Institute of Actuaries of India

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

September 2021 Examination

INDICATIVE SOLUTION

Introduction

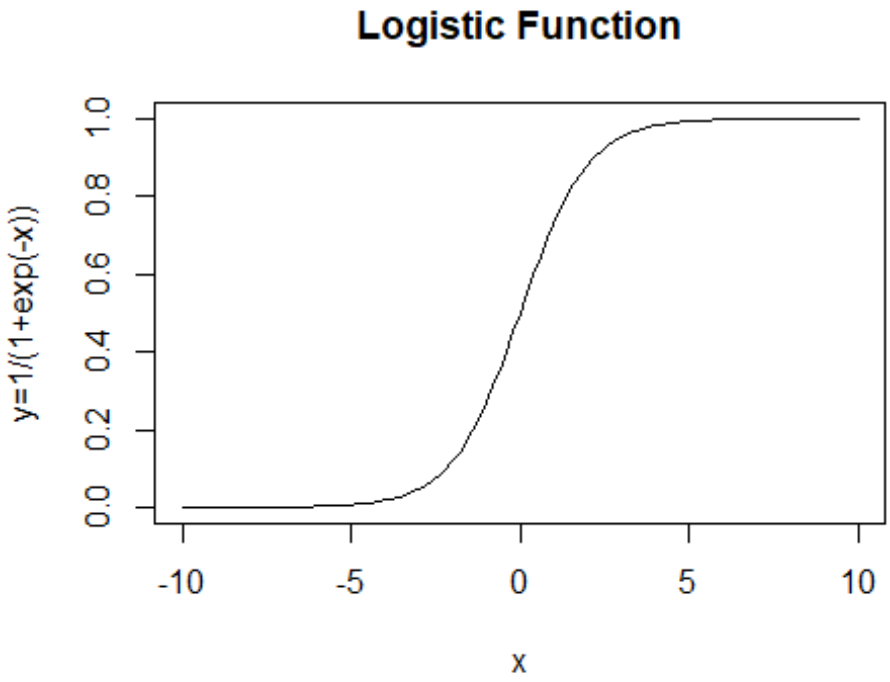
The indicative solution has been written by the Examiners with the aim of helping candidates. The solutions given are only indicative. It is realized that there could be other points as valid answers and examiner have given credit for any alternative approach or interpretation which they consider to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

Solution 1:

```
# (i)
logistic<-function(x) 1/(1+exp(-x))
```

[3M]

```
# (ii)
plot(logistic,xlim = c(-10,10), main = "Logistic Function", ylab = "y=1/(1+ex
p(-x))")
```



[5M]

[8 Marks]

Solution 2:

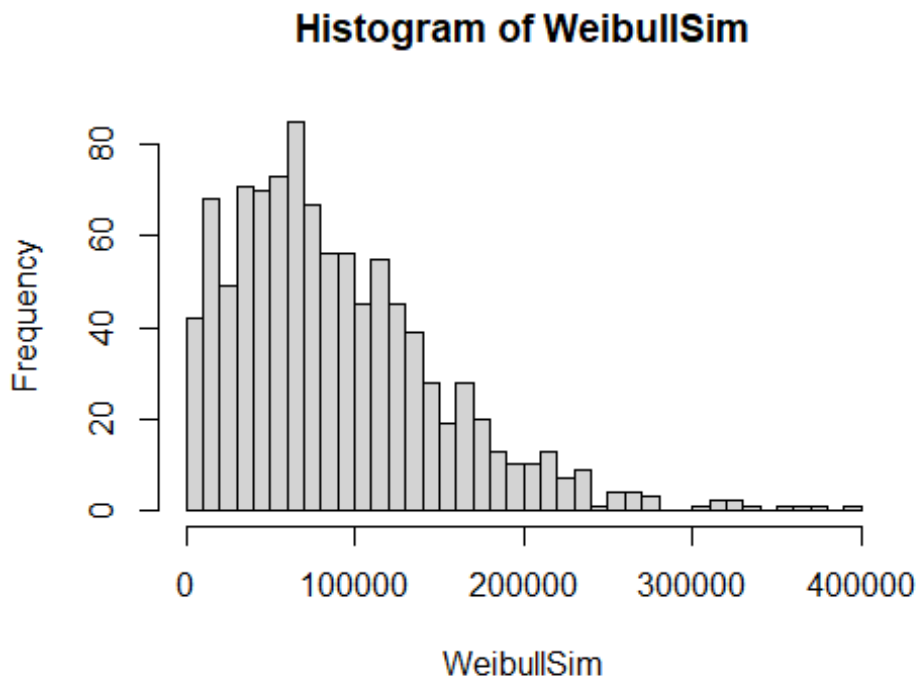
```
set.seed(1234)
options(scipen=5)

# (i)
shape<-1.5
scale<-100000
WeibullSim<-rweibull(1000,shape,scale)
```

