INSTITUTE AND FACULTY OF ACTUARIES

EXAMINERS' REPORT

April 2022

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 Specialist Advanced (SA) and Specialist Principles (SP) 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
July 2022

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.*

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 57
1175 presented themselves and 452 passed.

Solutions for Subject CS2B - April 2022

Q1

(i)

```
set.seed(914)                                                    [½]
n = rpois(20000, 1000)                                           [1]
s = numeric(20000)                                               [1]
for(i in 1:20000)                                                [1]
{x = rgamma(n[i], shape = 750, rate = 0.25)                      [1]
s[i] = sum(x)}                                                   [1]
head(s, 7)                                                       [1]
[1] 2860469 2915250 2998362 3223837 2915546 2971731 3132371
                                                                 [½]
```

ALTERNATIVE SOLUTION:

```
set.seed(914)                                                    [½]
s = numeric(20000)                                               [1]
for(i in 1:20000)                                                [1]
{n = rpois(1, 1000)                                              [1]
x = rgamma(n, shape = 750, rate = 0.25)                          [1]
s[i] = sum(x)}                                                   [1]
head(s, 7)                                                       [1]
[1] 2856968 2929369 2910782 2941784 3041930 3057008 2953528
                                                                 [½]
```

(ii)

```
mean(s)                                                          [½]
[1] 3000982                                                      [½]
```

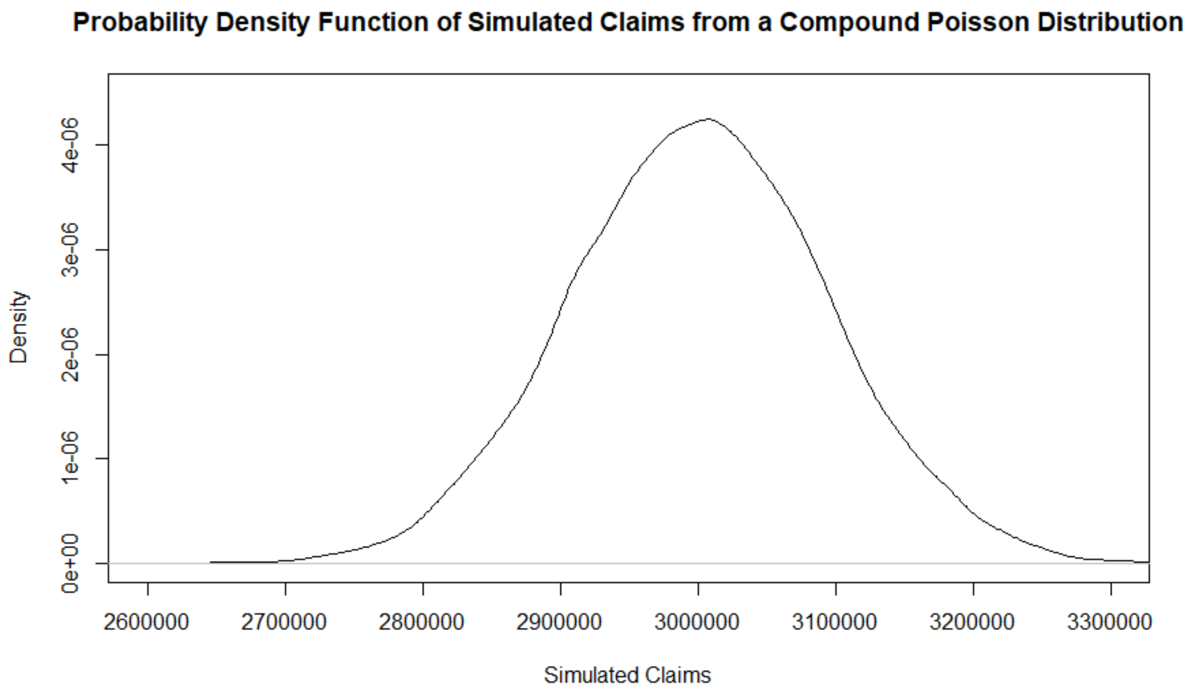
The mean of the simulated claim values is 3,000,982 [½]

```
sd(s)                                                            [½]
[1] 93872.61                                                     [½]
```

The standard deviation of the simulated claim values is 93,872.61 [½]

