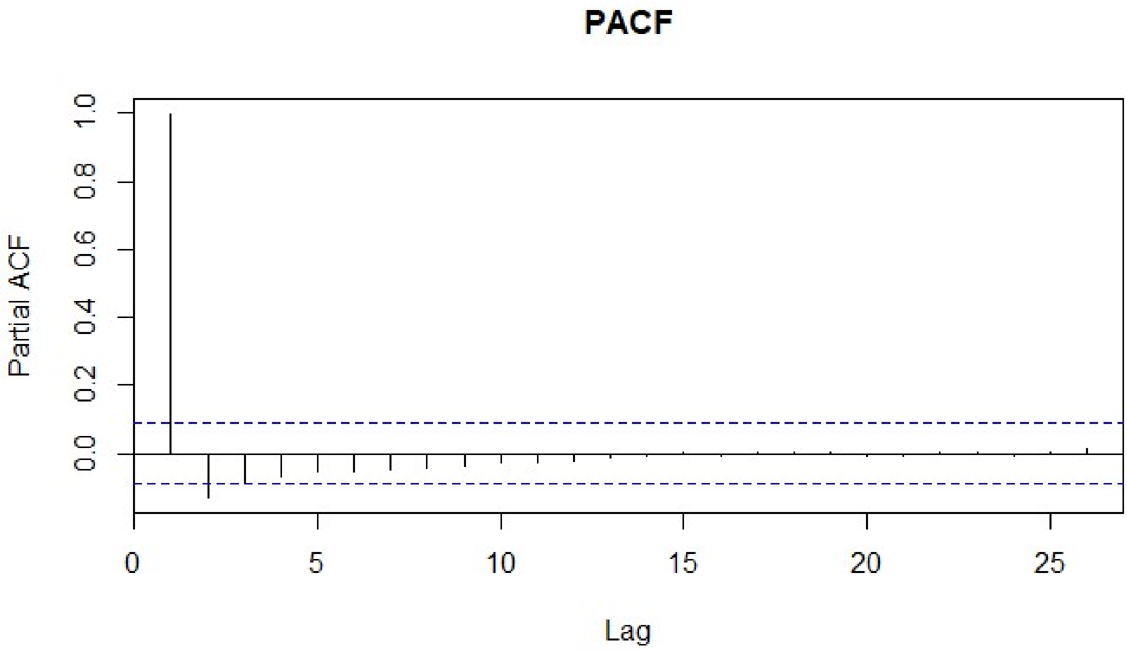
ii)
- The data is not stationary as we observe that the values are changing with time
- Downward trend is observed in the data and an upward trend towards the end, which indicates the data being non stationary
- Mean and Standard Deviation are different at different points in time , mean is not constant. [2]

iii)

```
acf(observations, main = "ACF")
```



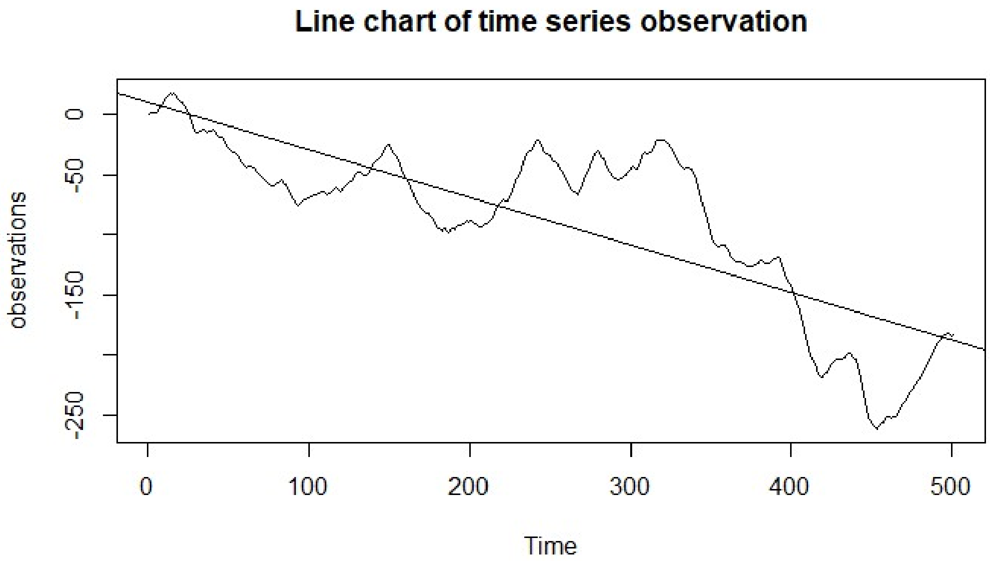

```
pacf(observations, main = "PACF")
```



[2]