[3M]

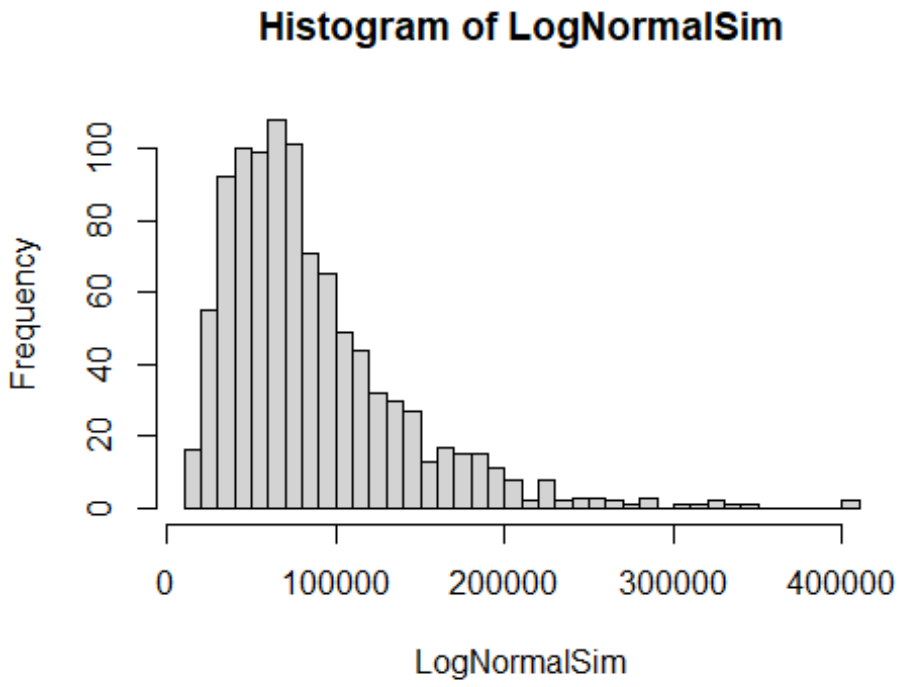
```
# (ii)
SimMean<-mean(WeibullSim)
SimMean

## [1] 89244.03

SimSD<-sd(WeibullSim)
SimSD
```

[2M]

```
## [1] 62717.89

# (iii)
mu<-log(SimMean/(1+SimSD^2/SimMean^2)^0.5)
mu

## [1] 11.19844

sigma<-log(1+SimSD^2/SimMean^2)^0.5
sigma

## [1] 0.6335449                                                    [5M]

# (iv)
LogNormalSim<-rlnorm(1000,mu,sigma)
mean(LogNormalSim)

## [1] 86421.17

sd(LogNormalSim)

## [1] 54384.16                                                     [5M]

# (v)
SimResults<-as.data.frame(cbind(WeibullSim,LogNormalSim))
hist(WeibullSim, breaks = 30)
```