(iii)

```
plot(                                                            [½]
density(s),                                                      [1]
xlim = c(2600000, 3300000),                                      [1]
ylim = c(0, 4.5e-06),                                            [1]
xlab = "Simulated Claims",                                       [½]
main = "Probability Density Function of Simulated Claims
from a Compound Poisson Distribution")                           [½]
```



[½]

(iv)

Normal distribution [1]

with mean = 3,000,982 … [½]

and standard deviation = 93,872.61

OR

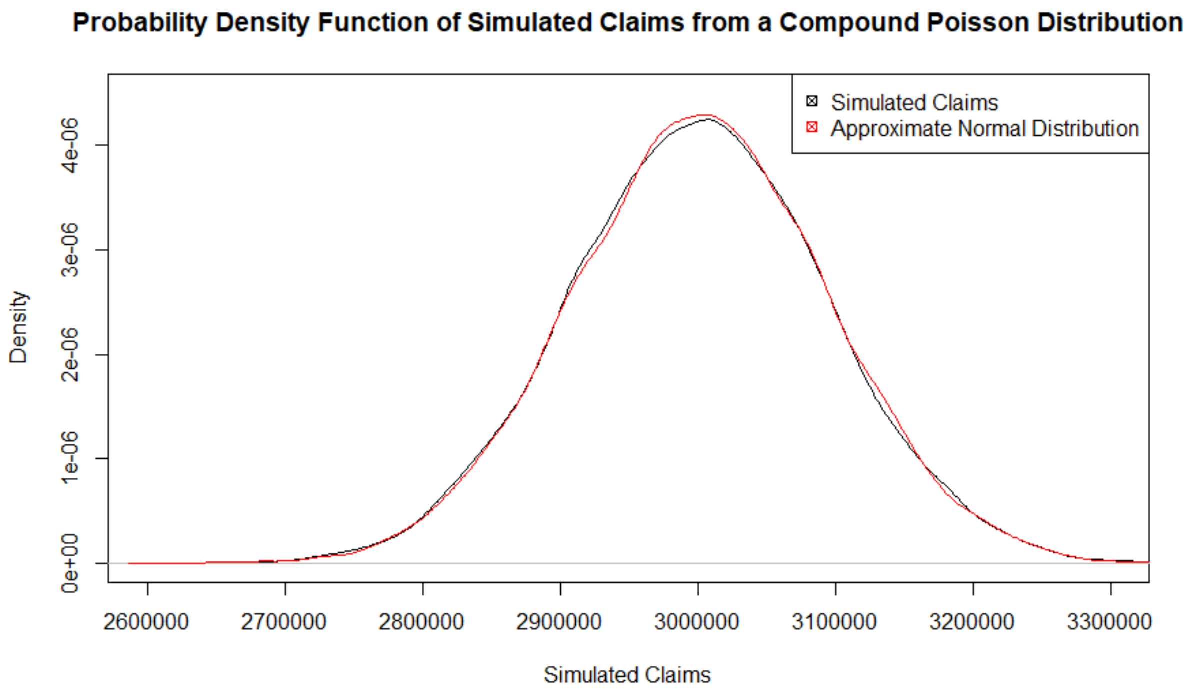
and variance = 93,872.61^2 = 8,812,067,277 [½]

(v)

```
set.seed(914)                                    [½]
approx_dist = rnorm(20000, mean(s), sd(s))       [1]
head(approx_dist, 7)                             [1]
[1] 2856910 3082046 2903046 2999750 3229462 2919827 2976414
                                                 [½]
```

(vi)

```
lines(                                           [½]
density(approx_dist),                            [1]
 col = "red")                                    [1]
legend("topright",
legend = c("Simulated Claims", "Approximate Normal
Distribution"),                                  [½]
  col = c("black","red"),                        [½]
  pch = 7)
```



[½]

[Total 24]

In Part (i), candidates performed well.

Part (ii) was fairly well answered. Candidates are only awarded full marks if they quote their answer separately from their R output. Candidates using the alternative solution in part (i) should have obtained a mean of 3,000,283 and a standard deviation of 95,263.4.

Part (iii) was fairly well answered, although many candidates were not awarded full marks for not using sufficiently specific titles and/or x-axis labels. Titles were awarded credit provided they referred ***both*** *to Density* ***and*** *to Claims, and x-axis labels were awarded credit provided they referred to Claims.*

Part (iv) was fairly well answered. Candidates who estimated the mean and standard deviation by eye from the graph in part (iii), rather than using their answers to part (ii), were not awarded full marks as their estimates of the standard deviation were not sufficiently accurate.