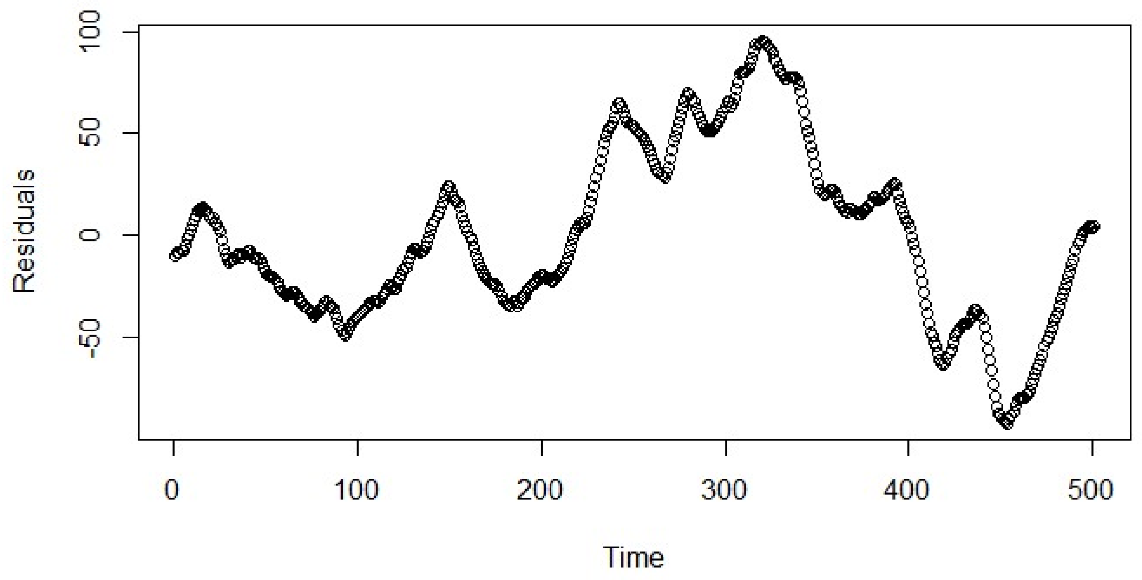
iv)

```
x = 1:501

leastsquarefit <- lm(observations~x)

leastsquarefit$coefficients

(Intercept)           x
 10.0453231  -0.3941461


plot(observations, main = "Line chart of time series observation")
abline(leastsquarefit)
```



[3]

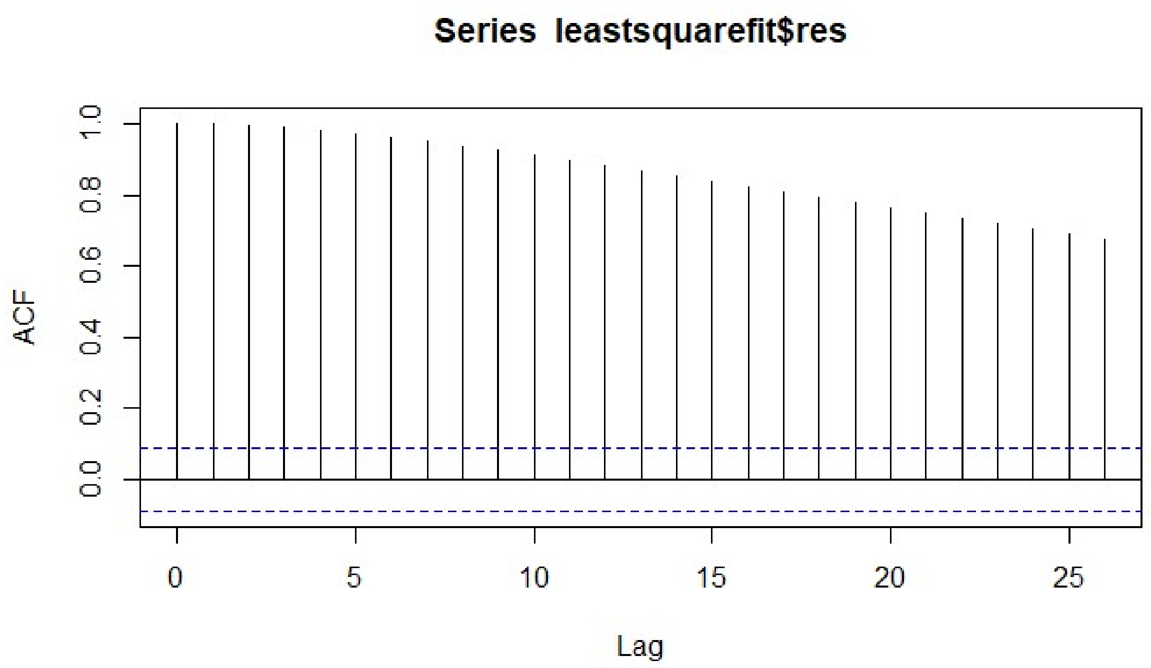
v)
```
plot(leastsquarefit$res, xlab = "Time" , ylab = "Residuals")
```



It is clear that residuals are not stationary as they are negative in the first, then followed by positive residuals in the middle part and then negative in the last part.