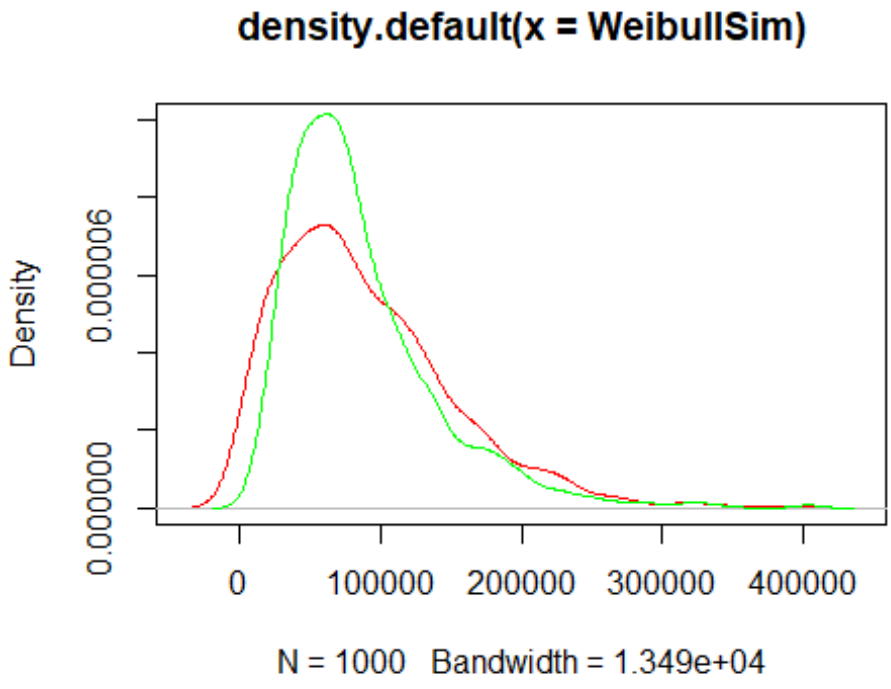



```
hist(LogNormalSim, breaks = 30)                                  [3M]
```




```r
# (vi)
plot(density(WeibullSim),type = "l",col="red",ylim = c(0,0.00001))
lines(density(LogNormalSim),col="green")
```

[3M]



[21 Marks]

Solution 3:

```
library(markovchain)

## Warning: package 'markovchain' was built under R version 4.0.5

## Package:  markovchain
## Version:  0.8.5-4
## Date:     2021-01-07
## BugReport: https://github.com/spedygiorgio/markovchain/issues
```

```
# (i)

TransitionName<-c("Pass","Shoot","Dribble")
MarkovData<-c(0.5,0.1,0.4,0.7,0.2,0.1,0.4,0.3,0.3)
```

[3M]

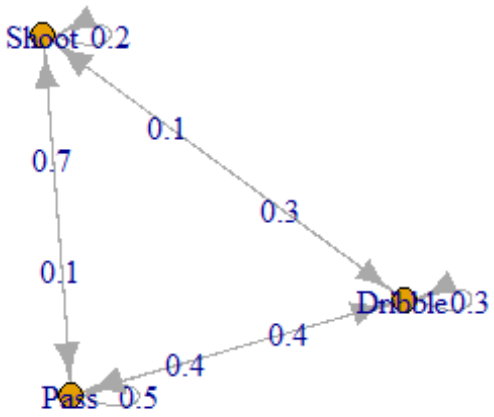
```
# (ii)

MarkovData<-c(0.5,0.1,0.4,0.7,0.2,0.1,0.4,0.3,0.3)
TransitionName<-c("Pass","Shoot","Dribble")
BallTransition<-matrix(MarkovData,nrow = 3,byrow = TRUE,dimnames = list(Trans
itionName,TransitionName))
BallTransition

##         Pass Shoot Dribble
## Pass     0.5   0.1     0.4
## Shoot    0.7   0.2     0.1
## Dribble  0.4   0.3     0.3

markov2<-new("markovchain",transitionMatrix=matrix(data=MarkovData,
             byrow=TRUE,nrow=3,dimnames=list(TransitionName,TransitionName)))
```

[1M]

