# INSTITUTE AND FACULTY OF ACTUARIES

# EXAMINERS' REPORT

September 2021

## CS2 – Risk Modelling and Survival Analysis
## Core Principles
## Paper B

**Introduction**

The Examiners' Report is written by the Chief Examiner with the aim of helping candidates, both those who are sitting the examination for the first time and using past papers as a revision aid and also those who have previously failed the subject.

The Examiners are charged by Council with examining the published syllabus. The Examiners have access to the Core Reading, which is designed to interpret the syllabus, and will generally base questions around it but are not required to examine the content of Core Reading specifically or exclusively.

For numerical questions the Examiners' preferred approach to the solution is reproduced in this report; other valid approaches are given appropriate credit. For essay-style questions, particularly the open-ended questions in the later subjects, the report may contain more points than the Examiners will expect from a solution that scores full marks.

The report is written based on the legislative and regulatory context pertaining to the date that the examination was set. Candidates should take into account the possibility that circumstances may have changed if using these reports for revision.

Sarah Hutchinson
Chair of the Board of Examiners
December 2021

## A. General comments on the *aims of this subject and how it is marked*

The aim of the Risk Modelling and Survival Analysis subject is to provide a grounding in mathematical and statistical modelling techniques that are of particular relevance to actuarial work, including stochastic processes and survival models.

Candidates are reminded of the need to include the R code, that they have used to generate their solutions, together with the main R output produced, in their answer script. Where the R code was missing from a particular question part, no marks were awarded even if the output (e.g. a graph) was included. Partial credit was awarded in the cases where the R code was included but the R output was not.

The marking schedule below sets out potential R code solutions for each question. Other appropriate R code solutions gained full credit unless one specific approach had been explicitly requested in the question paper.

In cases where the same error was carried forward to later parts of the answer, candidates were given full credit for the later parts.

In higher order skills questions, where comments were required, well-reasoned comments that differed from those provided in the solutions also received credit as appropriate.

## B. Comments on *candidate performance in this diet of the examination.*

On the whole, performance was generally satisfactory. Candidates typically demonstrated their ability to use R to perform analysis but did not fully demonstrate their ability to interpret the results.

It is important that appropriate commentary is provided alongside the R code and R output in the answer script, where relevant, to fully demonstrate sufficient understanding. For example, in questions requiring charts, appropriate titles, axis labels and legends are necessary, and in questions requiring a specific numerical answer, this must be stated separately from the R output. Instructions to this effect were communicated to candidates at the time of the exam. Candidates are advised to take careful note of all instructions that are provided with the exam in order to maximise their performance in CS2B examinations. The instructions applicable to this diet can be found at the beginning of the solutions contained within this document.

Higher order skills questions were answered very poorly. Candidates should recognise that these are generally the questions which differentiate those candidates with a good grasp and understanding of the subject.

Candidates are reminded that, where they are unable to answer one part of a question, the best approach is to provide a "dummy" answer and carry on with the remaining parts of the question to receive carry forward credit.

## C. Pass Mark

The Pass Mark for this exam was 58
1,264 presented themselves and 440 passed.

## Solutions for Subject CS2B – September 2021

Please note the following conventions / principles that apply to this marking schedule:

Candidates **MUST** include the R code used to obtain their answers in the Word document. Please note that failure to include the R code used will result in **ZERO MARKS** for that particular question.

Candidates **MUST** include the main R output produced from the R code in the Word document. Please note that failure to include the R output will result in full credit not being given.

When a question requires data to be simulated or generated in R, candidates **DO NOT** need to paste the individual values of the generated data into the Word document, unless specifically instructed to do so in the question.

When a question requires a particular numerical answer or conclusion, candidates **MUST** explicitly and clearly state this in the Word document, separately from, and in addition to the R output that contains the relevant numerical information. Please note that failure to include a separate answer or conclusion will result in full credit not being given.

Candidates should type any non-R code workings and answers into the Word document using standard keyboard typing. Candidates **DO NOT** need to use notation that requires specialised equation editing e.g. the “Equation Editor” functionality in Word.