Alternate Solution

```
acf(leastsquarefit$res)
```



The residuals are not stationary as the ACF values are decaying very slowly. [2]

vi)
```
fit1 = arima(observations, order= c(1,0,0))
fit1

Call:
arima(x = observations, order = c(1, 0, 0))
```

```
Coefficients:
         ar1  intercept
      0.9997   -89.3179
s.e.  0.0004    85.5509

sigma^2 estimated as 5.091:  log likelihood = -1122.25,  aic = 2250.5

fit2 = arima(observations, order= c(3,0,0))
fit2

Call:
arima(x = observations, order = c(3, 0, 0))

Coefficients:
         ar1      ar2     ar3  intercept
      2.0064  -1.1350  0.1278   -89.0734
s.e.  0.0443   0.0864  0.0444    46.5760

sigma^2 estimated as 1.01:  log likelihood = -718.61,  aic = 1447.22

fit3 = arima(observations, order= c(1,0,1))
fit4

Call:
arima(x = observations, order = c(1, 0, 1))

Coefficients:
         ar1     ma1  intercept
      0.9996  0.7731   -89.3280
s.e.  0.0006  0.0210    83.2395

sigma^2 estimated as 2.128:  log likelihood = -904.58,  aic = 1817.16
```

[3]

vii)

```
fit1$coef[1] - qnorm(0.975)*sqrt(fit1$var.coef[1,1])
     ar1
0.998827

fit1$coef[1] + qnorm(0.975)*sqrt(fit1$var.coef[1,1])
     ar1
1.000529
```

The confidence interval is (0.998827 , 1.000529) [2]

viii) The AIC is lowest for AR(3) among the models above and hence is the best fit among the above models.

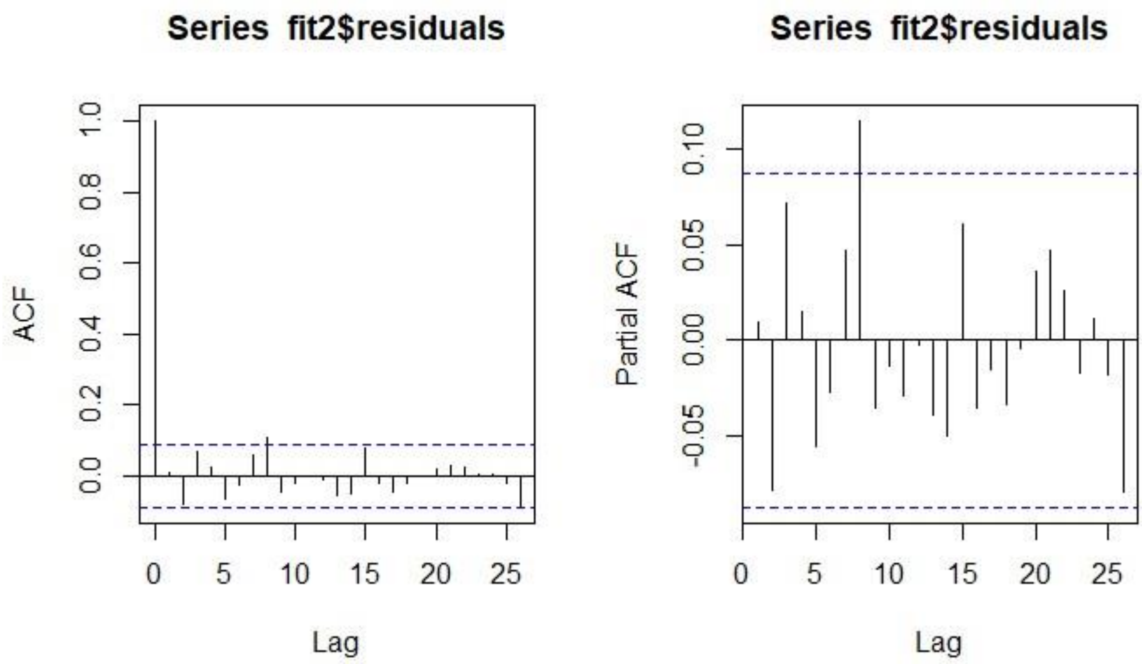
```
predict(fit2, n.ahead = 10)

$pred
```

```
Time Series:
Start = 502
End = 511
Frequency = 1
 [1] -181.5409 -180.3970 -179.3290 -178.3314 -177.3959 -176.5145 -175.6804 -
174.8877 -174.1311
[10] -173.4062

$se
Time Series:
Start = 502
End = 511
Frequency = 1
 [1]  1.005205  2.253488  3.677225  5.195037  6.758213  8.335575  9.906391 1
1.456656 12.976922
[10] 14.460927
```

[2]

ix)

```
par(mfrow = c(1,2))
acf(fit2$residuals)
pacf(fit2$residuals)
```



[2]

x) PACF for the data shows no significance from lag 2 which could indicate stationarity but the ACF is decaying very slowly indicating it is not stationary. For example, this is consistent with ARIMA (1,1,1) behaviour.