[2M]

```
# (iii)
plot(markov2)
```

[3M]




```
# (iv)
steadyStates(markov2)
```

[4M]

```
##           Pass     Shoot   Dribble
## [1,] 0.5047619 0.1809524 0.3142857

#(v)
#(a)
markov2^2

## Unnamed Markov chain^2
##  A  3 - dimensional discrete Markov Chain defined by the following states:
##  Pass, Shoot, Dribble
##  The transition matrix  (by rows)  is defined as follows:
##         Pass Shoot Dribble
## Pass    0.48  0.19    0.33
## Shoot   0.53  0.14    0.33
## Dribble 0.53  0.19    0.28

After 2 plays the probability is 0.48
```

[2.5M]

```
#(b)
markov2^5

## Unnamed Markov chain^5
##  A  3 - dimensional discrete Markov Chain defined by the following states:
##  Pass, Shoot, Dribble
##  The transition matrix  (by rows)  is defined as follows:
##            Pass   Shoot Dribble
## Pass    0.50475 0.18075 0.31450
## Shoot   0.50525 0.18100 0.31375
## Dribble 0.50450 0.18125 0.31425
```

```
After 5 plays the probability is 0.50475
```

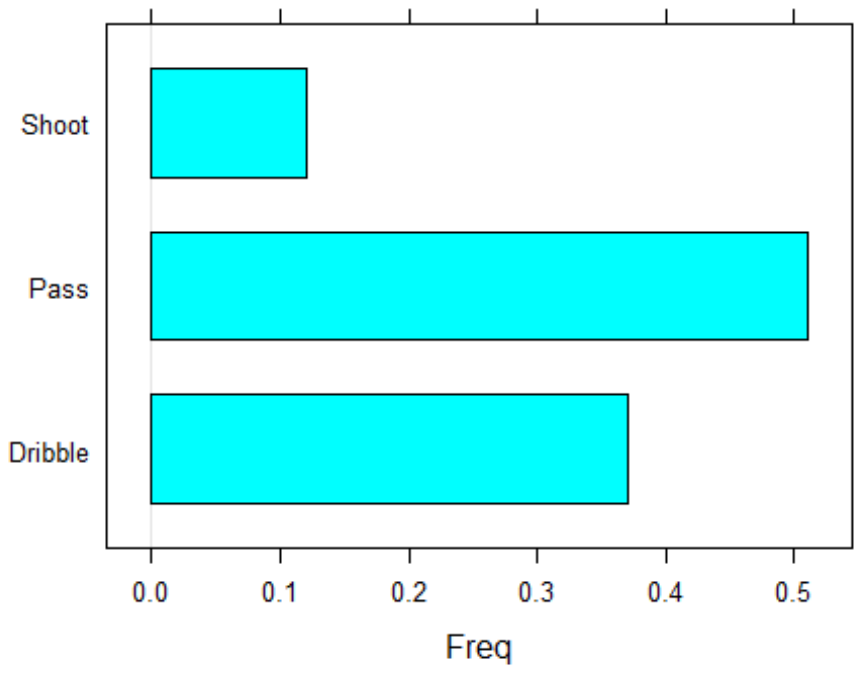
[2.5M]

```
#(vi)

set.seed(100)
seq<-markovchainSequence(100,markov2)
#Frequency of the terms
table(seq)

## seq
## Dribble    Pass   Shoot
##      37      51      12
```

