

INSTITUTE OF ACTUARIES OF INDIA

Subject CS1B – Actuarial Statistics (Paper B)

May 2026 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 examiners have given credit for any alternative approach or interpretation which they consider to be reasonable.

Solution 1: i)**Code:**

```

matches <- data.frame(
  Season1 = c(14, 16, 18),
  Season2 = c(15, 17, 19),
  Season3 = c(16, 18, 20),
  Season4 = c(14, 17, 19)
)

runs <- data.frame(
  Season1 = c(420, 640, 720),
  Season2 = c(450, 680, 760),
  Season3 = c(500, 720, 800),
  Season4 = c(430, 660, 750)
)

n <- ncol(runs)
N <- nrow(runs)

X <- runs / matches
Xibar <- rowSums(runs) / rowSums(matches)

Pibar <- rowSums(matches)
Pbar <- sum(Pibar)

Pstar <- sum(Pibar * (1 - Pibar / Pbar)) / (N * n - 1)

m <- sum(runs) / Pbar

s <- mean(rowSums(matches * (X - Xibar)^2) / (n - 1))

v <- (sum(rowSums(matches * (X - m)^2)) / (n * N - 1) - s) / Pstar

X

```

Output:

```

> X
  Season1 Season2 Season3 Season4
1    30    30 31.25 30.71429
2    40    40 40.00 38.82353
3    40    40 40.00 39.47368

```

[1]**ii)****Code:**

```

m
s
v

```

Output:

```

> m

```

```
[1] 37.0936
> s
[1] 4.275894
> v
[1] 26.73528
```

```
E[m(θ)], = m = 37.0936
E[s2 (θ)] = s = 4.275894
Var[m(θ)] = v = 26.73528
```

[3]

iii)

Code:

```
Zi = Pibar/(Pibar+s/v)
Zi
```

Output:

```
[1] 0.9972966 0.9976535 0.9979000
```

[2]

iv)

Code:

```
Prem=Zi*Xibar + (1-Zi)*m
Prem
```

```
Runs=Prem*15
Runs
```

Output:

```
> Runs
[1] 457.8942 595.4963 597.9389
```

[3]

v) As the credibility factors for all the players are **very close to 1**, we are giving **more importance to direct data** than the collateral data while calculating risk premiums.

[1]

vi)

Code:

We will just make the number of matches for each player equal to 1 and use the same code as in EBCT Model 2.

```

matches <- data.frame(
  Season1 = c(1, 1, 1),
  Season2 = c(1, 1, 1),
  Season3 = c(1, 1, 1),
  Season4 = c(1, 1, 1)
)

runs <- data.frame(
  Season1 = c(420, 640, 720),
  Season2 = c(450, 680, 760),
  Season3 = c(500, 720, 800),
  Season4 = c(430, 660, 750)
)

n <- ncol(runs)
N <- nrow(runs)

X <- runs / matches
Xibar <- rowSums(runs) / rowSums(matches)

Pibar <- rowSums(matches)
Pbar <- sum(Pibar)

Pstar <- sum(Pibar * (1 - Pibar / Pbar)) / (N * n - 1)

m <- sum(runs) / Pbar

s <- mean(rowSums(matches * (X - Xibar)^2) / (n - 1))

v <- (sum(rowSums(matches * (X - m)^2)) / (n * N - 1) - s) / Pstar

Zi = Pibar/(Pibar+s/v)
Zi

```

Alternative Code:

```

n1<-ncol(runs)
m1<- mean(rowMeans(runs))
s1<-mean(apply(runs,1,var))
v1<-var(rowMeans(runs))-mean(apply(runs,1,var))/n1
z1<-n1/(n1+s1/v1)
Prem1<-z1*rowMeans(runs)+(1-z1)*m1

z1

```

Output:

```

> Zi
[1] 0.9884037 0.9884037 0.9884037

```

So, the credibility factor under the assumptions of EBCT Model 1 is 0.9884.