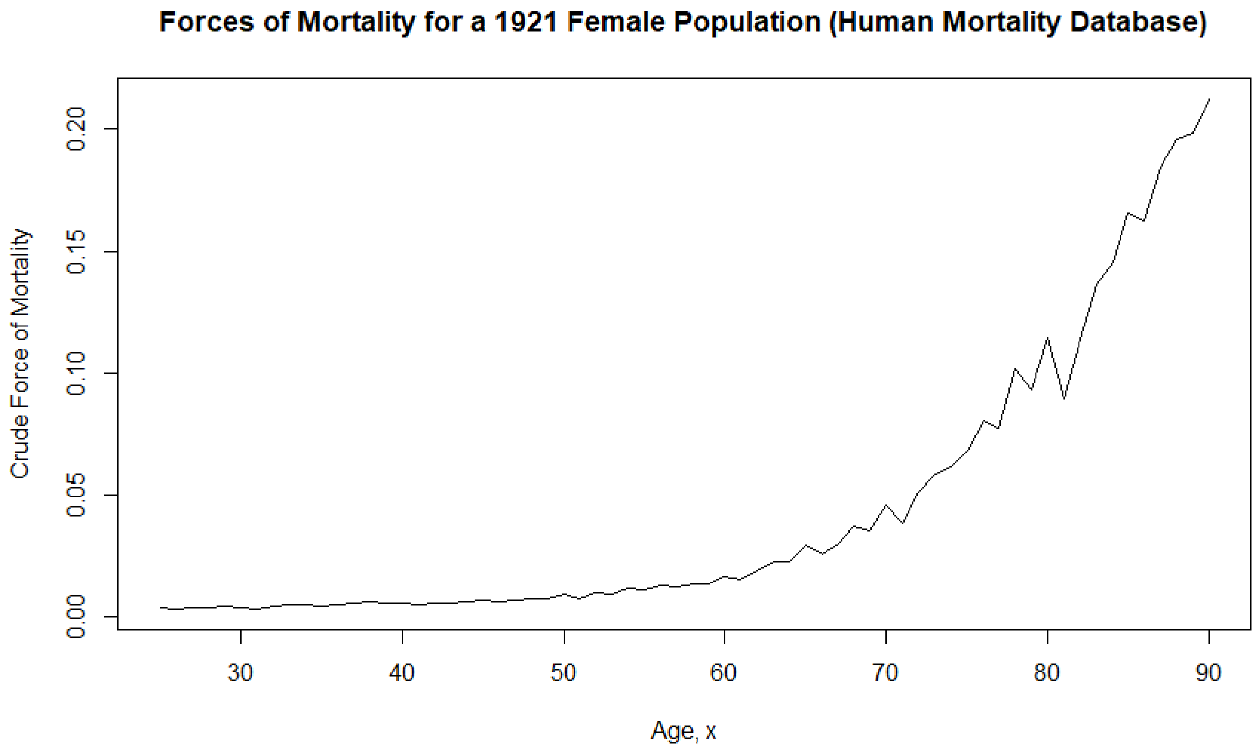
Candidates **MUST** include appropriate titles, axes labels, and where relevant, legends in all graphical output that is generated in R for inclusion in the Word document. Please note that failure to include appropriate annotations will result in full credit not being given.

Candidates should provide relevant comments when instructed to do so in the question.
Your Word document **MUST NOT** contain links to any other documents.

### Q1

(i)

```
mortalitydata = read.csv("CS2B_Sept_21_Qu_1_Data.csv”)          [1½]
plot(                                                            [½]
mortalitydata$Age,                                               [½]
mortalitydata$mu_x,                                              [½]
type = "l",                                                      [1]
xlab = "Age, x",                                                 [½]
ylab = "Crude Force of Mortality",                               [½]
main = "Forces of Mortality for a 1921 Female Population (Human
     Mortality Database)")                                       [½]
```



[½]

(ii)

**EITHER:**

The *mu_x*'s increase with age [1]
increasing rapidly from age 60 onwards [1]

**OR:**

The *mu_x*'s increase approximately exponentially with age [2]

**THEN:**

They exhibit some roughness [½]
especially between ages 70 and 85 years [½]
They should be graduated before being used [1]

(iii)