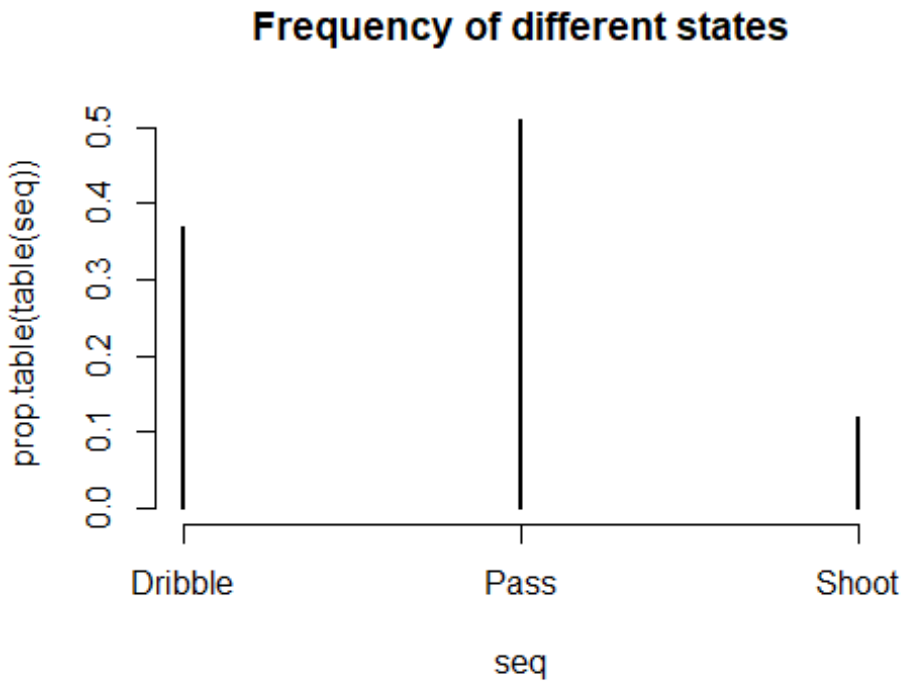
[4M]

```
#(vii)
library(lattice)

## Warning: package 'lattice' was built under R version 4.0.5

barchart(prop.table(table(seq)))
```

[2M]




```
# OR
plot(prop.table(table(seq)),main = "Frequency of different states")
```



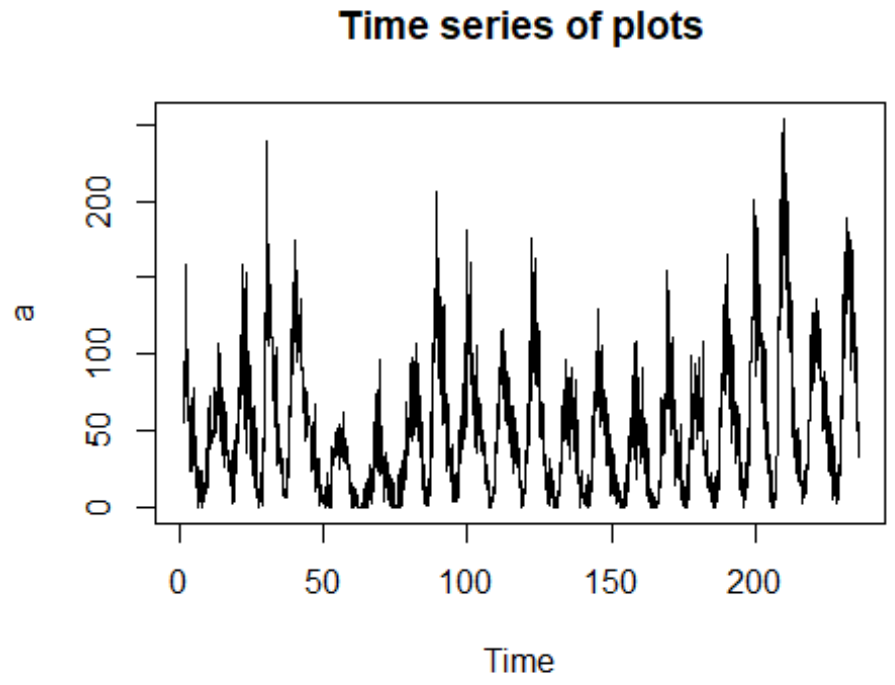

```
#(viii)                                                                    [3M]

# According to the steady state distribution, the frequency of Pass, Shoot an
d Dribble needs to be 50.47, 18.09, 31.42 respectively.
# Pass frequency of the simulation is matching with that of the expected. But
there is some discrepancy in the shoot and dribble.
# Main Reasons - (1) The number of simulations (100) is very small and needs
to be increased. When the simulations are small, different seed values can pr
oduce different results altogether because of sample error and those differen
ces might be significant
```