The plot for the residuals generally lies within the confidence intervals. This is consistent with the residuals forming the white noise process. [2]

xi)

```
Box.test(fit2$residuals, type = "Ljung", fitdf = 3, lag = 4)

Box-Ljung test

data:  fit2$residuals
X-squared = 5.8414, df = 1, p-value = 0.01565
```

```
Box.test(fit2$residuals, type = "Ljung", fitdf = 3, lag = 6)

Box-Ljung test

data:  fit2$residuals
X-squared = 8.3718, df = 3, p-value = 0.03892

Box.test(fit2$residuals, type = "Ljung", fitdf = 3, lag = 12)

Box-Ljung test

data:  fit2$residuals
X-squared = 17.565, df = 9, p-value = 0.04057
```

- The result above suggests that the residuals are forming a white noise process suggesting a good fit for ARIMA(3,0,0) model,
- The result above also suggests that the model requires differencing as it is consistent with ARIMA(p,1,q) behaviour.
- However, the three tests are not consistent with an ARIMA(3,0,0) model at the 5% significance level, since the p-values are lesser than 0.05
- Thus, there is not enough evidence to conclude ARIMA(3,0,0) to be a good fit.
- Also, we would expect the ARIMA(1,1,1) model that was used to generate the data to satisfy this test as well and thus can be shown to be also a good fit.

[5]

[28 Marks]

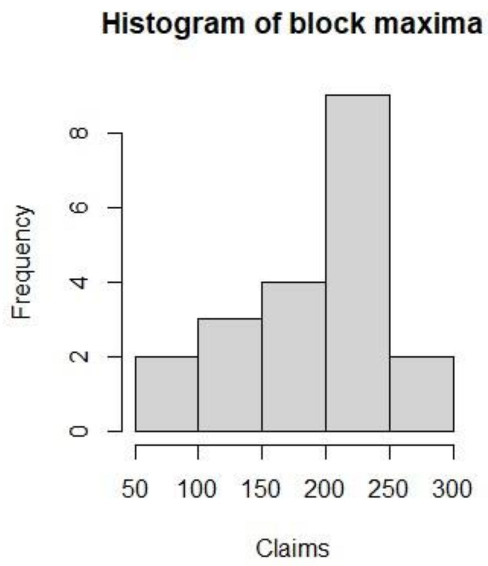
Solution 2:

i) <code for reading the input data file>

```
e.g. Claims <- read.csv(‘path’/Claims.csv")

Claims$block <-(Claims$Claim_number-1) %/% 5 +1

blockmax <- aggregate(Claims ~ block, Claims, max)

blockmax
   block Claims
1      1    104
2      2     94
3      3    218
4      4    235
5      5    140
6      6     84
7      7    213
8      8    222
9      9    128
10    10    247
11    11    152
12    12    202
13    13    193
14    14    201
15    15    180
16    16    291
17    17    243
18    18    163
19    19    267
20    20    203
```

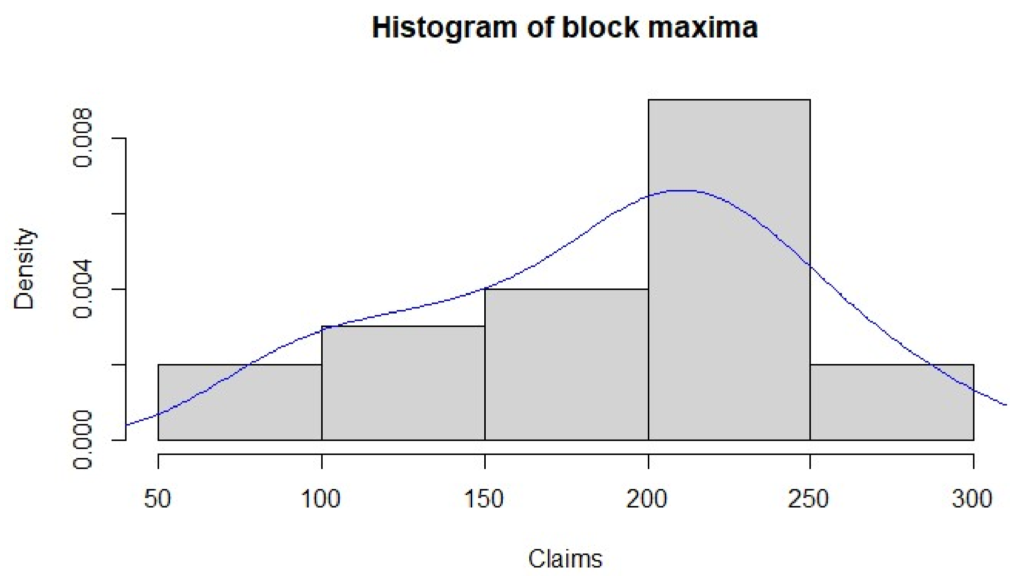
[2]

ii)
```
hist(blockmax$Claims, xlab = "Claims", main = "Histogram of block maxima")
```



[2]

iii)
```
hist(blockmax$Claims, xlab = "Claims", main = "Histogram of block maxima", f
req = FALSE)
lines(density(blockmax$Claims), col = "blue")
```



