Institute of Actuaries of India

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

December 2022 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

```
> library(MASS)
> options(scipen = 5)
> #set seed to 1234
> set.seed(1234)
```

i)

```
> x<-read.csv("CS2BQ1.csv")
```
(1)

ii)

```
> #Find mean and sd
> EmpiricalMean<-mean(x$Losses)
> EmpiricalSD<-sd(x$Losses)
> EmpiricalMean
[1] 49789.86
> EmpiricalSD
[1] 102948.2
```
(2)

iii)

```
> MoMmu<-log(EmpiricalMean/(1+EmpiricalSD^2/EmpiricalMean^2)^0.5)
> MoMsigma<-log(1+EmpiricalSD^2/EmpiricalMean^2)^0.5
> MoMmu
[1] 9.984059
> MoMsigma
[1] 1.28958
```
(3)

iv)

```
> #fit a lognormal distribution to the dataset using fitdistr in the MASS package
> fitLogNormal<-fitdistr(x$Losses,"lognormal")
> mu1<-fitLogNormal$estimate[1]
> sigma1<-fitLogNormal$estimate[2]
> mu1
 meanlog
9.989917
> sigma1
sdlog
1.268845
```
(3)

v)

> 'The Mu and Sigma estimates between the Method of Moments and Method of MLE approach are quite close'.

(3)

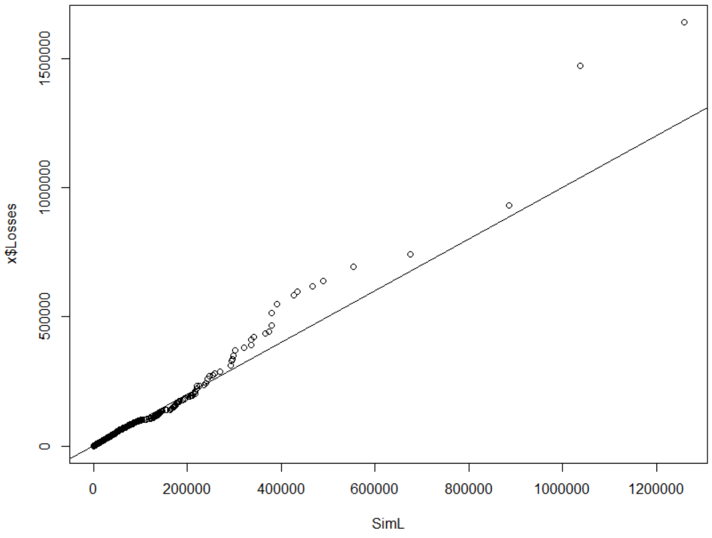
vi)

```
> #Simulate losses from a lognormal distribution using the mu and sigma estimated above
> SimL<-rlnorm(n=1000,mu,sigma)
> #calculate the mean and sd for the simulated distribution
> mean(SimL)
[1] 47238.53
> sd(SimL)
```

```
[1] 88548.16
```

(4)

vii)



(4)

viii)

> 'The fit of the loss distribution is fairly good till loss values of 25,000. Beyond this the qqplot indicates the data is not normally distributed for larger/tail values'

(3)

ix)

```
> #using the quantile function calculate the loss percentiles for every percentile from 0 to 100% with
steps of 10%
> Empirical<-quantile(x$Losses,probs = seq(0, 1, 0.1))
> Simulated<-quantile(SimL,probs = seq(0, 1, 0.1))
> Empirical
      0%       10%       20%       30%       40%       50%       60%       70%       80%       90%      100%
   384.0    4522.9    7587.8   11050.5   15884.4   22200.0   29871.6   41800.1   63515.2  106102.3
1638436.0
> Simulated
        0%          10%          20%          30%          40%          50%          60%          70%          80%
   293.1955    4677.5738    7429.3605   11021.5122   15189.9273   20731.8091   27842.9382
39392.2496   57282.2480
        90%         100%
 118552.0676 1257982.3513
```