**EITHER:**

`GompModel = lm(` [1]
`log(mu_x)` [1]
`~ Age,` [1]
`data = mortalitydata)` [1]

**OR**

`GompModel = lm(` [1]
`log(mortalitydata$mu_x)` [1½]
`~ mortalitydata$Age)` [1½]

**THEN EITHER:**

```
exp(as.numeric(coef(GompModel)[1]))                          [½]
[1] 0.0003828807                                             [½]
exp(as.numeric(coef(GompModel)[2]))                          [½]
[1] 1.06997                                                  [½]
```

And hence:
$B = 0.0003828807$ [½]
$c = 1.06997$ [½]

**OR:**

```
summary(GompModel)                                           [½]
Call:
lm(formula = log_mux ~ Age, data = mortalitydata)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.49034 -0.23744  0.02004  0.23381  0.53057 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -7.867787   0.106390  -73.95   <2e-16 ***
Age          0.067630   0.001756   38.51   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2718 on 64 degrees of freedom
Multiple R-squared:  0.9586,	Adjusted R-squared:  0.958 
F-statistic:  1483 on 1 and 64 DF,  p-value: < 2.2e-16
                                                             [½]
```

The fitted Gompertz formula for ln(*mu_x*) is therefore:
ln (*mu_x*) = -7.867787 + 0.067630*x* [1]

And hence:
$B = 0.0003829$ [½]
$c = 1.0699693$ [½]

(iv)
The null hypothesis is that the Gompertz graduated rates are the true rates underlying the observed data [½]
The alternative hypothesis is that the Gompertz graduated rates are NOT the true rates underlying the observed data [½]

```
mortalitydata$fittedmu_x = exp(GompModel$coefficients[1] +
GompModel$coefficients[2] * mortalitydata$Age)               [1]

mortalitydata$zx = ((mortalitydata$mu_x * mortalitydata$Exposed_x) -
(mortalitydata$fittedmu_x * mortalitydata$Exposed_x)) /
sqrt(mortalitydata$fittedmu_x * mortalitydata$Exposed_x)     [2]

mortalitydata$zxsquared = (mortalitydata$zx)^2               [½]
chisquare = sum(mortalitydata$zxsquared); chisquare          [1]
```

```
[1] 114.8394
```
[½]

The test statistic has a chi-square distribution with *m* degrees of freedom, where *m* is the number of age groups less one for each Gompertz parameter fitted
So, in this case *m* = 66 - 2 = 64. [1]

**THEN EITHER:**
The critical value at the 5% level with 64 degrees of freedom is:

```
qchisq(0.95, 64)
[1] 83.67526
```

**OR:**
Using Page 169 of the Golden Book, the critical value at the 5% level with 70 degrees of freedom is 90.53 and so the critical value with 64 degrees of freedom is less than this

[1]

Since 114.8394 > critical value [½]
there IS enough evidence to reject the null hypothesis at the 5% level [½]

**[Total 26]**

*Part (i) was very well answered, except that few candidates specified what population the forces of mortality related to in their chart title. As the R code to import the data frame mortality data is provided in the question, candidates who did not include this code in their answer scripts were not penalised.*

*Part (ii) was fairly well answered. Most candidates included the comments about mortality increasing with age and about the increase being exponential or more rapid at higher ages. Fewer candidates included the comments about roughness, and fewer still commented on the need for graduation.*

*Part (iii) was well answered, although many candidates did not state their answers separately from the R output. Alternative valid methods of determining B and c were also awarded credit.*

*Part (iv) was well answered. The most common errors were:*
*Calculating the test statistic based on the observed and expected forces of mortality, rather than the numbers of deaths.*
*Using the wrong number of degrees of freedom.*