[4]

vii)

Code:

$$\text{Prem} = Z_i \cdot \text{Xibar} + (1 - Z_i) \cdot m$$

Prem

Output:

> Prem

[1] 452.0584 674.4492 755.9925

[3]

viii)

There are significant differences in the runs to be scored especially for Virat and Shivam. Under EBCT Model 1, the estimated runs to be scored by Virat and Shivam are much higher than the estimated runs scored as per EBCT Model 2.

From part (i), we know that the batting averages of Virat, Rohit and Shivam are close to 30, 40 and 40 respectively. So, under EBCT Model 2 when all players are assumed to play the same number of matches (i.e. 15), the estimated runs scored for Rohit and Shivam are very close and also higher than the estimated runs for Virat.

However, EBCT Model 1, projects highest number of runs for Shivam, followed by Virat and then Rohit. This is because across all seasons the matches played by Shivam is higher than Virat and the matches played by Virat is higher than the matches played by Rohit. The trend relating to number of matches played observed in all the past seasons may not be repeated in the fifth season.

When all players are given a level playing field (same number of matches to be played in the fifth season), the estimated runs are reasonable and in line with the batting averages of the respective players. Hence, EBCT Model 2 seems to be more appropriate in this case.

[3]

[20 marks]

Solution 2:

i)

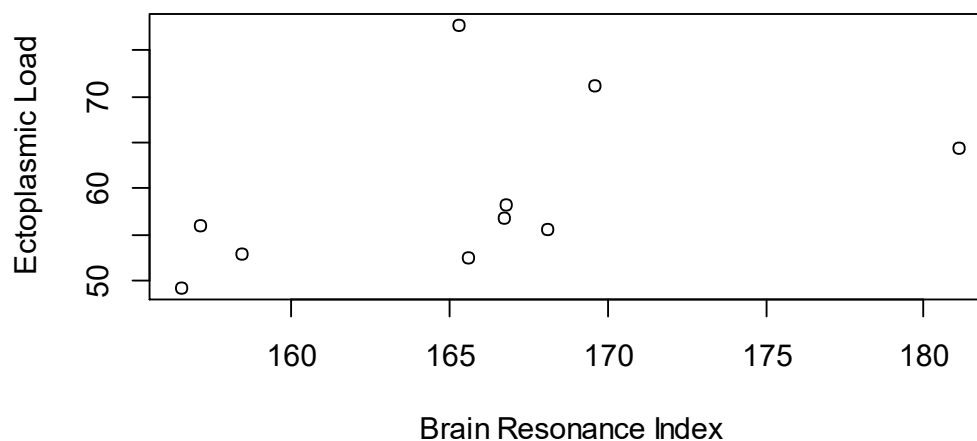
Code:

```
BRI <- c(169.6,166.8,157.1,181.1,158.4,165.6,166.7,156.5,168.1,165.3)
```

```
EL <- c(71.2,58.2,56.0,64.5,53.0,52.4,56.8,49.2,55.6,77.8)
```

```
plot (BRI, EL, xlab="Brain Resonance Index", ylab="Ectoplasmic Load",  
main="Scatterplot of BRI v EL")
```

Plot:

Scatterplot of BRI v EL**Comment:**

The scatterplot shows a weak positive association between brain resonance index and ectoplasmic load.