(3)

x)

```
> E<-20000
> Retained<-pmin(E,SimL)
> Transferred<-SimL-Retained
```

(3)

xi)

```
> #quantiles for retained and transferred losses
> RetainedPercentile<-quantile(Retained,probs = seq(0, 1, 0.1))
```

```
> TransferredPercentile<-quantile(Transferred,probs = seq(0, 1, 0.1))
> RetainedPercentile
      0%        10%        20%        30%        40%        50%        60%        70%        80%        90%       100%
  293.1955   4677.5738   7429.3605  11021.5122  15189.9273  20000.0000  20000.0000  20000.0000
20000.0000 20000.0000 20000.0000
> TransferredPercentile
        0%          10%          20%          30%          40%          50%          60%          70%          80%
    0.0000       0.0000       0.0000       0.0000       0.0000     731.8091    7842.9382   19392.2496
37282.2480
       90%        100%
  98552.0676 1237982.3513
```

(3)

xii)

```
> #Combine the quantiles for Empirical, Simulated, Retained and Transferred losses into a data frame
with the first column
> #reflecting the quantile value
>c<-as.data.frame(cbind(seq(0,1,0.1),Empirical,Simulated,RetainedPercentile,TransferredPercentile))
> c
     % Empirical    Simulated RetainedPercentile TransferredPercentile
0%    0.0      384.0     293.1955          293.1955              0.0000
10%  0.1     4522.9    4677.5738         4677.5738              0.0000
20%  0.2     7587.8    7429.3605         7429.3605              0.0000
30%  0.3   11050.5   11021.5122        11021.5122              0.0000
40%  0.4   15884.4   15189.9273        15189.9273              0.0000
50%  0.5   22200.0   20731.8091        20000.0000            731.8091
60%  0.6   29871.6   27842.9382        20000.0000           7842.9382
70%  0.7   41800.1   39392.2496        20000.0000          19392.2496
80%  0.8   63515.2   57282.2480        20000.0000          37282.2480
90%  0.9  106102.3 118552.0676         20000.0000          98552.0676
100% 1.0 1638436.0 1257982.3513        20000.0000        1237982.3513
```

(3)

xiii)

'Comment on the difference in percentile loss values
There are differences at the higher percentiles with the simulated losses being higher than the empirical this reflects the lack of empirical data related to larger losses. The retained losses get capped at 20k at the 50th percentile suggesting that 1 out of 2 claims will get capped by the current policy Excess.'

(2)

xiv)

```
> LossVolatility<-quantile(Transferred,0.9)
> TechnicalPremium<-(mean(Transferred)+LossVolatility*0.1)
> TechnicalPremium
     90%
42229.48
```

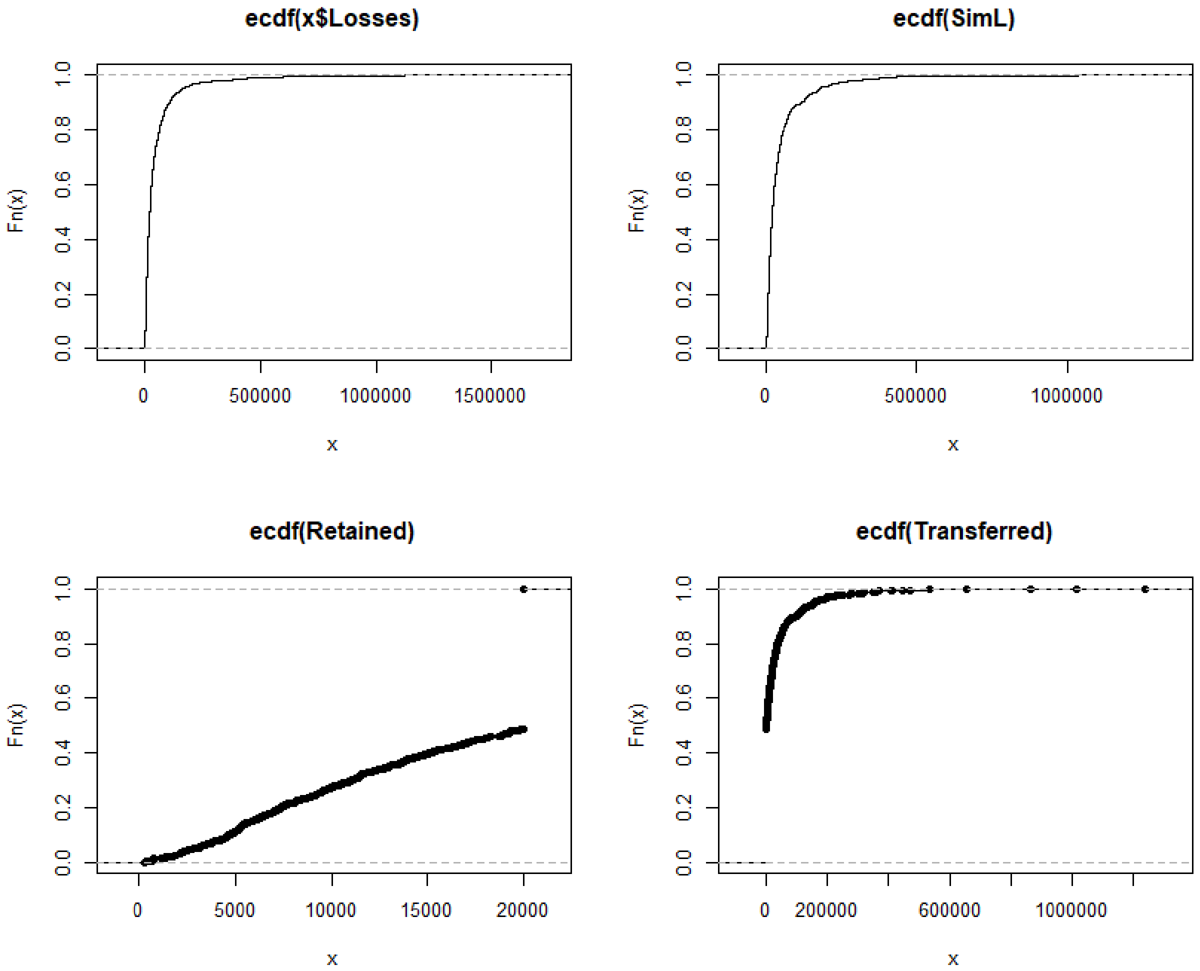
(2)