**Q2**

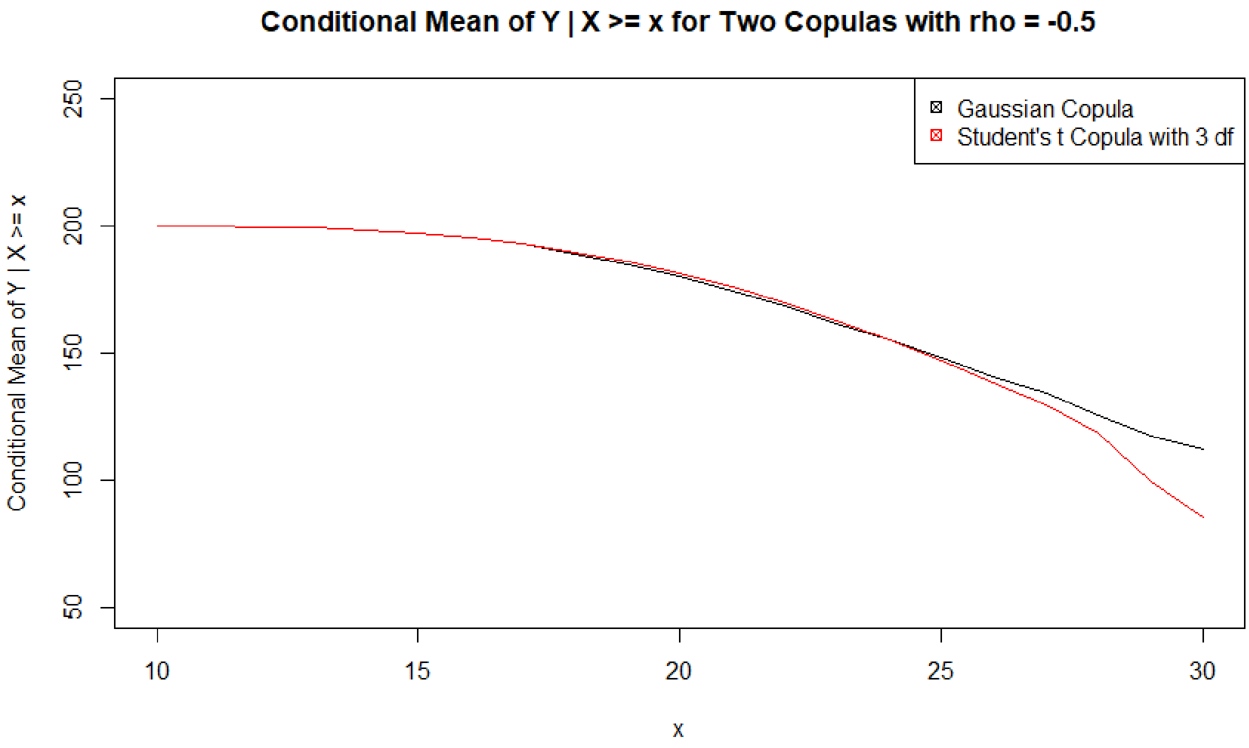
```
install.packages("copula")
library(copula)

set.seed(3)
GaussUV = rCopula(200000, normalCopula(param = -0.5))
GaussXY = matrix(nrow = 200000, ncol = 2)
GaussXY[,1] = qnorm(GaussUV[,1], mean = 20, sd = 3)
GaussXY[,2] = qnorm(GaussUV[,2], mean = 200, sd = 50)
```

(i)

```
set.seed(3)                                                          [½]

t3UV = rCopula(200000, tCopula(param = -0.5, df = 3))                [2]

t3XY = matrix(nrow = 200000, ncol = 2)                               [½]
t3XY[,1] = qnorm(t3UV[,1], mean = 20, sd = 3)                        [½]
t3XY[,2] = qnorm(t3UV[,2], mean = 200, sd = 50)                      [½]

head(t3XY)                                                           [½]
         [,1]     [,2]
[1,] 16.41179 197.2892
[2,] 21.82020 141.0359
[3,] 20.44033 199.1250
[4,] 19.41271 246.2435
[5,] 15.33383 278.9245
[6,] 18.40018 148.0110                                               [½]
```

(ii)

```
x = seq(from = 10, to = 30, by = 1)                                  [1]

y1 = vector(length = 21)                                             [½]
y2 = vector(length = 21)                                             [½]

for (i in 1:21) {                                                    [½]
     y1[i]= mean(GaussXY[,2][GaussXY[,1] >= x[i]])                   [3]
     y2[i]= mean(t3XY[,2][t3XY[,1] >= x[i]])                         [2]
}

plot(                                                                [½]
x,                                                                   [½]
y1,                                                                  [½]
ylim = c(50, 250),                                                   [1]
type = "l",                                                          [1]
ylab = "Conditional Mean of Y | X >= x",                             [½]
main = "Conditional Mean of Y | X >= x for Two Copulas with rho = -
0.5")                                                                [½]

lines(                                                               [½]
x,                                                                   [½]
y2,                                                                  [½]
col = "red")                                                         [1]

legend("topright",
legend = c("Gaussian Copula", "Student's t Copula with 3 df"),       [½]
col = c("black", "red"),                                             [½]
pch=7)
```