[3]**ii)****Code:**

```
cor(BRI,EL,method="pearson")
```

Output:

```
[1] 0.4674602
```

[2]**iii)****Code:**

```
model<-lm(EL~BRI)
summary(model)
```

Output:**Call:**

```
lm(formula = EL ~ BRI)
```

Residuals:

```
   Min     1Q   Median     3Q    Max
-7.1166 -4.7744 -2.8412  0.5696 18.4581
```

Coefficients:

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -36.8759   64.4728  -0.572   0.583
BRI           0.5821    0.3892   1.496   0.173
```

Residual standard error: 8.456 on 8 degrees of freedom

Multiple R-squared: 0.2185, Adjusted R-squared: 0.1208

F-statistic: 2.237 on 1 and 8 DF, p-value: 0.1731

From the summary of the regression model as generated above,

Estimated Standard Error of Intercept Parameter = 64.4728

Estimated Standard Error of Slope Parameter = 0.3892

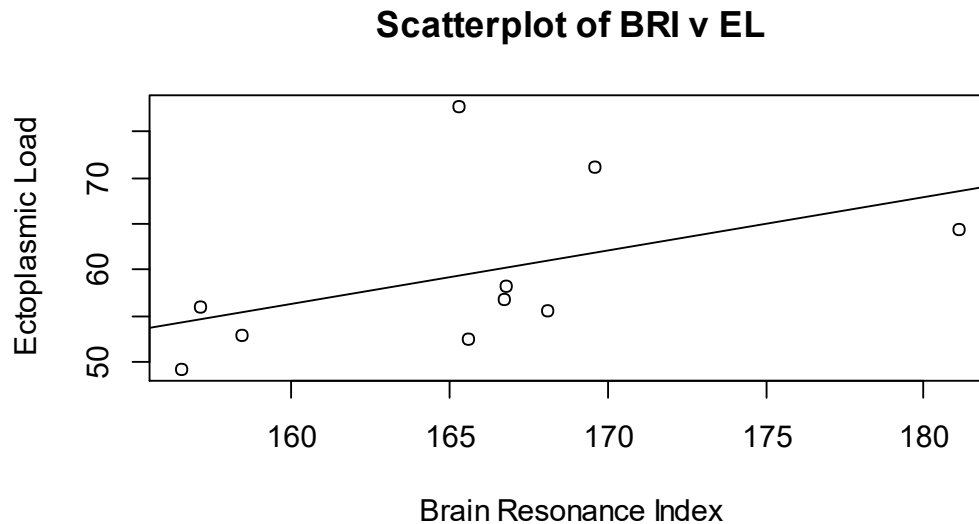
[4]

iv)

Code:

```
abline(model)
```

Plot:



[2]

v)

Code:

```
predict(model, data.frame(BRI=160.4))
```

Output:

```
1
56.48975
```

For a person with brain resonance index of 160.40, the predicted ectoplasmic load is 56.48975.

[2]

vi)

Code:

```
vcov(model)
```

Output:

```
(Intercept)    BRI
(Intercept) 4156.74194 -25.0700325
BRI        -25.07003  0.1514623
```

Covariance between the slope and intercept parameters = -25.0700325

[2]

vii)

H_0 : Slope parameter $\beta = 0$

H_1 : Slope parameter $\beta < 0$

Code:

```
confint(model, level=0.95)
```

Output:

```

      2.5 %    97.5 %
(Intercept) -185.5504257 111.798661
BRI         -0.3153742  1.479534

```

So, 95% confidence interval for the slope parameter is (-0.3154, 1.4795)

Conclusion:

As the 95% confidence interval includes the value 0, we have **no sufficient evidence to reject** the null hypothesis at 5% level of significance.

[4]

viii)

H_0 : population correlation coefficient $\rho = 0$
 H_1 : population correlation coefficient $\rho \neq 0$

Code:

```
cor.test(BRI,EL)
```

Output:

```

      Pearson's product-moment correlation

data: BRI and EL
t = 1.4957, df = 8, p-value = 0.1731
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.2298029  0.8476127
sample estimates:
      cor
0.4674602

```

Conclusion:

As the p-value is $0.1731 > 5\%$, we do not have sufficient evidence to reject the null hypothesis at the 5% level of significance.

Alternatively, the 95% confidence interval for the population correlation coefficient i.e. (-0.23, 0.85) contains the value 0 and hence we do not have sufficient evidence to reject the null hypothesis at the 5% level of significance.

[4]

ix)

The statement is correct.

If the beta coefficient is equal to 0, it is likely that the two variables are uncorrelated.

$$\text{Beta} = S_{xy} / S_{xx}$$

$$r = S_{xy} / \sqrt{S_{xx} * S_{yy}}.$$

Beta coefficient equal to 0 implies S_{xy} will be equal to 0 and hence the coefficient of correlation would in turn be equal to 0.

[2]

[25 marks]

Solution 3: i)**Code:**