[2]

iv)
```
library(MASS)

est <- fitdistr(blockmax$Claims, "weibull", lower = 0)
est
      shape         scale
    3.8973242   209.3505841
 (  0.7069667) ( 12.6163461)

c = est$estimate["scale"]^ (-est$estimate["shape"])
> c
       scale
9.011579e-10

> g = est$estimate["shape"]
```

```
> g
   shape
3.897324
```

[3]

v)

```
alpha = mean(blockmax$Claims)
alpha
[1] 189
beta = sd(blockmax$Claims)
beta
[1] 57.78271
gamma = skewness(blockmax$Claims)
gamma
[1] -0.273219
```

[3]

vi)

```
MLE = function(x){f <- 1/x[2]*(1+x[3]*(blockmax$Claims - x[1])/x[2])^(-1-1/x
[3])*exp(-(1+x[3]*(blockmax$Claims- x[1])/x[2])^(-1/x[3]))
 lnf <- log(f)
 sum(-lnf)
 }
```

[2]

vii)

```
p = c(alpha,beta,gamma)

MLE(p)
[1] 110.3667

f_MLE <- nlm(MLE,p)
f_MLE
$minimum
[1] 108.4947

$estimate
[1] 173.0482986  59.5496685  -0.4257482

$gradient
[1] -2.207406e-07 -7.863145e-07 -5.439915e-05

$code
[1] 1

$iterations
[1] 33
```

[2]

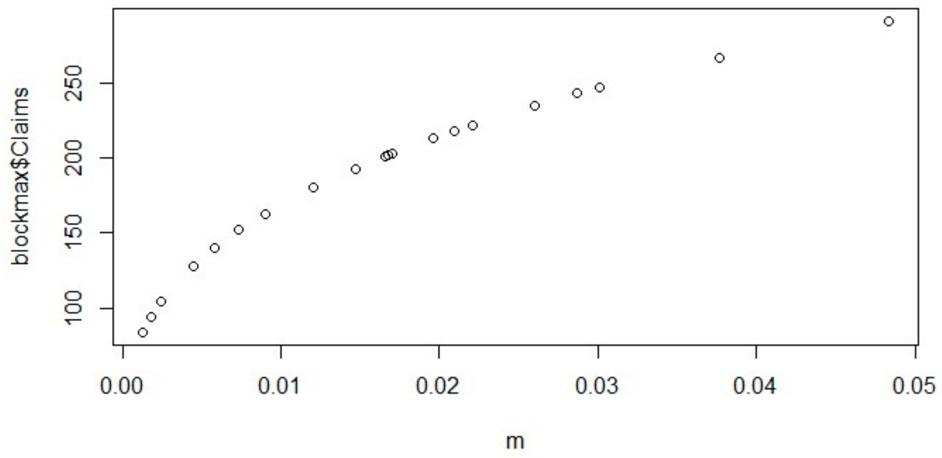
viii)

```
GEV <- function(x,a,b,c){f = 1/b * (1+c*(x-a)/b)^-(1+1/c)*exp(-((1+c*(x-a)/b
)^(-1/c)))}


fit = GEV(blockmax$Claims,f_MLE$estimate[1],f_MLE$estimate[2],f_MLE$estimate
[3])

fit
 [1] 0.002216593 0.001752716 0.006657504 0.005922606 0.004311508 0.001356100
0.006776331
 [8] 0.006529717 0.003563970 0.005131517 0.005053503 0.006876667 0.006798624
0.006874941
[15] 0.006456144 0.001361035 0.005416809 0.005683342 0.003474704 0.006876611
```

[2]

ix)

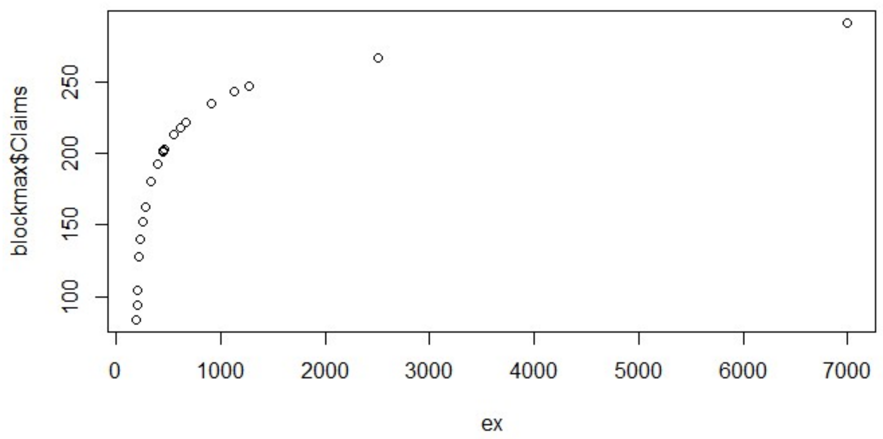
```
h = dweibull(blockmax$Claims,g,est$estimate["scale"])/(1-pweibull(blockmax$C
laims,g,est$estimate["scale"]))
h
 [1] 0.002452262 0.001829610 0.020933123 0.026020746 0.005802246 0.001320774
0.019572126
 [8] 0.022065445 0.004475462 0.030059810 0.007363354 0.016784861 0.014708485
0.016545242
[15] 0.012017747 0.048335279 0.028670960 0.009015548 0.037666671 0.017026742

plot(m,blockmax$Claims)
```



The hazard function is an increasing function of x. An increasing hazard function indicates lighter tail.