[½]

(iii)
For the lower values of *x*, the conditional means for both copulas are similar [1]
and close to the unconditional mean of 200 [1]
The conditional means for both copulas decrease with increasing *x* [1]
because of the negative value of *rho* [1]
Since the *t*3 copula exhibits positive tail dependence and the Gaussian copula has zero tail dependence [1]
the graph for the *t*3 copula slopes downwards more steeply than for the Gaussian copula [1]
The order of the Gaussian and *t*3 copulas is not consistent for all values of *x* (i.e. the red line is slightly above the black line from around *x* = 18 to *x* = 23) [2]
[Marks available 8, maximum 6]

(iv)
In order to draw any conclusions here, it is necessary to assume that the conditional distribution of *Y* given that *X* >= *x* remains unchanged for each value of *x* [1]
The graph in part (ii) indicates that the mean level of rainfall can be expected to decrease [1]
with the extent of the decrease being greater for the Student's *t*3 copula than for the Gaussian copula [1]
**[Total 30]**

*Part (i) was very well answered.*

*Part (ii) was poorly answered. Candidates who were unable to provide the correct R code for calculating the conditional mean could still have gained most of the marks by plotting "dummy" data, e.g. the unconditional mean.*

*Part (iii) was very poorly answered. Most candidates who produced a correct or nearly correct graph in part (ii) identified one or more relevant features of the graph, but few commented on those features by reference to the theoretical properties of the copulas. Alternative comments that were clear, distinct and relevant to the context of the question were also awarded credit.*

*Part (iv) was the most poorly answered question part on the whole paper. Many candidates provided comments on which copula is a more appropriate model, which cannot be assessed from the information in the question.*

**Q3**

```
install.packages("rpart")
install.packages("rpart.plot")
library(rpart)
library(rpart.plot)
```

(i)

```
set.seed(12345)                                              [½]
```

**THEN EITHER:**

```
U = matrix(runif(30000), ncol = 3)                          [2½]
```

**OR:**

```
U = matrix(nrow = 10000, ncol = 3)                           [½]

for (j in 1:3) {                                             [½]
     U[,j] = runif(10000)                                   [1½]
}
```

**THEN:**

```
head(U)                                                      [½]
              [,1]      [,2]       [,3]
     [1,] 0.7209039 0.2443204 0.56406258
     [2,] 0.8757732 0.6894012 0.99741215
     [3,] 0.7609823 0.8696410 0.70260977
     [4,] 0.8861246 0.9812336 0.95430918
     [5,] 0.4564810 0.5692775 0.09716026
     [6,] 0.1663718 0.1643290 0.74324952                     [½]
```

(ii)

```
a = 0                                                        [½]
b = 0.5                                                      [½]
c = 0.5                                                      [½]
```

**THEN EITHER:**

```
n = 10000                                                    [½]
f1 = vector(length = n)                                      [½]
f2 = vector(length = n)                                      [½]
```

```
f = vector(length = n)                                              [½]
defprob = vector(length = n)                                        [½]
default = vector(length = n)                                        [½]

for (i in 1:n) {                                                    [½]
     f1[i] = -1 *(U[i, 1] <= 1/3) + 1 * (U[i, 1] > 2/3)             [2]
     f2[i] = 2 * (U[i, 2] - 0.5)                                    [2]
     f[i] = a + b * f1[i] + c * f2[i]                               [1]
     defprob[i] = exp(f[i]) /(1 + exp(f[i]))                       [1½]
     default[i] = 1 * (U[i, 3] <= defprob[i])                       [2]
}
```