Part (v) was fairly well answered. The most common error was to use R code similar to that in part (i) with the gamma distribution replaced by a normal distribution, which is not correct since the normal distribution represents the ***aggregate*** *claims.*

Answers to part (vi) were generally satisfactory. The most common areas where candidates did not receive marks were:

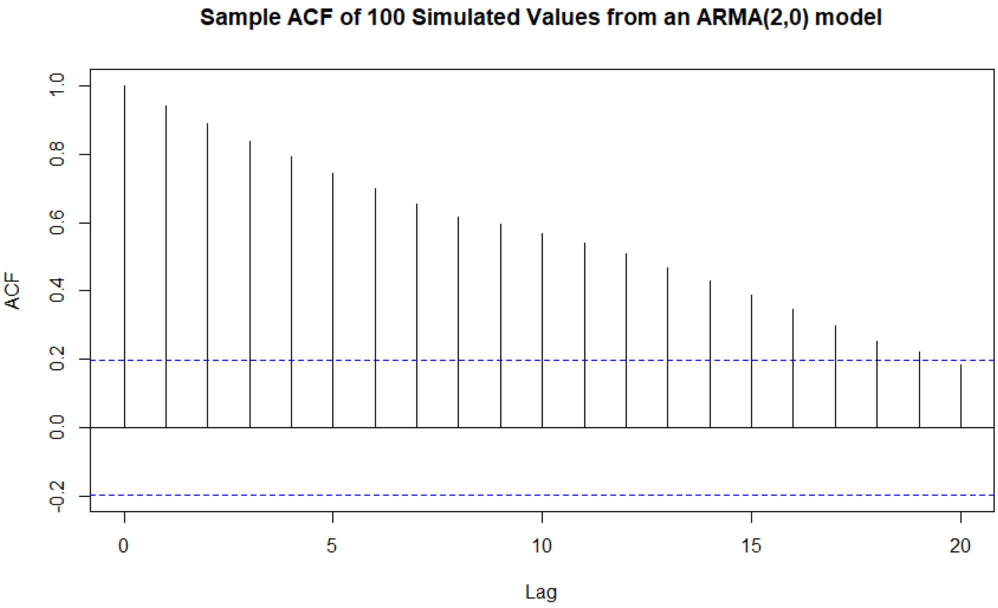
- *Plotting the theoretical density function of the normal distribution, rather than the empirical density function of the simulated values in part (v) as per the question.*

Q2

```
set.seed(12456)
x = arima.sim(n = 100, model = list(ar = c(0.7,0.2)))
```

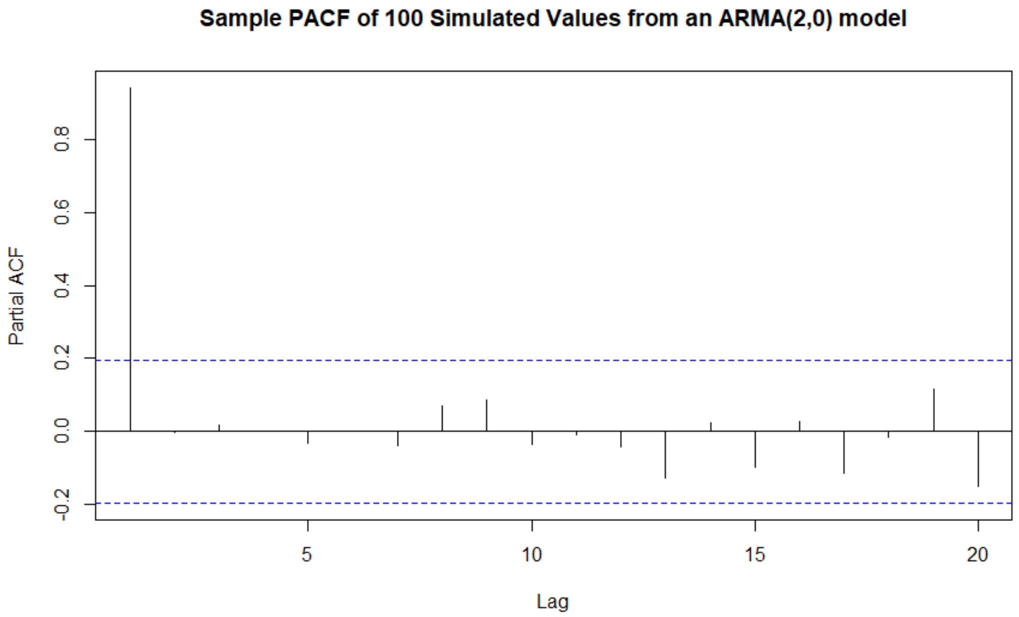
(i)

```
acf(x,                                                        [½]
main = "Sample ACF of 100 Simulated Values from an ARMA(2,0)
model")                                                       [½]
```