[4]

x)
```
Sy = function(y,g,b){(1-pweibull(y,g,b))}
int = integrate(Sy,0, Inf, g, est$estimate["scale"])
ex = int$value/Sy(blockmax$Claims,g,est$estimate["scale"])

plot(ex,blockmax$Claims)
```



The mean residual life function is an increasing function of x. An increasing mean residual function indicates a lighter tail.

[5]

[27 Marks]

Solution 3:

i) <code for reading the input data file>
```
e.g. Std_Table <- read.csv('path'/Std_Table.csv")

Std_Table$Exmux_1 = Std_Table$Exposure * Std_Table$Graduation_1

Std_Table$Exmux_2 = Std_Table$Exposure * Std_Table$Graduation_2

Std_Table$zx_1 = (Std_Table$Deaths - Std_Table$Exmux_1)/(sqrt(Std_Table$Exmux_1))

Std_Table$zx_2 = (Std_Table$Deaths - Std_Table$Exmux_2)/(sqrt(Std_Table$Exmux_2))
```

```
head(Std_Table, 10)
# A tibble: 10 x 9
     Age Exposure Deaths Graduation_1 Graduation_2 Exmux_1 Exmux_2   zx_1    zx_2
   <dbl>    <dbl>  <dbl>        <dbl>        <dbl>   <dbl>   <dbl>  <dbl>   <dbl>
 1    30    70000     39     0.000388     0.000555    27.2    38.8  2.27   0.0241
 2    31    69747     43     0.000429     0.000623    29.9    43.5  2.39  -0.0686
 3    32    68140     34     0.000474     0.000488    32.3    33.3  0.299  0.130
 4    33    68744     31     0.000524     0.000432    36.0    29.7 -0.837  0.239
 5    34    66852     23     0.000579     0.000486    38.7    32.5 -2.52  -1.66
 6    35    69230     50     0.00064      0.000596    44.3    41.3  0.855  1.36
 7    36    61580     48     0.000708     0.000685    43.6    42.2  0.667  0.896
 8    37    67582     43     0.000782     0.000713    52.8    48.2 -1.35  -0.747
 9    38    68363     48     0.000865     0.000709    59.1    48.5 -1.45  -0.0674
10    39    65914     47     0.000956     0.000733    63.0    48.3 -2.02  -0.189
```

[3]

ii)

```
diff_1 = data.frame(grad_1 = diff(Std_Table$Graduation_1),grad_2 = diff(Std_Table$Graduation_2))

diff_2 = data.frame(grad_1 = diff(diff_1$grad_1),grad_2 = diff(diff_1$grad_2))

diff_3 = data.frame(grad_1 = diff(diff_2$grad_1),grad_2 = diff(diff_2$grad_2))

head(diff_3, 10)
          grad_1      grad_2
1   1.000000e-06    0.000282
2  -2.710505e-19    0.000031
3   1.000000e-06   -0.000054
4   1.000000e-06   -0.000077
5  -1.000000e-06   -0.000040
6   3.000000e-06    0.000029
7  -1.000000e-06    0.000060
8   1.000000e-06    0.000046
9   3.000000e-06    0.000012
10 -1.000000e-06   -0.000026
```

The third differences are larger for Graduation 2 than for Graduation 1 and they progress in less regular manner and hence Graduation 2 is not as smooth as Graduation 1.

[3]

iii)

```
chisq = vector(length = 2)

chisq[1] = sum(Std_Table$zx_1^2)

chisq[2] = sum(Std_Table$zx_2^2)

df = c(46,47)

1 - pchisq(chisq, df = df)

[1] 1.332268e-15 0.000000e+00
```

The *p*-value for graduation 1 is 1.332268e-15
The *p*-value for graduation 2 is 0.000000e+00

Graduation-2 is overfitted as observed the respective p-value. [3]

iv)

```
positive = vector(length = 2)
> negative = vector(length = 2)
> positive[1] = length(Std_Table$zx_1[Std_Table$zx_1 > 0])
> positive[2] = length(Std_Table$zx_2[Std_Table$zx_2 > 0])
> negative[1] = length(Std_Table$zx_1[Std_Table$zx_1 < 0])
> negative[2] = length(Std_Table$zx_2[Std_Table$zx_2 < 0])
> positive
[1] 38 30
> negative
[1] 23 31
```

[3]

v) For Graduation 1 we have more positive values

So p value is
2 * P(P>=38) = 2* [1 – P(P <=37)]

```
2 * (1 - pbinom(37, size = 61, prob = 0.5))
[1] 0.07217744
```