```
motorclaims<-read.csv("C:\\Users\\Desktop\\MotorClaims.csv")

motorclaims[c("Sex","Region","Car_Type","Job")]<-
lapply(motorclaims[c("Sex","Region","Car_Type","Job")], factor)
```

[2]**ii)****Code:**

```
model_1<-glm(Claims ~ Sex + Region + Job + Car_Type, family =
poisson(link = "log"), data = MotorClaims)
summary(model_1)
```

Output:

Call:

```
glm(formula = Claims ~ Sex + Region + Job + Car_Type, family =
poisson(link = "log"),
    data = MotorClaims)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-5.6186	-1.5503	0.0589	1.4143	5.1168

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.768769	0.046900	80.357	< 2e-16 ***
Sex2	0.105003	0.029127	3.605	0.000312 ***
Region2	0.262172	0.038383	6.830	8.47e-12 ***
Region3	0.492846	0.036624	13.457	< 2e-16 ***
Job2	0.005805	0.035915	0.162	0.871606
Job3	0.051067	0.035517	1.438	0.150487
Car_Type2	0.417802	0.038812	10.765	< 2e-16 ***
Car_Type3	0.574171	0.037680	15.238	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 778.52 on 53 degrees of freedom
Residual deviance: 325.63 on 46 degrees of freedom
AIC: 677.33

Number of Fisher Scoring iterations: 4

Alternatively,

Call:

```
glm(formula = Claims ~ Sex + Region + Car_Type + Job, family = poisson,
    data = motorclaims)
```

Coefficients:

```

      Estimate Std. Error z value Pr(>|z|)
(Intercept) 3.17801    0.08107 39.200 < 2e-16 ***
Sex          0.10500    0.02913  3.605 0.000312 ***
Region       0.24513    0.01808 13.558 < 2e-16 ***
Car_Type     0.27432    0.01815 15.117 < 2e-16 ***
Job          0.02570    0.01781  1.443 0.149112
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 778.52 on 53 degrees of freedom
 Residual deviance: 344.01 on 49 degrees of freedom
 AIC: 689.71

Number of Fisher Scoring iterations: 4

[4]

iii)

We can see that the parameters Job2, and Job3 are insignificant at 5% significance level, because their p-values are greater than 0.05.

Whereas Sex2, Region2, Region3, Car_type2 and Car_type3 are significant at 5% level.

Alternatively, if the candidate has not converted the variables into factors, "job" will be insignificant at 5% level of significance as p-value is greater than 0.05.

Other variables like Region, Sex and Car have p-value of less than 1% and hence all other variables will be significant.

[2]

iv)

Code:

```

new_driver <- data.frame(Sex = "1", Region = "1", Car_Type = "2", Job = "3")
expected_claims <- predict(model_1, newdata = new_driver, type = "response")
expected_claims

```

Output:

```

1
69.24405

```

[3]

Alternatively, if someone calculates the value using the fitted Poisson model in part (ii), full credit is to be given even if R is not used to predict the value.

Under the Poisson GLM with log link, the expected number of claims is $\exp(3.7688 + 0.0511 + 0.4178) \approx 69.3$, for a male driver in an urban area driving a sedan with high job-related exposure.

Alternatively, if the candidate has not converted the variables into factors, then the expected number of claims would be 63.68.

Consider $\exp(3.17801 + 1 \cdot 0.105 + 1 \cdot 0.24513 + 2 \cdot 0.27432 + 3 \cdot 0.0257) = 63.68$.

v)

Code:

