Institute of Actuaries of India

Subject CS1-Actuarial Statistics (Paper B)

March 2022 Examination

INDICATIVE SOLUTION

Introduction
The indicative solution has been written by the Examiners with the aim of helping candidates. The solutions given are only indicative. It is realized that there could be other points as valid answers and examiner have given credit for any alternative approach or interpretation which they consider to be reasonable.

Solution 1:

i)

```
x= c(5,10,15,20,25,30,35,40,45,50,55,60)
y=c(15,12,25,23,35,36,33,38,43,45,50,53)

> meanx = mean(x)
> meany=mean(y)
> meanx
[1] 32.5
> meany
[1] 34

> x_sq=x*x
> x_sq
 [1]   25  100  225  400  625  900 1225 1600 2025 2500 3025 3600
> y_sq=y*y
> xy=x*y
> xy
 [1]   75  120  375  460  875 1080 1155 1520 1935 2250 2750 3180



> sumx_sq=sum(x_sq)
> sumy_sq=sum(y_sq)
> sumxy=sum(xy)

> sumx_sq
[1] 16250
> sumy_sq
[1] 15760
> sumxy
[1] 15775


> Sxx=Sumx_sq-12*meanx^2
> Sxx
[1] 3575
> Sxy=sumxy-12*meanx*meany
> Sxy
[1] 2515
> Syy=sumy_sq-12*meany^2
> Syy
[1] 1888
```

(7)

ii)

```
> beta=Sxy/Sxx
> beta
[1] 0.7034965


> alpha=meany-beta*meanx
> alpha
[1] 11.13636
```

```
> sigmasq=(1/(12-2))*(Syy-Sxy^2/Sxx)
> sigmasq
[1] 11.87063
```

(3)

iii)

```
> expectedy=alpha+beta*x
> expectedy
 [1] 14.65385 18.17133 21.68881 25.20629 28.72378 32.24126 35.75874 39.27622 42.79371
46.31119 49.82867 53.34615
```

(1)

iv)

```
> e=y-alpha-beta*x
> e
 [1]  0.3461538 -6.1713287  3.3111888 -2.2062937  6.2762238  3.7587413
 [7] -2.7587413 -1.2762238  0.2062937 -1.3111888  0.1713287 -0.3461538

> meane=mean(e)
> meane
[1] -1.702344e-15

> var(e)
[1] 10.79148
```

Mean value of residuals is close to zero as expected as e~N(0,sigma^2)

(Otherwise, “e” could be calculated as e = y-expectedy)

Var of e is slightly lower than sigma square as calculated in part ii – as denominator is not adjusted
When denominator of 10 gets used instead of 11 we see that var of residuals = sigma^2

```
> var(e)*11/10
[1] 11.87063
```

(3)

v) <u>95% confidence interval for beta</u>

Ho: Beta is zero (i.e. no linear relationship between x and y)
H1: Beta is not equal to zero

(Beta_cap-0)/ sqrt(sigma^2_cap/ Sxx) ~ t_{10}

We use t distribution with n-2 i.e. 10 degrees of freedom

```
> qt(p=0.025, lower.tail = T, df=10)
[1] -2.228139
```

Being symmetric distribution, 97.5% point would be 2.228139

```
> sqrt(sigmasq/Sxx)
[1] 0.0576234
```

Hence, endpoints of CI would be

```
> end1=beta+sqrt(sigmasq/Sxx)*qt(p=0.025, lower.tail = T, df=10)
> end1
[1] 0.5751036
```

```
> end2=beta-sqrt(sigmasq/Sxx)*qt(p=0.025, lower.tail = T, df=10)
> end2
[1] 0.8318894
```

Hence 95% Confidence interval for beta is (0.5751, 0.8319)

As confidence interval for beta does not include zero, we can reject null hypothesis (viz. beta=0) and Hence, can conclude that beta is not equal to zero at 5% level.

<u>95% CI for sigma^2</u>

(n-2)sigmacap^2/ sigma^2 ~ Chi sq distribution with 10 degrees of freedom

Tabulated values of Chi square having 10 df can be obtained as

```
> chitenend1=qchisq(df=10, p=0.025)
> chitenend2=qchisq(df=10,p=0.975)
> chitenend1
[1] 3.246973
> chitenend2
[1] 20.48318
```

End points of CI would be

```
> sigmasqend1=(12-2)*sigmasq/chitenend1
> sigmasqend2=(12-2)*sigmasq/chitenend2
> sigmasqend1
[1] 36.55907
> sigmasqend2
[1] 5.795307
```

Hence 95% Confidence interval for sigma^2 is (5.795,36.559)

(7)

vi)

SS_{TOT} = Syy = 1888 (as calculated in part i)

SS_{REG} = Sxy^2/Sxx

```
> ss_reg=Sxy^2/Sxx
> ss_reg
[1] 1769.294
```

$SS_{RES} = SS_{TOT} - SS_{REG}$

```
> ss_res=Syy-ss_reg
> ss_res
[1] 118.7063
```

R^2 denotes the % of variability explained by the model

R^2 = SS_{REG} / (SS_{REG} + SS_{RES})

```
> Rsq = ss_reg/(ss_reg+ss_res)
> Rsq
[1] 0.9371259
```

Model is a good fit as 93.7% of the variability is explained by the model.

```
> adj_Rsq = 1-((12-1)/(12-1-1))*(1-Rsq)
> adj_Rsq
[1] 0.9308385
```