[27 Marks]

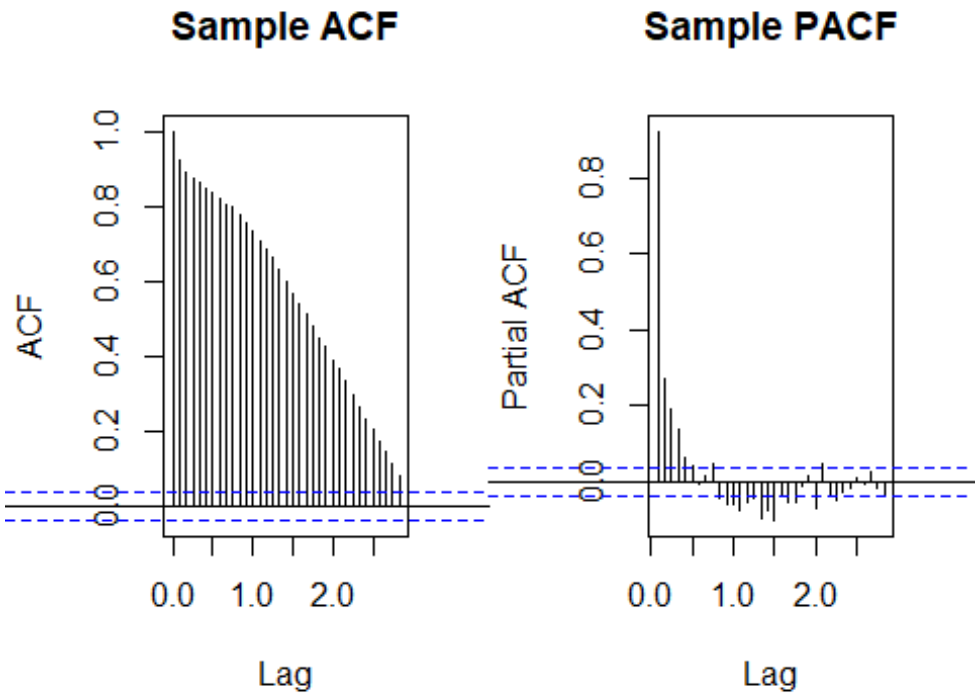
Solution 4:

```
library(tseries)

## Warning: package 'tseries' was built under R version 4.0.5

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

library(forecast)

## Warning: package 'forecast' was built under R version 4.0.5

# (i)
data(sunspots)
a<-ts(sunspots,frequency = 12)
plot(a, main = "Time series of plots")
```

[3M]




```r
#(ii)
par(mfrow=c(1,2))
acf(a,main="Sample ACF")
pacf(a,main="Sample PACF")
```



[5M]

```
# (iii)
acf(a,plot = FALSE)

##
## Autocorrelations of series 'a', by lag
##
## 0.0000 0.0833 0.1667 0.2500 0.3333 0.4167 0.5000 0.5833 0.6667 0.7500 0.83
33
##  1.000  0.922  0.890  0.875  0.864  0.850  0.836  0.819  0.805  0.797  0.7
78
## 0.9167 1.0000 1.0833 1.1667 1.2500 1.3333 1.4167 1.5000 1.5833 1.6667 1.75
00
##  0.756  0.734  0.710  0.686  0.663  0.632  0.602  0.569  0.542  0.512  0.4
80
## 1.8333 1.9167 2.0000 2.0833 2.1667 2.2500 2.3333 2.4167 2.5000 2.5833 2.66
67
##  0.451  0.426  0.391  0.367  0.334  0.298  0.264  0.233  0.204  0.173  0.1
47
## 2.7500 2.8333
##  0.115  0.083

pacf(a,plot = FALSE)

##
## Partial autocorrelations of series 'a', by lag
##
## 0.0833 0.1667 0.2500 0.3333 0.4167 0.5000 0.5833 0.6667 0.7500 0.8333 0.91
67
##  0.922  0.272  0.189  0.135  0.064  0.044 -0.005  0.014  0.046 -0.045 -0.0
58
## 1.0000 1.0833 1.1667 1.2500 1.3333 1.4167 1.5000 1.5833 1.6667 1.7500 1.83
33
## -0.059 -0.077 -0.056 -0.046 -0.099 -0.075 -0.102 -0.037 -0.055 -0.054 -0.0
13
## 1.9167 2.0000 2.0833 2.1667 2.2500 2.3333 2.4167 2.5000 2.5833 2.6667 2.75
00
##  0.014 -0.068  0.045 -0.034 -0.050 -0.029 -0.019  0.012 -0.006  0.026 -0.0
17
## 2.8333
## -0.039
```