```
model_2 <- glm(Claims ~ Region + Job + Car_Type, family = poisson(link =
"log"), data = MotorClaims)
```

```
summary(model_2)
```

Output:

Call:

```
glm(formula = Claims ~ Region + Job + Car_Type, family = poisson(link =
"log"),
    data = MotorClaims)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-5.5054	-1.6675	-0.0976	1.2968	5.6946

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.822648	0.044325	86.241	< 2e-16 ***
Region2	0.262172	0.038383	6.830	8.47e-12 ***
Region3	0.492846	0.036624	13.457	< 2e-16 ***
Job2	0.005805	0.035915	0.162	0.872
Job3	0.051067	0.035517	1.438	0.150
Car_Type2	0.417802	0.038812	10.765	< 2e-16 ***
Car_Type3	0.574171	0.037680	15.238	< 2e-16 ***

 Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 778.52 on 53 degrees of freedom
 Residual deviance: 338.64 on 47 degrees of freedom
 AIC: 688.34

Number of Fisher Scoring iterations: 4

Alternatively,

Call:

```
glm(formula = Claims ~ Region + Car_Type + Job, family = poisson,
    data = motorclaims)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.33689	0.06780	49.219	< 2e-16 ***
Region	0.24513	0.01808	13.558	< 2e-16 ***
Car_Type	0.27432	0.01815	15.117	< 2e-16 ***
Job	0.02570	0.01781	1.443	0.149

 Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 778.52 on 53 degrees of freedom
 Residual deviance: 357.03 on 50 degrees of freedom
 AIC: 700.73

Number of Fisher Scoring iterations: 4

[3]

vi)

Job can be removed as a rating factor since both levels have high p-values (> 0.05)

[1]

vii)

Code:

```
model_3 <- glm(Claims ~ Region + Car_Type, family = poisson(link = "log"), data = MotorClaims)
summary(model_3)
```

Output:

Call:

```
glm(formula = Claims ~ Region + Car_Type, family = poisson(link = "log"), data = MotorClaims)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-5.6166	-1.6225	-0.0551	1.2761	5.5439

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.84187	0.03911	98.24	< 2e-16 ***
Region2	0.26217	0.03838	6.83	8.47e-12 ***
Region3	0.49285	0.03662	13.46	< 2e-16 ***
Car_Type2	0.41780	0.03881	10.77	< 2e-16 ***
Car_Type3	0.57417	0.03768	15.24	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 778.52 on 53 degrees of freedom
 Residual deviance: 341.13 on 49 degrees of freedom
 AIC: 686.83

Number of Fisher Scoring iterations: 4

Alternatively,

Call:

```
glm(formula = Claims ~ Region + Car_Type, family = poisson, data = motorclaims)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.38851	0.05749	58.94	< 2e-16 ***
Region	0.24513	0.01808	13.56	< 2e-16 ***

```
Car_Type    0.27432    0.01815    15.12 <2e-16 ***
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for poisson family taken to be 1)

```
Null deviance: 778.52 on 53 degrees of freedom
Residual deviance: 359.11 on 51 degrees of freedom
AIC: 700.81
```

Number of Fisher Scoring iterations: 4

[3]

viii)

Code:

```
model_4 <- glm(Claims ~ Region * Car_Type, family = poisson(link = "log"),
data = MotorClaims)
summary(model_4)
```

Output:

Call:

```
glm(formula = Claims ~ Region * Car_Type, family = poisson(link = "log"),
    data = MotorClaims)
```

Deviance Residuals:

```
    Min      1Q  Median      3Q     Max
-5.7265 -1.2117  0.0901  1.4978  5.1145
```