[½]

```
pacf(x,                                                       [½]
main ="Sample PACF of 100 Simulated Values from an
ARMA(2,0) model”)                                             [½]
```



[½]

(ii)
EITHER:
The ACF is decreasing (exponentially) to zero [½]
which is in line with the theoretical behaviour of an ARMA(2,0) process [½]

OR:
The ACF decreases to zero more slowly than might be expected from an ARMA(2,0) process [1]

THEN:
But the PACF seems to cut off after lag 1. [½]
We might expect the PACF of an ARMA(2,0) process to cut off after lag 2 [½]
However, given the small sample size and the fact that this process is close to being non-stationary, the two plots are not out-of-line with expectations. [1]
[Marks available 3, maximum 2]

(iii)

```
fitx10 = arima(x, order = c(1,0,0))                    [1]
fitx10$aic                                             [½]
[1] 305.4467                                           [½]
fitx11 = arima(x, order = c(1,0,1)); fitx11$aic        [½]
[1] 307.4325                                           [½]
fitx20 = arima(x, order = c(2,0,0)); fitx20$aic        [½]
[1] 307.4309                                           [½]
```

(iv)
The preferred model here is ARMA(1,0) [1]
as the AIC value is the lowest [1]

(v)

We might have expected the ARMA(2,0) model to be the best fit as that is the "true" model that *x* was generated from [1]
However, the number of observations of *x* is quite small [1]
and so sampling uncertainty has produced a better fitting / forecasting model that is not ARMA(2,0) [1]

(vi)
We could increase the value of *n* in the R code (to say 1,000) [1]
which would reduce the sampling uncertainty of the observations [1]

(vii)

```
set.seed(12456)                                             [½]
y = arima.sim(n = 1000, model = list(ar = c(0.7,0.2)))     [1½]
```

(viii)

```
fity10 = arima(y, order = c(1,0,0)); fity10$aic            [½]
[1] 2904.756                                                [½]
fity11 = arima(y, order = c(1,0,1)); fity11$aic            [½]
[1] 2861.953                                                [½]
fity20 = arima(y, order = c(2,0,0)); fity20$aic            [½]
[1] 2861.243                                                [½]
```

(ix)
The preferred model here is ARMA(2,0) as the AIC value is the lowest [1]

(x)

```
fitx10
Call:
arima(x = x, order = c(1, 0, 0))
Coefficients:
         ar1   intercept
       0.9530    -1.0632
s.e.   0.0279     1.9238

sigma^2 estimated as 1.142:  log likelihood = -149.72,
aic = 305.45                                                [½]

fity20
Call:
arima(x = y, order = c(2, 0, 0))
Coefficients:
         ar1     ar2  intercept
      0.7116  0.2112     0.2409
s.e.  0.0309  0.0309     0.4064

sigma^2 estimated as 1.014:  log likelihood = -1426.62,
aic = 2861.24                                               [½]
```

For data set *y*, the best fitting model is the "true" model [½]
Whereas the best fitting model for data set *x* is not the "true" model [½]

This is what we expected as the more observations we have the more likely it is that the "true" model is the best fit [½]
The estimated AR parameters for data set *y* are close to the real values of 0.7 and 0.2 [1]
The estimated error variance sigma^2 is closer to the real value of 1 for data set *y* than for data set *x*. [½]
The estimated intercept term is closer to the real value of zero for data set y than for data set x. [½]
These observations on the estimated error variance and intercept term are to be expected given that the data volume is higher in data set y than in data set x [½]
and given that the fitted model matches the true model for data set y but not for data set x [½]

[Marks available 5½, maximum 3]

[Total 25]

Part (i) was fairly well answered. Credit was only awarded for titles that included ***both*** *the model name, i.e. ARMA(2,0) or AR(2),* ***and*** *ACF or PACF as appropriate.*

Part (ii) was fairly well answered. To gain full marks candidates needed to comment ***both*** *on the key features of both the ACF and PACF graphs* ***and*** *on how they compare with the theoretical behaviour of both the ACF and PACF under an ARMA(2,0) process.*

Part (iii) and (iv) were very well answered in general.