xv)

```
> Premium<-70000
> EfficiencyRatio<-TechnicalPremium/Premium
> 'The Actual Premium is higher than the Technical Premium and the efficiency ratio is 60%.'
```

(2)

xvi)



(4)

[45 Marks]

Solution.2

library(survival)

i)

> cph1r<-coxph(Surv(futime,fustat)~rx,ovarian)

[1 mark for each correct variable and 1 for overall formula] (4)

ii)

```
> summary(cph1r)
Call:
coxph(formula = Surv(futime, fustat) ~ rx, data = ovarian)

  n= 26, number of events= 12

      coef exp(coef) se(coef)      z Pr(>|z|)
rx -0.5964    0.5508   0.5870 -1.016     0.31

   exp(coef) exp(-coef) lower .95 upper .95
rx    0.5508      1.816    0.1743      1.74

Concordance= 0.608  (se = 0.07 )
Likelihood ratio test= 1.05  on 1 df,   p=0.3
Wald test            = 1.03  on 1 df,   p=0.3
Score (logrank) test = 1.06  on 1 df,   p=0.3
```

(2)

iii)
> 'the p-value of rx is 0.31 which is greater than a
 significance level of 0.05 and hence is not a siginificant predictor'
> 'The hazard ratio indicates that an rx value of 2 has a 0.5508 or ~50%
 lower hazard rate than rx value of 1'

(3)

iv)

```
> KM<-survfit(Surv(futime,fustat)~rx,ovarian)
> km <- Surv(time = ovarian[['futime']], event = ovarian[['fustat']])
> km_treatment<-survfit(km~rx,data=ovarian,type='kaplan-meier',conf.type='log')
> KM or km_treatment [Either approach is fine]
Call: survfit(formula = Surv(futime, fustat) ~ rx, data = ovarian)

      n events median 0.95LCL 0.95UCL
rx=1 13      7    638     268      NA
rx=2 13      5     NA     475      NA
> km_treatment
```

(3)