**OR:**

```
f1 = -1 *(U[,1] <= 1/3) + 1 * (U[,1] > 2/3)                         [3]
f2 = 2 * (U[,2] - 0.5)                                              [3]
f = a + b * f1 + c * f2                                            [1½]
defprob = exp(f) /(1 + exp(f))                                      [2]
default = 1 * (U[,3] <= defprob)                                   [2½]
```

**THEN:**

```
specimen = data.frame("f1" = f1, "f2" = f2, "default" = default)
                                                                    [½]

head(specimen)                                                      [½]
   f1         f2 default
 1  1 -0.5113592       0
 2  1  0.3788024       0
 3  1  0.7392819       1
 4  1  0.9624672       0
 5  0  0.1385551       1
 6 -1 -0.6713421       0                                            [½]
```

(iii)

```
mean(defprob)
```

**OR**

```
sum(defprob)/length(defprob)
                                                                    [1]

[1] 0.4993524                                                       [½]
```

The expected probability of customer default is 0.4993524 [½]

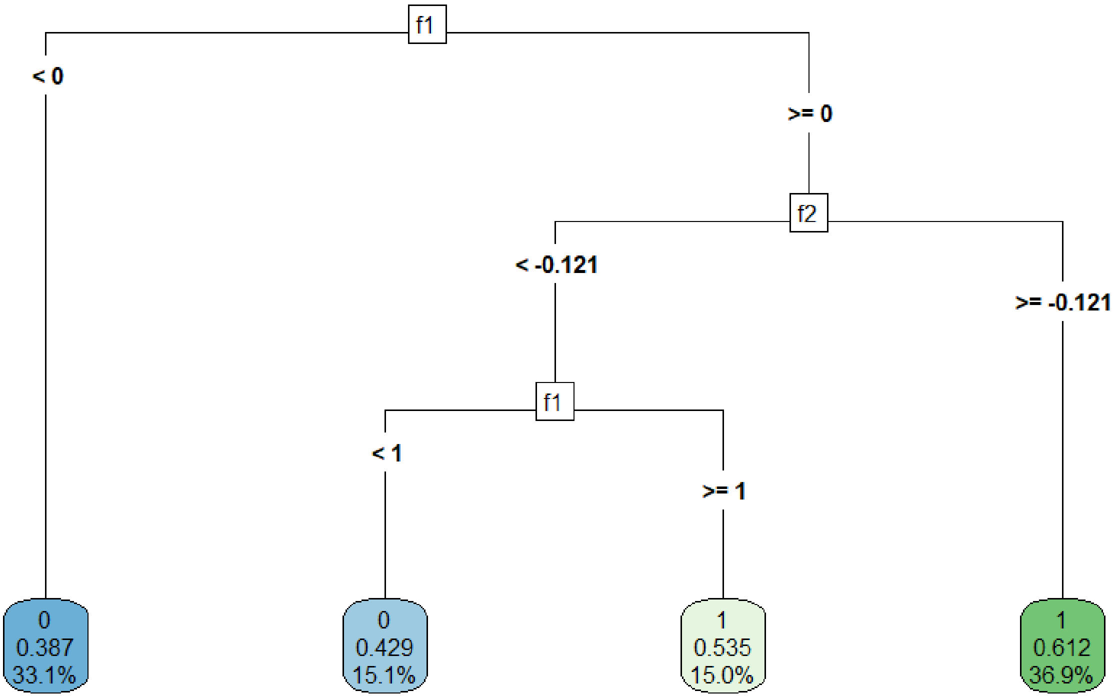
(iv)

The answer to part (iii) is not likely to be realistic [1]
as the mean default probability of their potential customers would be expected to be significantly less than 0.5 [1]
A negative value of *a* would be better [1]
so as to reduce the expected probability of default [1]

(v)

```
tree = rpart(default ~ ., data = specimen, method = "class")
rpart.plot(tree, digits = 3, type = 5, extra = 106)
```

[½]



[½]

(vi)
According to the decision tree, this customer is NOT expected to default [1]

(vii)

```
actual = read.csv("CS2B_Sept_21_Qu_3_Data.csv")                 [1]
head(actual)                                                    [½]
  f1         f2 default
1 -1  0.4842612       0
2  1  0.8991797       0
3  0 -0.2146289       0
4  1 -0.7641544       0
5 -1 -0.5532892       0
6  1 -0.9808636       1                                         [½]
```