For Graduation 2 we have more negative values
So p value is
2 * P(P<=30)

```
2 * pbinom(30, size = 61, prob = 0.5)
[1] 1
```

[3]

vi)

```
groups = vector(length = 2)

for(j in 1:2){positive_z = (Std_Table[, j+7]>0)*1
+ groups[j] = sum(duplicated(c(which(positive_z == 1) - 1, which(positive_z =
= 0 )))*1)
+ positive_z[1]*1}

groups
[1] 12 14
```

[3]

vii)

```
pvalue = vector(length = 2)

for (j in 1:2){pvalue[j]=0
+ for (k in 1 : groups[j]){pvalue[j]= pvalue[j]+choose(positive[j]-1,k-1)* ch
oose(negative[j]+1,k)/choose(positive[j]+negative[j], positive[j])}}

pvalue
[1] 0.09281982 0.26299014
```

[3]

viii)

```
scf = vector(length = 2)
m = length(Std_Table$Age)

for (j in 1:2) {scf[j] = (cor(Std_Table[1:m-1,j+7],Std_Table[2:m,j+7])*1)*sqr
t(m)}

scf
[1]  1.2153759212 -0.0003617506
```

For Graduation 1 , p value is less than 1.6449, the upper 5% point of standard normal distribution so there is no evidence of grouping of deviations of the same sign.
For Graduation 2, the p – value is negative and close to 0, indicating nearby values of Zx tend to have opposite values. [3]

ix)

```
cdt = vector(length = 2)

for (j in 1:2) {cdt[j] = (sum(Std_Table$Deaths) - sum(Std_Table[,j+5]))/sqrt(
sum(Std_Table[,j+5]))}
> cdt
[1]   3.767196 -30.221501
```

Graduation 1 p value is higher than 2.5% points of N(0,1) i.e. 1.96, there is sufficient evidence to reject null hypothesis. Therefore, there is bias in the Graduated rates 1.
Graduation 2 has high magnitude of negative test statistic that means, the bias in graduated rates is too high. [3]

x) Based on above tests,

- Graduation 1 is smoother than Graduation 2
- Both the graduation passes the goodness of fit, but Graduation 2 seems to be overfitted

- Signs test indicates slightly higher positive signs for Graduation 1 as compared to Graduation 2, however p value for both the Graduation passes the test and thus the rates are not biased.
- Grouping of signs test & Serial correlation test shows no evidence of grouping of deviations of the same sign.
- Both the graduation has biasedness as having large positive or negative deviation. However, the biasedness seems to be too high for Graduation 2.
- Thus, both the graduation are good fit, however, Graduation 2 is slightly overfitted and less smooth and thus I would suggest Graduation 1 to be published. [3]

[30 Marks]

Solution 4:

i)

```
> Employment <- c("Marketing","Admin","Training")
> Employment
[1] "Marketing" "Admin"     "Training"
```

[1]

ii)

```
> M2A<-function(x){0.0025*x}
> M2T<-function(x){0.0075*x}
> A2M<-function(x){0.003*x}
> A2T<-function(x){0.004*x}
> T2M<-function(x){0.001*x}
> T2A<-function(x){0.003*x}
> EmploymentTransition<-function(x){
+   M<-matrix(0,nrow=3,ncol=3)
+   M[1,1]<-1-M2A(x)-M2T(x)
+   M[1,2]<-M2A(x)
+   M[1,3]<-M2T(x)
+   M[2,1]<-A2M(x)
+   M[2,2]<-1-A2M(x)-A2T(x)
+   M[2,3]<-A2T(x)
+   M[3,1]<-T2M(x)
+   M[3,2]<-T2A(x)
+   M[3,3]<-1-T2M(x)-T2A(x)
+   M
+ }

> x<-30
> Employmentchange_age30<-EmploymentTransition(x)
> Employmentchange_age30
     [,1]  [,2]  [,3]
[1,] 0.70 0.075 0.225
[2,] 0.09 0.790 0.120
[3,] 0.03 0.090 0.880

> y<-40
> Employmentchange_age40<-EmploymentTransition(y)
> Employmentchange_age40
     [,1] [,2] [,3]
[1,] 0.60 0.10 0.30
[2,] 0.12 0.72 0.16
[3,] 0.04 0.12 0.84
```

[3]

iii)

```
> install.packages("markovchain")
> library(markovchain)
> MCobject_age30<-new("markovchain",states=Employment,byrow=T,transitionMatri
x=Employmentchange_age30,name="Markovchain_age30")

> MCobject_age30

Markovchain_age30
 A  3 - dimensional discrete Markov Chain defined by the following states:
 Marketing, Admin, Training
 The transition matrix  (by rows)  is defined as follows:
          Marketing Admin Training
Marketing      0.70 0.075    0.225
Admin          0.09 0.790    0.120
Training       0.03 0.090    0.880
```

```
> MCobject_age40<-new("markovchain",states=Employment,byrow=T,transitionMatri
x=Employmentchange_age40,name="Markovchain_age40")
> MCobject_age40

Markovchain_age40
 A  3 - dimensional discrete Markov Chain defined by the following states:
 Marketing, Admin, Training
 The transition matrix  (by rows)  is defined as follows:
          Marketing Admin Training
Marketing      0.60  0.10     0.30
Admin          0.12  0.72     0.16
Training       0.04  0.12     0.84
```

