INSTITUTE AND FACULTY OF ACTUARIES

EXAMINERS' REPORT

September 2021

CS1 – Actuarial Statistics
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 Actuarial Statistics subject is to provide a grounding in mathematical and statistical techniques that are of particular relevance to actuarial work.

In particular, the CS1B paper is a problem-based examination and focuses on the assessment of computer-based data analysis and statistical modelling skills.

For the CS1B exam candidates are expected to include the R code that they have used to obtain the answers, together with the main R output produced, such as charts or tables.

When a question requires a particular numerical answer or conclusion, this should be explicitly and clearly stated, separately from, and in addition to the R output that may contain the relevant numerical information.

Some of the questions in the examination paper accept alternative solutions from those presented in this report, or different ways in which the provided answer can be determined. In particular, there are variations of the R code presented here, which are valid and can produce the correct output. All mathematically and computationally valid solutions or answers received credit as appropriate.

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 questions where comments were required, valid comments that were different from those provided in the solutions also received full credit where appropriate.

In cases where a question is based on simulations, and no seed was specified, all numerical answers provided in this document are examples of possible results. The numerical values presented here will be different if the simulations are repeated.

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

Overall performance in CS1B was particularly satisfactory. Well prepared candidates were able to score highly.

Most candidates demonstrated sufficient knowledge of the key R commands required for the application of the statistical techniques involved in this subject.

The quality of the commentary given alongside the R output was not always strong. In some cases, the provided comments were vague (e.g. Question 5 (iii)) or insufficient (e.g. Question 5(vi)).

In certain parts of the exam paper, candidates provided answers to particular question parts while answering different parts of the question (e.g. in Question 4). Cross marking was used to give credit where appropriate.

C. Pass Mark

The Pass Mark for this exam was 58
1372 presented themselves and 578 passed.

Solutions for Subject CS1B – September 2021

Q1

Enter the data into R using the commands provided in the question

```
approval_before <- c(8, 6, 8, 7, 7, 4, 2, 10, 8, 7, 10, 8, 8, 9, 6)
approval_after <- c(5, 6, 2, 3, 4, 1, 4, 7, 4, 2, 10, 1, 5, 3, 7)
```

(i)

```
> mean(approval_before)                                [½]
[1] 7.2                                                [½]
> mean(approval_after)                                 [½]
[1] 4.266667                                           [½]
```

(ii)

Kendall's rank correlation coefficient will measure the strength of monotonic relationship between approval before and approval after results
It considers only the relative values of the bivariate data, and not their actual values
It doesn't assume linearity between the bivariate data [2]

(iii)

```
> cor(approval_before,approval_after,method="kendall")      [1]
[1] 0.1819457                                               [1]
```

(iv)

From part (i), the mean approval has fallen from before (7.2) to after (4.3) [1]

Kendall's Tau in part (iii) shows that there is weak correlation between the approval scores before and after [1]

From parts (i) and (iii), this suggests that generally, the 15 voters' approval for the political party has changed, i.e. approval has fallen (or, the shift in opinion has not been the same across those surveyed) [1]

[Total 9]

Very well answered.
In part (ii) most candidates referred to rank or relative values, but many failed to give a full answer.

Q2

```
set.seed(2021)
lambda = 5
```

```
samplesize = 180
```

(i)(a)

```
x = rexp(samplesize,lambda)
```

[2]

Alternative solution

```
x=-(1/5)*log(runif(samplesize))
```

(b)

```
m = median(x)
m
# 0. 1463199
```

[2]

(c)

```
mq = qexp(0.5,lambda)
mq
# 0.1386294
```

[2]

(d)

The sample median in part (b) is an estimator for the true median in part (c) based on the sample in part (a) [1]

The results in (b) and (c) are not equal but similar [1]

(ii)(a)

For the parameter of an exponential distribution the MLE is the inverse of the sample mean

```
l = 1/mean(x)
```

[1]

```
l
# 4. 669902
```

[1]

(b)

```
1/mean(rexp(samplesize,lambda))
```

[2]

```
# 5.256819
```

[1]

(c)

The estimated values are both close to the true value (lambda = 5) [1]

but not identical. The actual value of the estimator is a random variable and the estimated value depends on the actual sample [1]

[Total 15]

Very well answered.
In part (ii) (b) many candidates reset the simulation seed, which resulted in obtaining exactly the same answer as earlier in (ii) (a). This also had implications in answering part (ii) (c). This was not penalised here and full credit was given where appropriate.