[2M]

```
# (iv)
'The ACF indicates that there is correlation between an observation and
the past observations. There is a correlation beyond 2.5 years but it keeps g
etting weaker'

## [1] "The ACF indicates that there is correlation between an observation an
d\nthe past observations. There is a correlation beyond 2.5 years but it keep
s getting weaker"
```

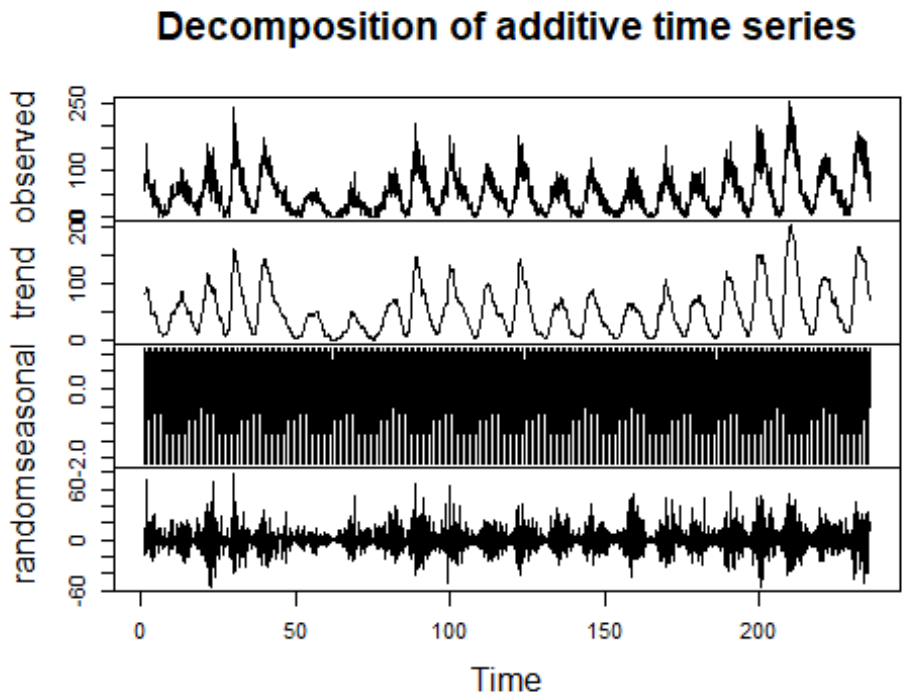
[2M]

```
# (v)
s<-diff(a, lag=12,differences=1)
head(s,12)
```

[2M]

```
##     Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
## 2  15.3  13.3  19.2  32.6   5.0  16.5  -9.4  36.7  15.3  -9.8 -95.3  -9.8
```

```
# (vi)
f<-decompose(a)
plot(f)
```



[4M]

```
# (vii)
'The cyclical trend with more variations at the longer time lags,
there is seasonality in the dataset'

## [1] "The cyclical trend with more variations at the longer time lags, \nth
ere is seasonality in the dataset"
```

[2M]

[20 Marks]

Solution 5:

```
library(tree)

## Warning: package 'tree' was built under R version 4.0.5

# (i)
DTData <- read.csv("DTData.csv",stringsAsFactors=TRUE)
train_Data<-DTData[1:round(0.7*332,0),]
test_Data<-DTData[(round(0.7*332,0)+1):332,]
```