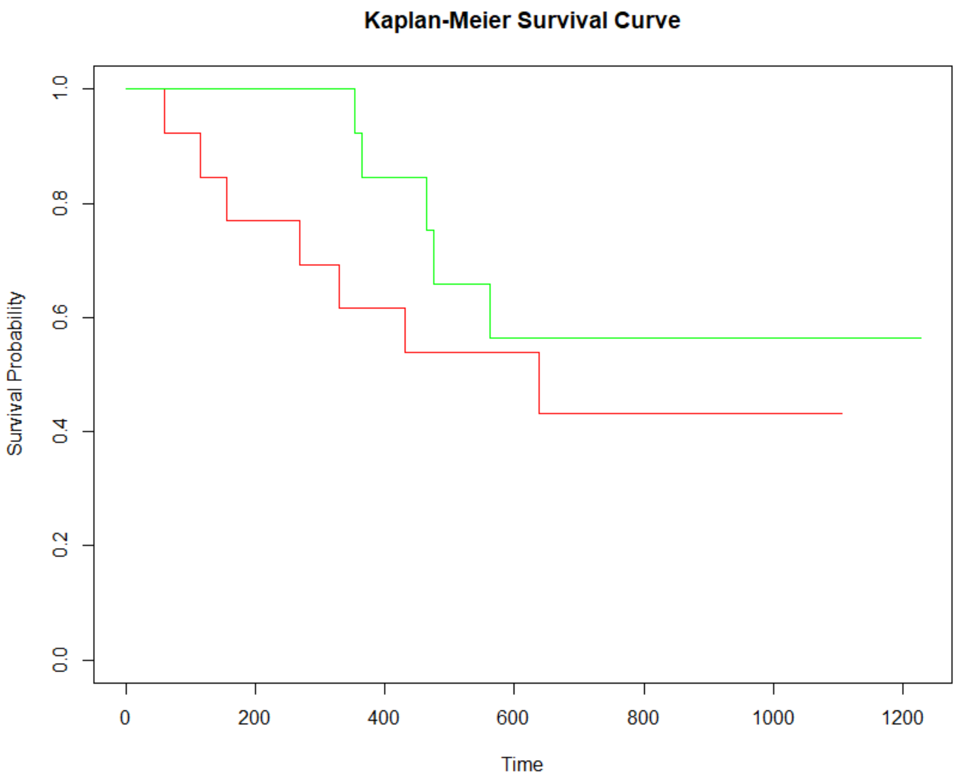
v)

```
> summary(KM)
Call: survfit(formula = Surv(futime, fustat) ~ rx, data = ovarian)

                rx=1
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
   59     13       1    0.923  0.0739        0.789        1.000
  115     12       1    0.846  0.1001        0.671        1.000
  156     11       1    0.769  0.1169        0.571        1.000
  268     10       1    0.692  0.1280        0.482        0.995
  329      9       1    0.615  0.1349        0.400        0.946
  431      8       1    0.538  0.1383        0.326        0.891
  638      5       1    0.431  0.1467        0.221        0.840

                rx=2
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
  353     13       1    0.923  0.0739        0.789        1.000
  365     12       1    0.846  0.1001        0.671        1.000
  464      9       1    0.752  0.1256        0.542        1.000
  475      8       1    0.658  0.1407        0.433        1.000
  563      7       1    0.564  0.1488        0.336        0.946
```

(2)

vi)

```
> plot(KM,col=c("red","green"),main="Kaplan-Meier Survival Curve",xlab="Time",ylab="Survival Probability")
```



(5)

vii)

> 'rx 2 seems to perform better than rx 1 initially upto time 353 post which there is a decline in the Survival Probability. There is a steep decline in the numbers at risk between times 365 and 464 for rx at 2'

(3)

[22 marks]