Q3

(i)

This is a Binomial(900, 0.36) distribution. [1]

(ii)

```
n = 900; p = 0.36
q = 220 + 20*(0:10)                                          [1]
cdfb = pbinom(q, n, p)                                       [1]
cdfb
pb = diff(cdfb) # prob. for intervals under Binomial distribution
                                                             [2]
# or:         pb = cdfb[-1] - cdfb[-11]
round(pb,4)
0.0000 0.0000 0.0011 0.0496 0.3544 0.4687 0.1202 0.0058 0.0001 0.0000   [1]
```

Alternative solution:

```
k <- seq(220,400,by=20)
pb <- round(pbinom(k+20,900,0.36)-pbinom(k,900,0.36),4)
```

(iii)

```
cdfn = pnorm(q, mean=324, sd=14.4)                           [1]
pn = diff(cdfn)                                              [2]
# or:         pn = cdfn[-1] - cdfn[-11]
round(pn, 4)
0.0000 0.0000 0.0011 0.0467 0.3428 0.4761 0.1271 0.0062 0.0001 0.0000   [1]
```

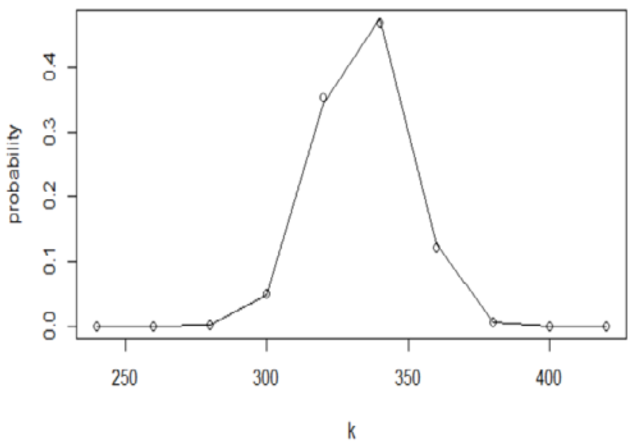
Alternative using continuity correction:

```
k <- seq(220,400,by=20)
pn <- round(pnorm(k+20.5, mean=324, sd=14.4) -pnorm(k+0.5,
mean=324, sd=14.4),4)
pn
[1] 0.0000 0.0000 0.0013 0.0501 0.3526 0.4701 0.1203 0.0056 0.0000 0.0000
```

(iv)

Plot both sets of probabilities on the same graph to comapre the values:

```
plot(q[-1], round(pb,4), type="p", xlab="k", ylab="probability")
# small circles show for Bin(n,p) prob.                      [1]
lines(q[-1], round(pn,4))                                    [1]
```



[2]

(v)
The plot shows that the suggested $N(324, 14.4^2)$ distribution is a very good approximation to the binomial distribution [1]
The reason for that is the CLT [1]
since the binomial distribution is the distribution of a sum of n independent random variables and $n = 900$ is large [1]
The approximation only works well since the parameters of the normal distribution have been chosen to match the first two moments of the normal distribution and the binomial distribution, $E[Y] = 900 \times 0.36 = 324$ [1]
and $Var(Y) = 900 \times 0.36 \times 0.64 = (30 \times 0.6 \times 0.8)^2 = 14.4^2$ [1]

[Total 19]

The quality of answers given here was mixed.
(i) Very well answered.
(ii) Very well answered, with some candidates failing to round the numbers.
(iii) Well answered. Again, some candidates failed to perform rounding. From those candidates who included the continuity correction, some used it in the wrong direction.
(iv) Common issues included unclear plots and wrong or insufficient annotation on the plot. A number of candidates did not attempt this part.
(v) Mixed answers. Some candidates mentioned the CLT, but not many noted the relevance of the choice of parameters.

Q4
Read the data into R
```
load("Claims_Experience.RData")
```

(i)
Fitting the GLM using the gaussian family gives:
```
>model_g <- glm(formula = Claim_number ~ Age + Region + Gender
+ Pre_existing_health_condition, data = data_claims, family
= gaussian())
```
[4]

The relevant information asked for in the question is:

```
> summary(model_g)
```

[1]

```
Coefficients:
                              Estimate   Std. Error  t value    Pr(>|t|)
(Intercept)                  -0.2430770  0.0181437 -13.397  < 2e-16 ***
Age                           0.0052998  0.0002653  19.974  < 2e-16 ***
RegionNorth                   0.0879968  0.0130929  6.721    2.01e-11 ***
RegionSouth                  -0.0356383  0.0124036  -2.873    0.00408 **
GenderMale                    0.0019173  0.0096233   0.199    0.84209
Pre_existing_health_condition 0.1108734  0.0096550  11.483   < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Null deviance: 654.57  on 4999  degrees of freedom
Residual deviance: 576.73  on 4994  degrees of freedom
AIC: 3404.3
```