[3M]

```
# (ii)
model<-tree(type~bmi+age,data = train_Data)
model

## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 232 297.60 No ( 0.65948 0.34052 )
```

[3M]

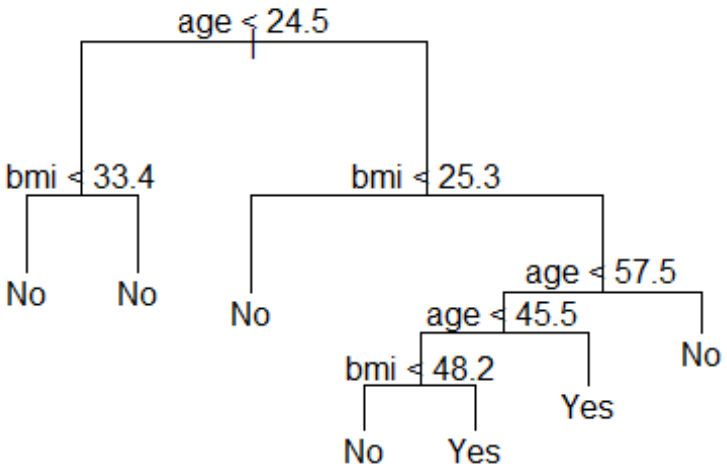
```
##    2) age < 24.5 73  46.13 No ( 0.90411 0.09589 )
##      4) bmi < 33.4 48   0.00 No ( 1.00000 0.00000 ) *
##      5) bmi > 33.4 25  29.65 No ( 0.72000 0.28000 ) *
##    3) age > 24.5 159 219.00 No ( 0.54717 0.45283 )
##      6) bmi < 25.3 16   0.00 No ( 1.00000 0.00000 ) *
##      7) bmi > 25.3 143 198.20 Yes ( 0.49650 0.50350 )
##       14) age < 57.5 137 189.60 Yes ( 0.47445 0.52555 )
##         28) age < 45.5 120 166.10 No ( 0.52500 0.47500 )
##           56) bmi < 48.2 114 156.80 No ( 0.55263 0.44737 ) *
##           57) bmi > 48.2 6   0.00 Yes ( 0.00000 1.00000 ) *
##         29) age > 45.5 17  12.32 Yes ( 0.11765 0.88235 ) *
##       15) age > 57.5 6   0.00 No ( 1.00000 0.00000 ) *
```

```
#(iii)
par(mfrow = c(1,1))
summary(model)
```

[4M]

```
##
## Classification tree:
## tree(formula = type ~ bmi + age, data = train_Data)
## Number of terminal nodes:  7
## Residual mean deviance:  0.8833 = 198.7 / 225
## Misclassification error rate: 0.2586 = 60 / 232
```

```
plot(model)
text(model)
```




```
#(iv)
pred<-predict(model,test_Data,type = "class")
```

[3M]

```
# (v)
summary(pred)
```

[3M]

```
##  No Yes
##  89  11

# (vi)
table(pred,test_Data$type)
```

[4M]

```
##
## pred  No Yes
##   No  67  22
##   Yes  3   8

accuracy<-sum(pred==test_Data$type)/nrow(test_Data)
type1_Error<-sum(pred=="Yes"&test_Data$type=="No")/nrow(test_Data)
type2_Error<-sum(pred=="No"&test_Data$type=="Yes")/nrow(test_Data)
accuracy

## [1] 0.75

type1_Error

## [1] 0.03

type2_Error

## [1] 0.22

# (vii)
# Accuracy is 75%. The null accuracy is 70% if all the observations would hav
e been classified to "No" class.
# Type 2 error is very high. Majority of the people who belonged to the type
"Yes", could not be successfully identified by the model
# Addressing the bias
# The bias of a model is a measure of how close our prediction is to the actu
al value on average from an average model
#a validation data set: the sample of data used to provide an unbiased evalua
tion of model fit on the training dataset while tuning model hyper-parameters
```

[4M]

[24 Marks]

****