Solution.3

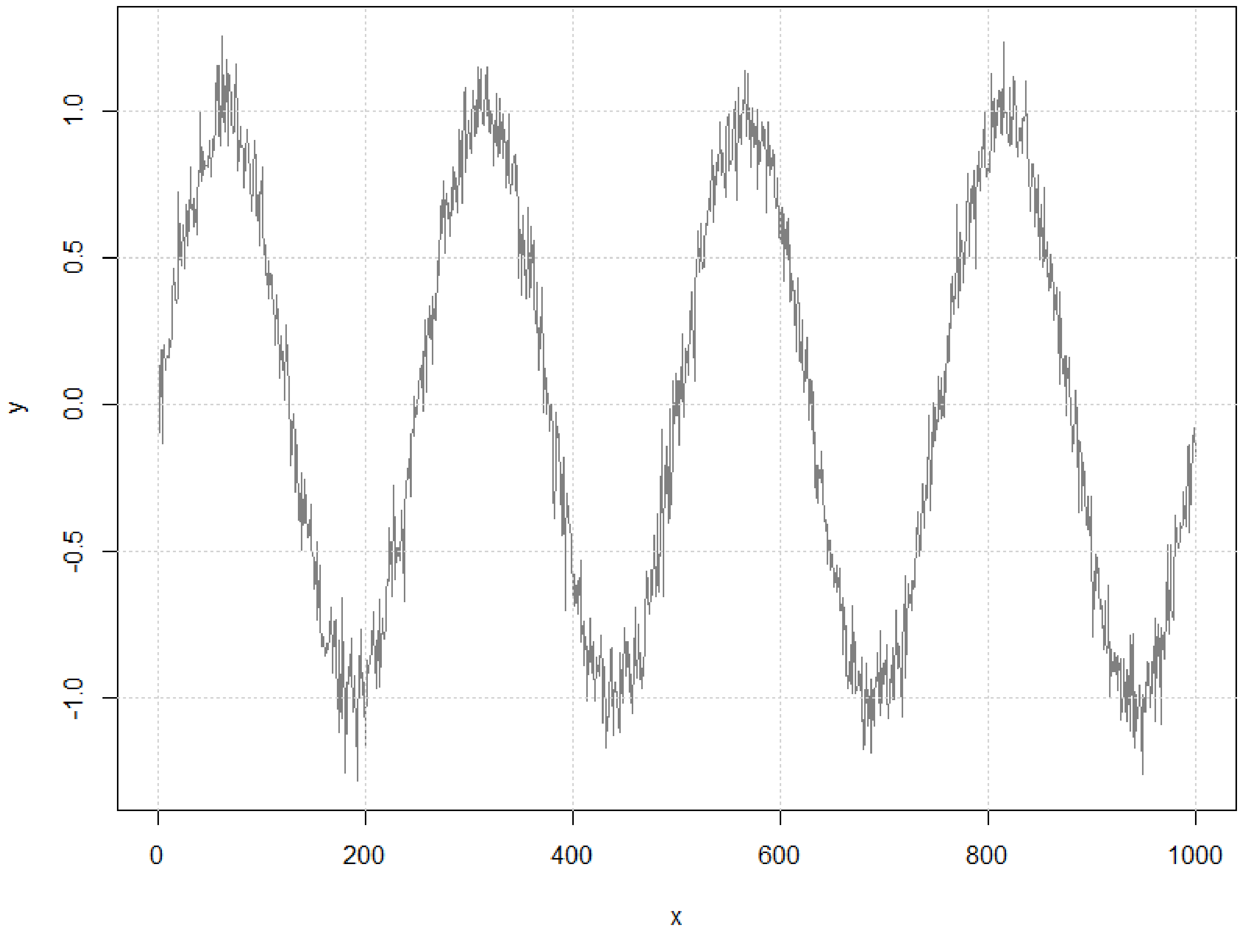
i)

```
> #Generate time series
> set.seed(1234)
> x <- 1:1000
> y <- sin(x/40) + rnorm(1000,sd=.1)
```

(4)

ii)