[2]

(ii)(a)

Fitting the GLM using the Poisson family gives:

```
> model_p <- glm(formula = Claim_number ~ Age + Region +
Gender + Pre_existing_health_condition, data = data_claims,
family = poisson())
```

[1]

The relevant information asked for in the question is:

```
> summary(model_p)
```

[1]

```
 Coefficients:
                                Estimate     Std. Error   z value    Pr(>|z|)
 (Intercept)                    -8.450835    0.327926     -25.771    < 2e-16 ***
 Age                            0.080738     0.004234     19.068     < 2e-16 ***
 RegionNorth                    0.628536     0.122263     5.141      2.74e-07 ***
 RegionSouth                    -0.765895    0.155179     -4.936     7.99e-07 ***
 GenderMale                     0.018923     0.095022     0.199      0.842
 Pre_existing_health_condition  1.541474     0.133980     11.505     < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Null deviance: 2469.1  on 4999  degrees of freedom
Residual deviance: 1507.3  on 4994  degrees of freedom
AIC: 2252.4
```

[1]

(b)

`Model_g` (Gaussian) AIC is 3404.3 compared to `Model_p` (Poisson) AIC which is 2252.4 [1]

`Model_p` AIC lower than `Model_g`, therefore `Model_p` (Poisson) should be fitted and used [1]

Alternative:
based on *p*-values, there is little difference between the two models.

(c)
Using the Poisson model, the coefficient of pre_existing_health_condition is
positive with p-value roughly 2e-16 [1]
indicating significant positive dependence [1]

(iii)
Backward selection approach: start with the fully fitted GLM from part (i)(b),
i.e. model_p. Removing each of the categorical factors one by one:

```
> model_p2 <- glm(formula = Claim_number ~ Age + Region +
Pre_existing_health_condition, data = data_claims,
family = poisson())                                          [1]
```

Alternative answer: `> model_p2 <- update(model_p, ~.- Gender)`

```
> model_p3 <- glm(formula = Claim_number ~ Age + Region ,
data = data_claims, family = poisson())                      [1]
```

Alternative answer: `> model_p3 <- update(model_p2, ~.- Pre_existing_health_condition)`

```
> model_p4 <- glm(formula = Claim_number ~ Age, data =
data_claims, family = poisson())                             [1]
```

Alternative answer: `> model_p4 <- update(model_p3, ~.- Region)`

```
> summary(model_p2)
> summary(model_p3)
> summary(model_p4)
```

The AICs from the 4 models and the change from each step are as follows:

```
                                AIC     Change
 Model_p (baseline)             2252.4
 Model_p2                       2250    -2.4
 Model_p3                       2429    +176.6
 Model_p4                       2612    +359.6
```

[3]

(iv)(a)
Removing `Gender (Model_p2)` decreases the AIC by 2.4 [½]
Removing `Gender and Pre_exisiting_health_condition (Model_p3)` increases the AIC by 176.6 . [½]
Removing `Gender, Pre_exisiting_health_condition` and `Region (Model_p4)` increases the AIC by 359.6 [½]
This suggests the pre-existing health condition and region should be selected in this model [½]

(b)
With full backward selection, at each stage all variables should be removed one by one to decide which (if anyone) should not be included in the model [1]

(v)(a)

```
> model_p_refined <- glm(formula = Claim_number ~
Age*Pre_existing_health_condition*Region + Gender, data =
data_claims, family = poisson())                                [3]

> summary(model_p_refined)                                      [½]
```

```
 Coefficients:
                                              Estimate    Std. Error  z value  Pr(>|z|)
 (Intercept)                                  -15.264157  2.610953    -5.846   5.03e-09 ***
 Age                                          0.182569    0.034697    5.262    1.43e-07 ***
 Pre_existing_health_condition                7.654562    2.740791    2.793    0.00522 **
 RegionNorth                                  4.473053    3.073499    1.455    0.14557
 RegionSouth                                  -1.429017   5.034894    -0.284   0.77655
 GenderMale                                   0.012532    0.095140    0.132    0.89520
 Age:Pre_existing_health_condition            -0.094318   0.036748    -2.567   0.01027 *
 Age:RegionNorth                              -0.062227   0.041359    -1.505   0.13244
 Age:RegionSouth                              -0.005486   0.067297    -0.082   0.93503
 Pre_existing_health_condition:RegionNorth    -2.388733   3.205211    -0.745   0.45611
 Pre_existing_health_condition:RegionSouth    1.957076    5.150128    0.380    0.70394
 Age:Pre_existing_health_condition:RegionNorth 0.044085   0.043408    1.016    0.30981
 Age:Pre_existing_health_condition:RegionSouth -0.009280  0.069114    -0.134   0.89319
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Null deviance: 2469.1  on 4999  degrees of freedom
Residual deviance: 1464.6  on 4987  degrees of freedom
AIC: 2223.7                                                     [½]
```