In Part (v) well prepared candidates were able to generate appropriate comments by referring back to the earlier parts of the question and observing that the model identified as most appropriate in part (iv) was not the model the data set was generated from, and that the sample size was only 100. Alternative comments that were clear, distinct and relevant to the context of the question were also awarded credit.

In Part (vi), most candidates performed well as they recognised that the value of n should be increased, but many did not provide clear reasoning.

Parts (vii) and (viii) were very well answered.

Part (ix) was fairly well answered. Where candidates did not obtain the result that the ARMA(2,0) model had the lowest AIC in part (viii), they were not awarded a follow-through mark in part (ix) since they were asked in part (vi) to ensure that the ARMA(2,0) model became the best fitting model.

In Part (x) many candidates repeated comments they made earlier in the question, without referring back to the fact that the true model is ARMA(2,0). Some candidates made comments about the parameter values without including the R code required to output those values, which could not be awarded credit. Alternative comments that were clear, distinct and relevant to the context of the question were also awarded credit.

Q3
(i)

```
graduation =
read.csv(file = "CS2B_A22_Qu_3_Data.csv", head = TRUE)      [1]
graduation$zx1 = (graduation$Deaths - graduation$Exposure *
graduation$Graduation1) / sqrt(graduation$Exposure *
graduation$Graduation1)                                    [1½]

graduation$zx2 = (graduation$Deaths - graduation$Exposure *
graduation$Graduation2) / sqrt(graduation$Exposure *
graduation$Graduation2)                                     [1]

graduation$zx3 = (graduation$Deaths - graduation$Exposure *
graduation$Graduation3) / sqrt(graduation$Exposure *
graduation$Graduation3)                                     [1]

head(graduation[, 7:9], 7)                                  [1]

         zx1         zx2          zx3
1 -0.4532486  2.08008602  0.077224448
2 -0.7304714  1.21170927 -0.453050849
3  0.4301427  1.78813614  0.498299817
4 -0.1565084  0.57642875 -0.250408199
5  0.5517038  0.72246846  0.351202375
6  0.2811644 -0.07163144  0.002057777
7 -0.5559179 -1.34146800 -0.839474197                      [½]
```

(ii)
EITHER:

```
chisq = vector(length = 3)                                  [1]
chisq[1] = sum(graduation$zx1^2)                            [1]
chisq[2] = sum(graduation$zx2^2)                            [1]
chisq[3] = sum(graduation$zx3^2)                            [1]
df = c(36, 37, 35)                                          [1]
1 - pchisq(chisq, df = df)                                 [1½]
[1] 7.002971e-02 5.582300e-08 7.128381e-02                  [½]
```

OR:

```
chisq1 = sum(graduation$zx1^2)                              [1]
chisq2 = sum(graduation$zx2^2)                              [1]
chisq3 = sum(graduation$zx3^2)                              [1]
1 - pchisq(chisq1, df = 36)                                [1½]
[1] 0.07002971                                              [½]
1 - pchisq(chisq2, df = 37)                                 [½]
[1] 5.5823e-08                                              [½]
1 - pchisq(chisq3, df = 35)                                 [½]
[1] 0.07128381                                              [½]
```

THEN:
The *p*-value for graduation 1 is 0.07003
The *p*-value for graduation 2 is 5.582e-08

The *p*-value for graduation 3 is 0.07128 [1]

(iii)
Based on the *p*-values, there is enough evidence to reject graduation 2 at the 5% significance level but not graduations 1 and 3 [2]
Since graduation 2 does not fit the rates well, the parameter that was removed from graduation 1 to give this graduation is statistically significant and should be added back in [2]
The results do not support the need for the additional parameter in graduation 3 compared with graduation 1. [2]
For graduation 3 the improvement in fit approximately balances out the lost degree of freedom. [1]
Although graduations 1 and 3 appear suitable based on the chi-square test, they may be unsuitable for other reasons [1]
Any appropriate example, e.g. small consistent bias, a few large outliers, or clumps of deviations of the same sign [1]
[Marks available 9, maximum 6]