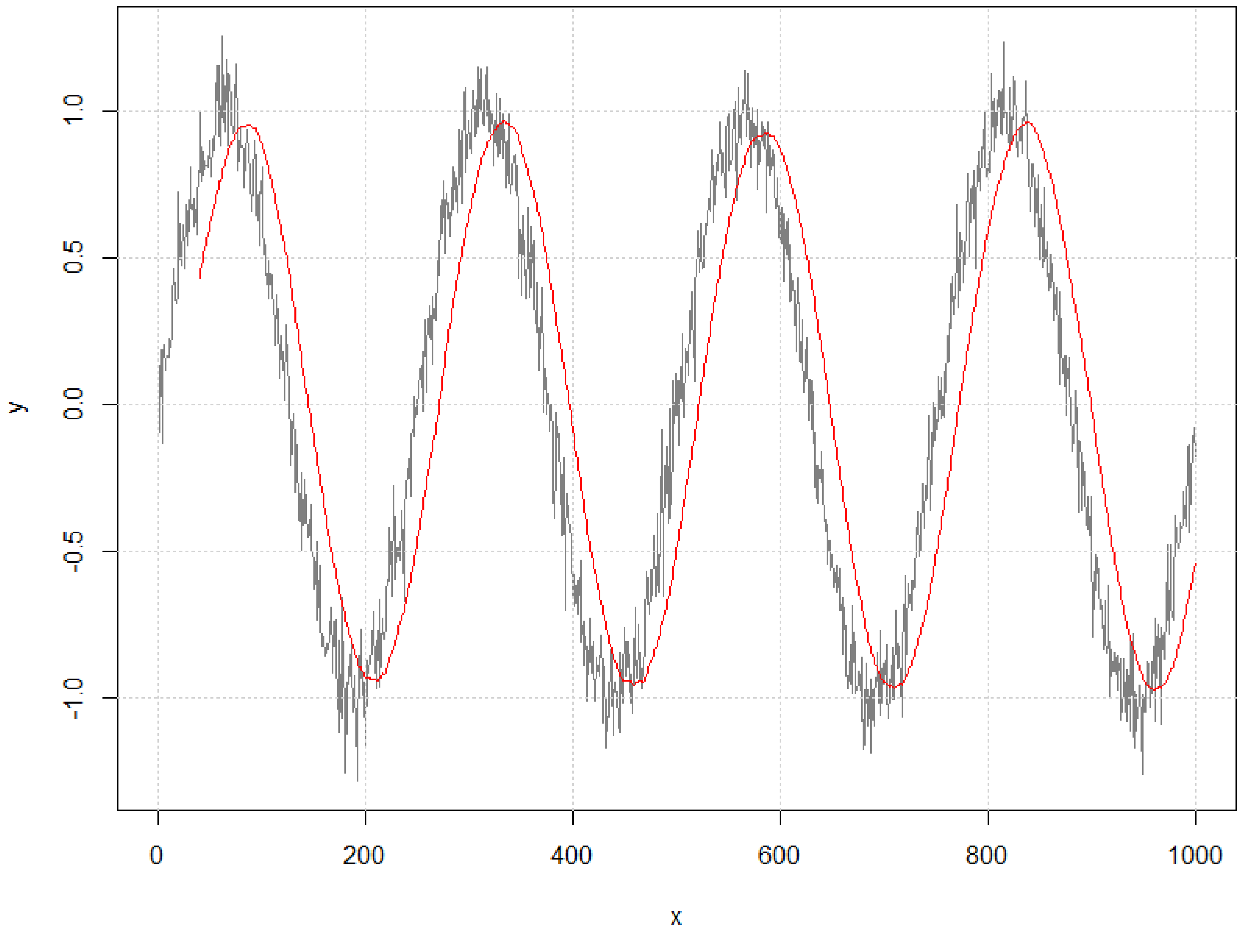
```
> # Plot the unsmoothed data (gray)
> plot(x, y, type="l", col=grey(.5))
> # add gridlines
> grid()
```



(2)

iii)