(b)
The AIC for the refined model is 2223.7 which is lower than the AIC for the original model (model_p). [1]
Therefore, the refinement has improved the fit of the GLM model. [1]

(c)
The p-value for the interaction between age and pre-existing health condition is significant at the 5% level. [½]
Only the interaction between these two is associated with the number of claims [½]

[Total 30]

Generally well answered.
(i) Many candidates scored full marks. A common error was using 'Age' as the response variable.
(ii) Very well answered.

(iii) Well answered. Note that this question only requires the AIC output from the summary() command.
(iv) (a): A number of candidates chose the best model but failed to mention any variables that had been excluded. Credit was given as appropriate in cases where candidates gave a partial answer to (iv) (a) in (iii).
(v) (a) Proposed models often varied. Appropriate credit was awarded for partial attempts.

Q5

(i)

```
load("onebedflat.RData")
t.stat = t.test(City1, City2, var.equal=TRUE)$stat          [1½]

t.stat
-6.010694                                                    [½]
```

(ii)

The p-value of the test is:

```
(pt(-6.010694, 48))*2                                        [1]
```

So, the p-value is 2.414801e-07 . [1]

The p-value is less than 1%, [1]
therefore the flat prices are not the same for the two cities [1]

(iii)

```
set.seed(123) # Set seed to allow replication of answers     [1]
alpha = 0.1
m=20
n=20
N=8000
nrej=0
for (i in 1:N) {  # Run N = 8000 MC iterations                 [1]
 x=rnorm(m,mean=0,sd=1); y=rnorm(n,mean=0,sd=1) # Simulate
2 random samples of size 20 from N(0,1), i.e with same mean and variance   [1]
ts=t.test(x, y, var.equal=TRUE)$stat # Compute the test statistic
under assumption of equal variances                          [1]
df = n+m-2  # Determine degrees of freedom                   [1]
if (abs(ts)>=qt(1-alpha/2, df)) # Determine if H0 should be rejected   [1]
nrej=nrej+1 # Count number of rejections                     [1]
}
alpha_est=nrej/N # Compute estimated significance level as proportion of
rejections when H0 is true                                   [1]
```

(iv)

```
set.seed(123)
alpha = 0.1
```

```
m=20
n=20
N=8000
nrej=0
for (i in 1:N) {
x=rnorm(m,mean=0,sd=1); y=rnorm(n,mean=0,sd=1)
ts=t.test(x, y, var.equal=TRUE)$stat
df = n+m-2
if (abs(ts)>=qt(1-alpha/2, df))
nrej=nrej+1
}
alpha_est=nrej/N
alpha_est
0.099                                                       [1]
```

(v)(a)

```
set.seed(123)
alpha = 0.1
m=20
n=20
N=8000
nrej=0
for (i in 1:N) {
x=rnorm(m,mean=0,sd=1); y=rnorm(n,mean=0,sd=10)             [2]
ts=t.test(x, y, var.equal=TRUE)$stat
df = n+m-2
if (abs(ts)>=qt(1-alpha/2, df))
nrej=nrej+1
}
alpha_est1=nrej/N

alpha_est1
0.1091                                                      [1]
```

(b)

```
set.seed(123)
alpha = 0.1
m=20
n=20
N=8000
nrej=0
for (i in 1:N) {
x=rnorm(m,mean=10,sd=2); y= rexp(n,rate=0.1)                [4]
ts=t.test(x, y, var.equal=TRUE)$stat
df = n+m-2
if (abs(ts)>=qt(1-alpha/2, df))
nrej=nrej+1
}
alpha_est2=nrej/N
```

```
alpha_est2
0.12875
```
[1]

(vi)
The absolute differences between the true level and the estimated levels are 0.001, 0.0091, 0.0287 respectively for the cases (iv), (v)(a) and (v)(b) [2]

Violating the assumption of same variance did not bring as large deviation from the true level as when the assumption of normal distribution is violated [2]

[Total 27]

Well answered overall.
(i) and (ii) A common error was that 'var.equal=TRUE' was omitted.
(ii) Some candidates failed to state the conclusion of the test, instead pasting only the R output.
(iii) Responses varied here with some answers being unclear or vague.
(v) A common error was using a rate 10, instead of 1/10.
(vi) Comments often contained insufficient detail.

[Paper Total 100]

END OF EXAMINERS' REPORT