[3]

iv)

a)

```
> n<-30
> B<-c(1,0,0)
> for(i in 1:3){B=B%*%EmploymentTransition(n+i-1)}
> B
          [,1]      [,2]     [,3]
[1,] 0.3625932 0.1791008 0.458306
```

So the required probability in 3 years is 45.8306%

b)

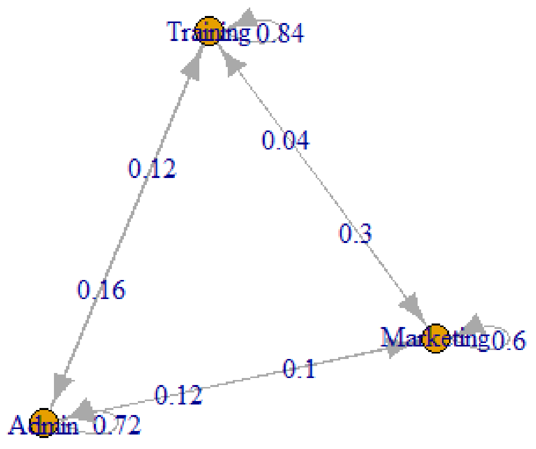
```
> n<-40
> B<-c(1,0,0)
> for(i in 1:5){B=B%*%EmploymentTransition(n+i-1)}
> B
          [,1]      [,2]      [,3]
[1,] 0.1681993 0.2658399 0.5659607
```

So the required probability in 5 years is 56.59607%.

[2]

v)

```
> plot(MCobject_age40)
```



[1]

vi)

```
> set.seed(250)
```

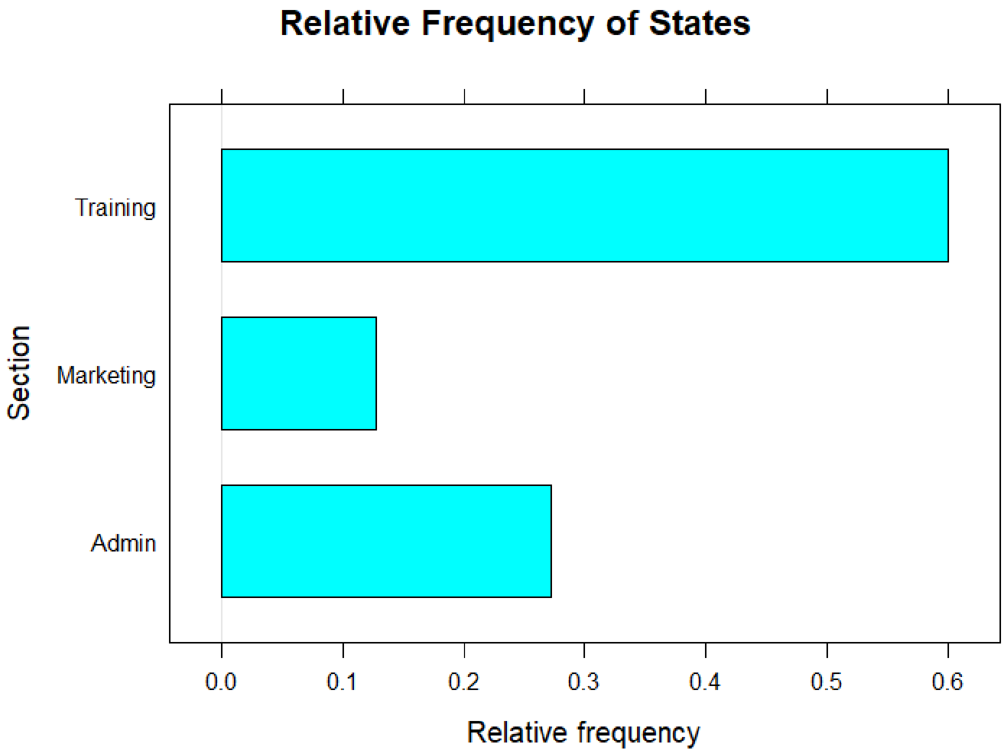
```
> seq_age30<-markovchainSequence(250,MCobject_age30)
> #frequeny of the terms person aged 30
> table(seq_age30)
Seq_age30
    Admin Marketing  Training
       68        32       150
```

```
> seq_age40<-markovchainSequence(250,MCobject_age40)
> #frequeny of the terms person aged 40
> table(seq_age40)
seq_age40
    Admin Marketing  Training
       67        34       149
```

[3]

vii)

```
> library(lattice)
> barchart(prop.table(table(seq_age30)),xlab="Relative frequency", ylab="Sect
ion",main="Relative Frequency of States")
```



[2]

[15 Marks]

****