Coefficients:

```
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    3.86073    0.05923  65.177 < 2e-16 ***
Region2         0.19971    0.07989   2.500  0.0124 *
Region3         0.49598    0.07514   6.601 4.08e-11 ***
Car_Type2       0.44783    0.07583   5.905 3.52e-09 ***
Car_Type3       0.50024    0.07508   6.663 2.68e-11 ***
Region2:Car_Type2 -0.06171    0.10285  -0.600  0.5485
Region3:Car_Type2 -0.02542    0.09638  -0.264  0.7920
Region2:Car_Type3  0.20006    0.09968   2.007  0.0447 *
Region3:Car_Type3  0.01570    0.09512   0.165  0.8689
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for poisson family taken to be 1)

```
Null deviance: 778.52 on 53 degrees of freedom
Residual deviance: 328.47 on 45 degrees of freedom
AIC: 682.17
```

Number of Fisher Scoring iterations: 4

Alternatively,

Call:

```
glm(formula = Claims ~ Region + Car_Type + Region:Car_Type, family =
poisson,
data = motorclaims)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.371280	0.121017	27.858	< 2e-16 ***
Region	0.253096	0.052429	4.827	1.38e-06 ***
Car_Type	0.282225	0.052056	5.422	5.91e-08 ***
Region:Car_Type	-0.003654	0.022563	-0.162	0.871

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 778.52 on 53 degrees of freedom
Residual deviance: 359.08 on 50 degrees of freedom
AIC: 702.78

Number of Fisher Scoring iterations: 4

[3]

ix)

From the QQ-plot it is clear that the model_4 may not be an adequate fit for the data.

The values at the beginning as well as towards the end are not lying on the qq-line, although in the middle the values are quite close to the qq-line.

[2]

x)

From the summaries of the various models, it is clear that:

- AIC for model_1 is 677.33
- AIC for model_2 is 688.34
- AIC for model_3 is 686.83
- AIC for model_4 is 682.17

Model_1 is ideally the best fit with the **lowest AIC** – but as sex cannot be used as a rating factor **due to regulations**, among the other three models, **Model_4** with the second lowest AIC appears to be a **better fit**.

Alternatively,

From the summaries of the various models, it is clear that:

- AIC for model_1 is 689.71
- AIC for model_2 is 700.73
- AIC for model_3 is 700.81
- AIC for model_4 is 702.78

Model_1 is ideally the best fit with the **lowest AIC** – but as sex cannot be used as a rating factor **due to regulations**, among the other three models, **Model_2** with the second lowest AIC appears to be a **better fit**.

[2]

Solution 4: i)**Code:**

```
x <- c(2,2,1,1,0,4,2,1,2,1,1,1,3,1,2,2,3,2,3,4)
```

```
mean(x)
```

```
var(x)
```

Output:

```
> mean(x)
```

```
[1] 1.9
```

```
> var(x)
```

```
[1] 1.147368
```

[2]

ii)**Code:**

```
lambda <- seq(1, 3, length = 1000)
```

[1]

iii)**Code:**

```
lik <- (lambda^(sum(x))*exp(-length(x)*lambda))/prod(factorial(x))
```

```
max_lik<-max(lik)
```

```
max_lik
```

```
lambda_lik<-lambda[which.max(lik)]
```

```
lambda_lik
```

Output:

```
> max_lik
```

```
[1] 7.7154e-14
```

```
> lambda_lik
```

```
[1] 1.900901
```

[2]

iv)**Code:**

```
loglik <- sum(x)*log(lambda) – length(x)*lambda - log(prod(factorial(x)))
```

```
max_loglik<-max(loglik)
```

```
max_loglik
```

```
lambda_loglik<-lambda[which.max(loglik)]
```

```
lambda_loglik
```

Output:

```
> max_loglik
```

```
[1] -30.19297
> lambda_loglik
[1] 1.900901
```

[3]

v)

Taking first derivative of the log-likelihood function and equating it to 0.