(iv)
EITHER:

```
pos = vector(length = 3)                                  [1]
neg = vector(length = 3)                                  [½]
pos[1] = length(graduation$zx1[graduation$zx1 > 0])       [1]
pos[2] = length(graduation$zx2[graduation$zx2 > 0])       [½]
pos[3] = length(graduation$zx3[graduation$zx3 > 0])       [½]
neg[1] = length(graduation$zx1[graduation$zx1 < 0])       [½]
neg[2] = length(graduation$zx2[graduation$zx2 < 0])       [½]
neg[3] = length(graduation$zx3[graduation$zx3 < 0])       [½]
pos                                                       [½]
[1] 22 20 21                                              [½]
neg                                                       [½]
[1] 19 21 20                                              [½]
```

OR:

```
pos1 = length(graduation$zx1[graduation$zx1 > 0])         [1]
pos2 = length(graduation$zx2[graduation$zx2 > 0])         [½]
pos3 = length(graduation$zx3[graduation$zx3 > 0])         [½]
neg1 = length(graduation$zx1[graduation$zx1 < 0])         [1]
neg2 = length(graduation$zx2[graduation$zx2 < 0])         [½]
neg3 = length(graduation$zx3[graduation$zx3 < 0]}         [½]
pos1
[1] 22                                                    [½]
pos2
[1] 20                                                    [½]
pos3
[1] 21                                                    [½]
neg1
[1] 19                                                    [½]
neg2
```

```
[1] 21                                                        [½]
neg3
[1] 20                                                        [½]
```

THEN:
Graduation 1 has 22 positive and 19 negative deviations.
Graduation 2 has 20 positive and 21 negative deviations.
Graduation 3 has 21 positive and 20 negative deviations. [1]

(v)
EITHER:

```
groups = vector(length = 3)                                   [1]
for(j in 1:3){                                                [1]
pos_z = (graduation[, j + 6] > 0) * 1                         [1]
groups[j] =                                                   [½]
sum(                                                          [1]
duplicated(c(which(pos_z == 1) - 1, which(pos_z == 0))) * 1)
                                                             [3½]
+ pos_z[1] * 1}                                               [2]
groups                                                        [½]
[1] 11  5 11                                                  [½]
```

OR:

```
pos_z = (graduation$zx1 > 0) * 1                              [1]
groups1 =                                                     [½]
sum(                                                          [1]
duplicated(c(which(pos_z == 1) - 1, which(pos_z == 0))) * 1)
                                                             [3½]
+ pos_z[1] * 1                                                [2]
pos_z = (graduation$zx2 > 0) * 1
groups2 =
sum(
duplicated(c(which(pos_z == 1) - 1, which(pos_z == 0))) * 1)
+ pos_z[1] * 1                                                [½]
pos_z = (graduation$zx3 > 0) * 1
groups3 =
sum(
duplicated(c(which(pos_z == 1) - 1, which(pos_z == 0))) * 1)
+ pos_z[1] * 1                                                [½]
groups1                                                       [½]
[1] 11                                                        [½]
groups2
[1] 5                                                         [½]
groups3
[1] 11                                                        [½]
```

THEN:
Graduation 1 has 11 groups of positive deviations.

Graduation 2 has 5 groups of positive deviations.
Graduation 3 has 11 groups of positive deviations. [1]

ALTERNATIVE SOLUTION:

EITHER:

```
groups = vector(length = 3)                                  [1]
for(j in 1:3){                                               [1]
pos_z = (graduation[, j + 6] > 0) * 1                        [1]
groups[j] =                                                  [½]
sum(                                                         [1]
duplicated(c(which(pos_z == 1) - 1,
which(pos_z == 0))) * 1)                                    [3½]
+ pos_z[1] * 1                                               [2]
}
groups                                                       [½]
[1] 11  5 11                                                 [½]
```

OR:

```
pos_z = (graduation$zx1 > 0) * 1                             [1]
groups1 =                                                    [½]
sum(                                                         [1]
duplicated(c(which(pos_z == 1) - 1,
which(pos_z == 0))) * 1)                                    [3½]
+ pos_z[1] * 1                                               [2]
pos_z = (graduation$zx2 > 0) * 1
groups2 =
sum(
duplicated(c(which(pos_z == 1) - 1,
which(pos_z == 0))) * 1)
+ pos_z[1] * 1                                               [½]
pos_z = (graduation$zx3 > 0) * 1
groups3 =
sum(
duplicated(c(which(pos_z == 1) - 1,
which(pos_z == 0))) * 1)
+ pos_z[1] * 1                                               [½]
groups1                                                      [½]
[1] 11                                                       [½]
groups2
[1] 5                                                        [½]
groups3
[1] 11                                                       [½]
```