(viii)

```
predict_defaults = predict(tree, actual, type = 'class')        [1]
head(predict_defaults)                                          [½]
1 2 3 4 5 6
0 1 0 1 0 1
Levels: 0 1                                                     [½]
```

(ix)

```
Confusion_matrix = table(actual$default,predict_defaults);
Confusion_matrix                                                [1½]
```

```
predict_defaults
   0  1
0 50 30
1  0 20                                                           [½]
```

(x)

```
precision = Confusion_matrix[2,2] / sum(Confusion_matrix[,2]);
precision                                                         [1½]
[1] 0.4                                                           [½]
recall = Confusion_matrix[2,2] / sum(Confusion_matrix[2,]);
recall                                                            [1½]
[1] 1                                                             [½]
```

The precision percentage is 40% [½]
and the recall percentage is 100% [½]

(xi)

The recall percentage is the percentage of defaults that the model managed to identify [½]
Here the model has performed well and has identified all 20 actual defaults [1]
However, the model is currently not very precise [½]
The precision percentage is the percentage of predicted defaults that are in fact actual defaults [½]
The model has predicted far more defaults than was actually the case [1]
Hence, if this model had been used to approve the loans of these 100 customers, 30% of them would have not been approved for a loan even though they did not actually default [1]
The model is therefore not commercially optimised [1]
This is in line with our conclusions from part (iv) i.e. that the probability of default in the specimen data used to train the model was unrealistically too high [1]
The Actuary could refine the parameters ($a$, $b$, $c$) used to construct the specimen data and re-train the model to improve it [1]
Better still, further data could be gathered from the public domain, if available, and used to train the model to improve it [1]
Additionally, other decision tree models (e.g. bagged decision trees, random forests, boosted decision trees) should be investigated to see if a better fit can be obtained [½]
Alternatively, other classification machine learning models (e.g. naïve Bayes classification) should be investigated to see if a better fit can be obtained [½]
Additionally, the Actuary could change the approach from classification of loan default/not default to a probability of loan default approach and use some form of regression machine learning algorithm to predict the probabilities [½]

[Marks available 10, maximum 6]

**[Total 44]**

*Part (i) was the best answered question part on the whole paper.*

*Part (ii) was well answered. Valid alternative methods, such as using ifelse statements to specify the vectors f1 and default, were awarded credit.*

*Part (iii) was poorly answered. Candidates who did not successfully complete part (ii) could still have gained credit for answering part (iii) based on "dummy" data. A valid alternative was to use default in place of defprob.*

*Part (iv) was very poorly answered overall. Most candidates who correctly obtained a probability close to 0.5 in part (iii) recognised that it was unrealistically high, but many did not suggest reducing the parameter a to make the probability more realistic.*

*Part (v) was fairly well answered. As the R code is provided in the question, candidates who did not include it in their answer scripts were not penalised.*

*Part (vi) was poorly answered overall. Most, but not all, candidates who successfully plotted a tree in part (v) were able to interpret it correctly.*

*Part (vii) was well answered. As the R code to import the data frame actual is provided in the question, candidates who did not include this code in their answer scripts were not penalised.*

*Part (viii) was fairly well answered.*

*Part (ix) was poorly answered. Candidates who generated the transpose of the required matrix did not receive full marks as the convention specified in the Core Reading is that the rows represent the actual outcomes and the columns represent the predicted outcomes. However, candidates who used a value of 1 to represent non-default rather than default were not penalised.*

*Part (x) was very poorly answered. Many candidates failed to gain credit as they::*
*Had the definitions of the precision and recall percentages the wrong way around.*
*Calculated the precision and recall percentages manually, whereas the question specifies that they should be calculated in R.*
*Did not quote their answers separately from the R output.*

*Part (xi) was extremely poorly answered overall. Of those candidates who successfully calculated the precision and recall percentages in part (x), many were able to interpret them correctly but very few suggested further investigations that should be carried out as a result. Alternative comments that were clear, distinct and relevant to the context of the question were also awarded credit.*

**[Paper Total 100]**

# END OF EXAMINERS' REPORT