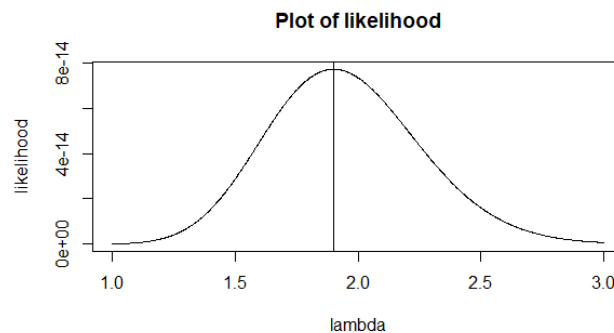
```
d/dλ (sum(x)*log(lambda) - n*lambda - log(prod(factorial(x)))) = 0
sum(x)/λ - n = 0
sum(x) / λ = n
λ = sum(x)/n = sample mean = 1.9
```

[2]

vi)

Code:

```
plot(lambda,lik, ylab="likelihood", type="l", main="Plot of likelihood")
abline(v=mean(x))
```

Plot:

[2]

vii)

It is clear from the graph that the likelihood function attains its maximum at $\lambda = 1.9$, which is the Maximum Likelihood Estimate (MLE) of the Poisson parameter as obtained in part (v).

The value of λ at which the likelihood function attains its maximum value as per part (iv) is 1.9. Hence, it is clear that $\lambda = 1.9$ is the maximum likelihood estimate of λ .

[2]

viii)

Code:

```
crlb<-mean(x)/length(x)
crlb

ci <- mean(x) + c(-1.96, 1.96) * sqrt(crlb)
ci
```

Output:

```
> crlb
[1] 0.095
```

```
> ci
[1] 1.295887 2.504113
```

95% CI for λ is given by [1.296,2.504]

[2]

ix)

Code:

```
poisson.test(sum(x), length(x), conf.level = 0.95)
```

Output:

Exact Poisson test

```
data: sum(x) time base: length(x)
number of events = 38, time base = 20, p-value = 0.000289
alternative hypothesis: true event rate is not equal to 1
95 percent confidence interval:
 1.344553 2.607898
sample estimates:
event rate
      1.9
```

So, the exact 95% confidence interval is [1.345, 2.608]

The exact confidence interval is **slightly wider and more accurate**, while the approximate interval is narrower due to normal approximation. Both intervals are close, indicating that the **normal approximation is reasonable** in this case.

[3]

x)

Code:

```
dpois(0,mean(x))
1-ppois(5,mean(x))
dpois(7,mean(x))
```

Output:

```
> dpois(0,mean(x))
[1] 0.1495686
> 1-ppois(5,mean(x))
[1] 0.01321918
> dpois(7,mean(x))
[1] 0.002652682
```

[3]

xi)

Code:

```
set.seed(2026)
```

```
samples <- matrix(NA, nrow = 100, ncol = 1000)
```

```
for (i in 1:1000) {
  samples[, i] <- rpois(100, lambda = mean(x))
}
```

```
}
```

Alternatively,

```
set.seed(2026)
```

```
samples <- replicate(1000, rpois(100, lambda = mean(x)))
```

[4]

xii)

Code:

```
sample_means <- colMeans(samples)
sample_vars <- apply(samples, 2, var)
```

```
mean(sample_means)
mean(sample_vars)
```

Output:

```
> mean(sample_means)
[1] 1.90717
> mean(sample_vars)
[1] 1.915354
```

Hence, both $E(\bar{X})$ and $E(S^2)$ based on the random samples are approximately equal to 1.9.

The maximum likelihood estimate of λ based on which the random samples have been generated is equal to 1.9.

Hence it is proved that $E(\bar{X}) = \lambda = \text{population mean}$ and $E(S^2) = \lambda = \text{population variance}$. (For a Poisson distribution, population mean and variance are equal to the parameter λ).

So, bias for both the estimators would be equal to 0 and it follows that $\bar{X}$ and S^2 are unbiased estimators of the population mean and population variance respectively.

[4]

[30 marks]