THEN:

Graduation 1 has 11 groups of positive deviations.
Graduation 2 has 5 groups of positive deviations.
Graduation 3 has 11 groups of positive deviations. [1]

(vi)
EITHER:

```
pval = vector(length = 3)                                    [1]
for(j in 1:3){                                               [1]
pval[j] = 0                                                  [1]
for(t in 1:groups[j]){                                       [1]
pval[j] = pval[j] + choose(pos[j] - 1, t - 1) *
choose(neg[j] + 1, t) /
choose(pos[j] + neg[j], pos[j])}}                            [3]
pval                                                         [½]
[1] 0.6844862501 0.0004065953 0.6787210548                   [½]
```

OR:

```
pval1 = 0                                                    [1]
for(t in 1:groups1){                                         [1]
pval1 = pval1 + choose(pos1 - 1, t - 1)*
choose(neg1 + 1, t) /
choose(pos1 + neg1, pos1)}                                   [3]
pval2 = 0
for(t in 1:groups2){
pval2 = pval2 + choose(pos2 - 1, t - 1)*
choose(neg2 + 1, t) /
choose(pos2 + neg2, pos2)}                                   [½]
pval3 = 0
for(t in 1:groups3){
pval3 = pval3 + choose(pos3 - 1, t - 1)*
choose(neg3 + 1, t) /
choose(pos3 + neg3, pos3)}                                   [½]
pval1                                                        [½]
[1] 0.6844862501                                             [½]
pval2
[1] 0.0004065953                                             [½]
pval3
[1] 0.6787210548                                             [½]
```

THEN:
The *p*-value for graduation 1 is 0.6845
The p-value for graduation 2 is 0.0004066
The p-value for graduation 3 is 0.6787 [1]

(vii)
The conclusions in part (iii) remain unchanged [1]
as the *p*-values calculated under the grouping of signs test support the conclusions in part (iii) [1]

[Total 51]

Part (i) was very well answered by candidates.

In Part (ii) was fairly well answered. However, many candidates calculated 1 less the required p-values or calculated the test statistics and critical values instead of the p-values. These candidates were not penalised again in part (iii) if they interpreted their answers to part (ii) correctly.

Part (iii) most candidates drew the correct conclusions from the chi-square tests based on their answers to part (ii), but only well prepared candidates provided sufficient further comments or referred back to the information in the question that Graduation 2 was obtained by removing one parameter from the formula underlying Graduation 1 and Graduation 3 was obtained by adding one parameter. Alternative comments that were clear, distinct and relevant to the context of the question were also awarded credit.

Part (iv) was fairly well answered, although many candidates did not quote their answers separately from their R output. `sum(graduation$zx1 > 0)`*was a valid alternative to* `length(graduation$zx1[graduation$zx1 > 0])`*, and similarly for the other graduations and for the negative deviations.*

In Part (v) the marking schedule presents an elementary method in order to give an indication of how many marks should have been awarded for alternative solutions that were not entirely correct. However, the number of positive groups for Graduation 1 could have been determined as `sum(diff(c(-1, sign(graduation$zx1))) == 2)`*, and similarly for the other graduations.*

In Part (vi) the marking schedule presents an elementary method in order to give an indication of how many marks should have been awarded for alternative solutions that were not entirely correct. However, since the required p-values are cumulative probabilities from the hypergeometric distribution, they could have been calculated using the R function phyper. Amongst those candidates who used an explicit method to calculate the p-values, the most common error was to evaluate the formula containing the choose functions only for t equal to the number of positive groups, rather than summed over all values of t less than or equal to the number of positive groups.

Part (vii) candidates who were unable to calculate the p-values in part (vi) could still have gained full marks for this part by stating that the conclusions are unchanged from part (iii) if and only if the p-value is greater than 0.05 for Graduations 1 and 3 but less than 0.05 for Graduation 2.

[Paper Total 100]

END OF EXAMINERS' REPORT