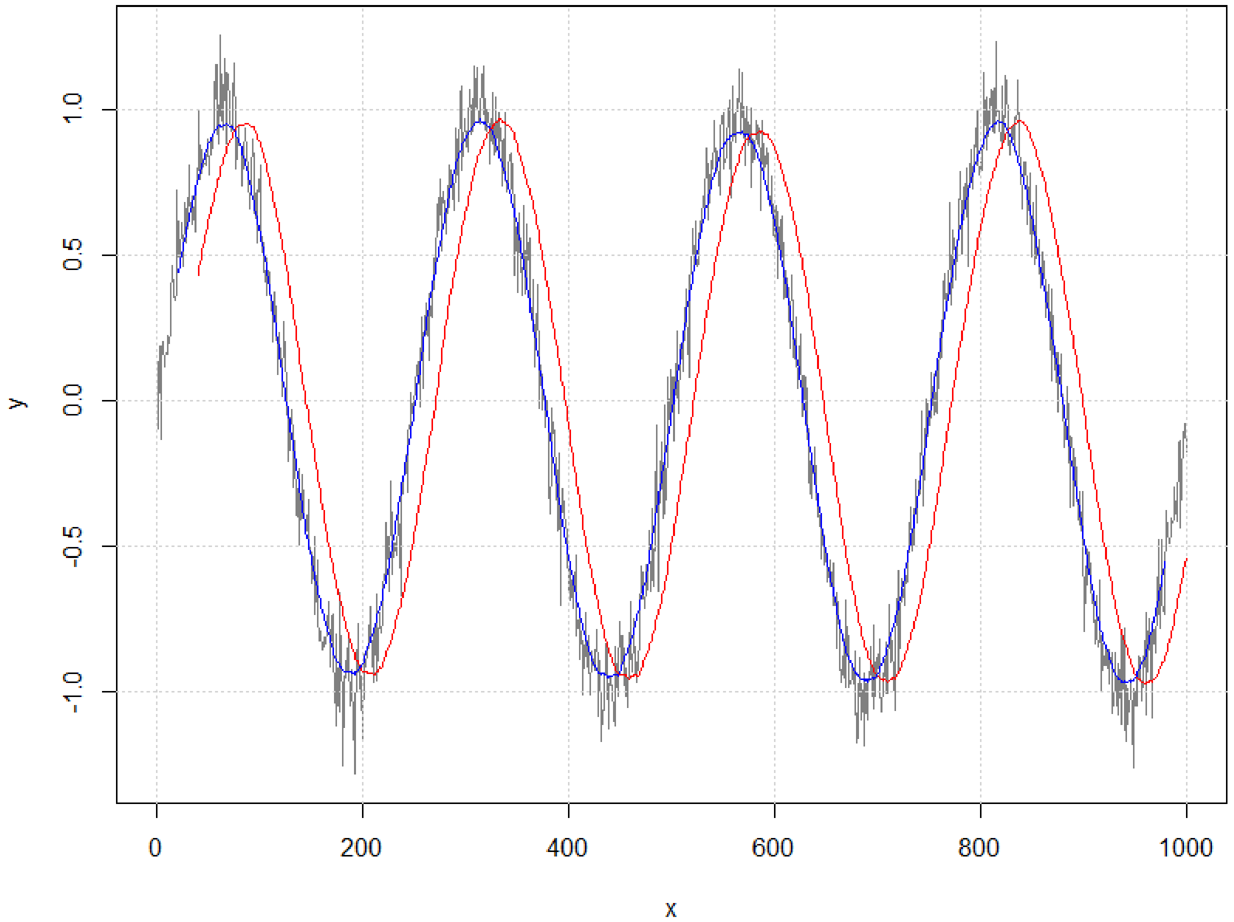
```
> # Smoothed with lag:
> # average of current sample and 39 previous samples (red)
> f40 <- rep(1/40, 40)
> f40
 [1] 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025
[21] 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025 0.025
> y_lag <- filter(y, f40, sides=1)
> lines(x, y_lag, col="red")
```



(3)

iv)

```
> # Smoothed symmetrically:
> # average of current sample, 20 future samples, and 20 past samples (blue)
> f41 <- rep(1/41,41)
> f41
 [1] 0.02439024 0.02439024 0.02439024 0.02439024 0.02439024 0.02439024 0.02439024
0.02439024 0.02439024 0.02439024
[11] 0.02439024 0.02439024 0.02439024 0.02439024 0.02439024 0.02439024 0.02439024
0.02439024 0.02439024 0.02439024
[21] 0.02439024 0.02439024 0.02439024 0.02439024 0.02439024 0.02439024 0.02439024
0.02439024 0.02439024 0.02439024
[31] 0.02439024 0.02439024 0.02439024 0.02439024 0.02439024 0.02439024 0.02439024
0.02439024 0.02439024 0.02439024
[41] 0.02439024
> y_sym <- filter(y, f41, sides=2)
> lines(x, y_sym, col="blue")
```



(4)

[13 Marks]

Solution.4

```
> fraud <- read.csv("C:/ fraud.csv", stringsAsFactors=TRUE)
```

i)

```
> x<-prop.table(table(fraud$Fraudulent))[2]
> x
  Yes
0.057
```

(2)

ii)

```
> y<-sum(fraud$State == "C3"& fraud$Sum_insured =="Medium")/nrow(fraud)
> y
[1] 0.199
```

(3)

iii)

```
>fraud_claims<-fraud[fraud$Fraudulent=="Yes",]
>z<-sum(fraud_claims$State=="C3"&fraud_claims$Sum_insured =="Medium")/nrow(fraud_claims)
> z
[1] 0.4035088
```

(4)

iv)

```
> prob<-x*z/y
```

```
> prob
      Yes
0.1155779
```

(4)

v)

```
> subset2<-fraud[fraud$Sum_insured=="Medium"&fraud$State == "C3",]
> sum(subset2$Fraudulent == "Yes")/nrow(subset2)
[1] 0.1155779
```

(3)

vi)

> 'The values need to be equal as this is an application of Bayes Theorem'

(2)

vii)

>'(a) Assumes that all the variables are independent'

>'(b) If your test data set has a categorical variable of a category that wasn't present in the training data set, the Naive Bayes model will assign it zero probability and will not be able to make any predictions in this regard'

(2)

[20 Marks]

****