Adjusted R^2 (93.08%) is lower than R^2 (93.71%) as adjusted R square penalises for extra predictors and
hence is better suited to assess the adequacy of the model (or for comparison between models) compared to just using R^2 for model comparison as
R^2 cannot decrease on addition of more explanatory variables which can be undesirable (as it may promote too many explanatory variables though not adding significant improvement in the predicted value)

(5)

vii)
using results from earlier parts mean predicted response is calculated (using regression line equation)

```
> Emean52=alpha+beta*52
> Emean52
[1] 47.71818
```

Expected value of mean predicted response is 47.718 when x=52

```
varofmean52=((1/12)+(52-meanx)^2/Sxx)*sigmasq
> varofmean52
[1] 2.251822

> mean52end1=Emean52+qt(p=0.025, lower.tail = T, df=10)*sqrt(varofmean52)
> mean52end2=Emean52-qt(0.025,lower.tail = T, df=10)*sqrt(varofmean52)
> mean52end1
[1] 44.37462
> mean52end2
[1] 51.06174
```

Hence 95% confidence interval for the mean predicted response is (44.3746,51.0617)

(4)
[30 Marks]

Solution 2:

i)

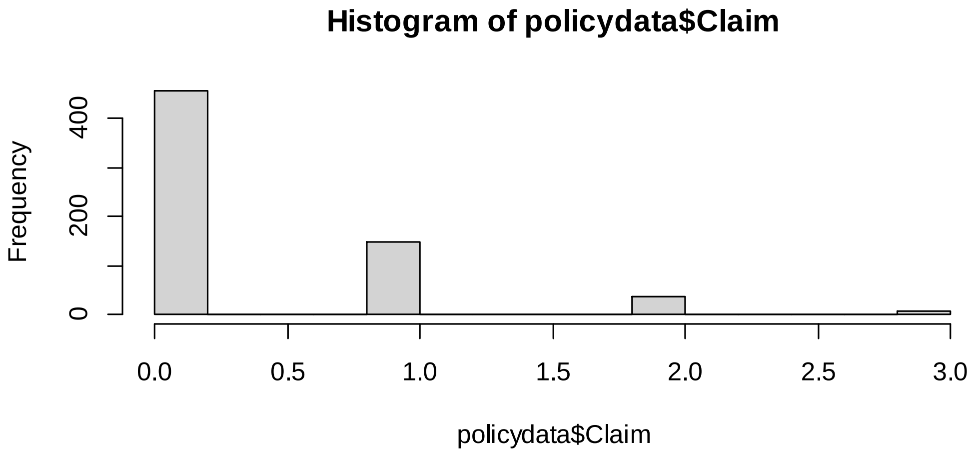
```
library(dplyr)

> str(policydata)
'data.frame':	650 obs. of  4 variables:
 $ Policy  : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Claim   : int  0 0 0 2 1 0 0 0 0 0 ...
 $ Cust_Exp: chr  "SA" "SA" "SA" "DS" ...
 $ Amount  : int  0 0 0 52601 56174 0 0 0 0 0 ...
>
> #a
> table(policydata$Claim)

  0   1   2   3 
458 149  36   7 
```

```
> #Alternative, if dplyr installed
> #count(policydata,Claim)

> print("458 Policies don't have any claim")
[1] "458 Policies don't have any claim"
```

(2)

ii)

```
> hist(policydata$Claim)
> #poisson and negative binomial distribution
```



(2)

iii)

```
> poisson.test(x=sum(policydata$Claim),T=length(policydata$Policy))

Exact Poisson test

data:  sum(policydata$Claim) time base: length(policydata$Policy)
number of events = 242, time base = 650, p-value < 2.2e-16
alternative hypothesis: true event rate is not equal to 1
95 percent confidence interval:
 0.3268739 0.4222903
sample estimates:
event rate
 0.3723077
> #0.35 is more suitable value of parameter since it lies between confidence interval.
```

(3)

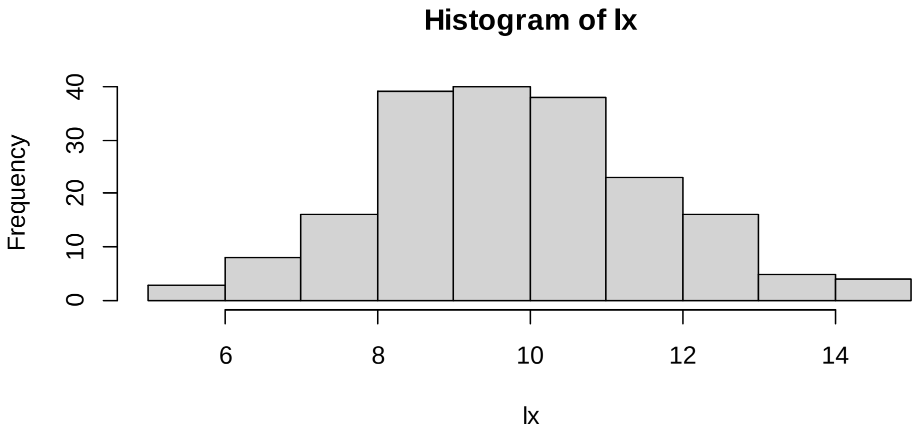
iv)

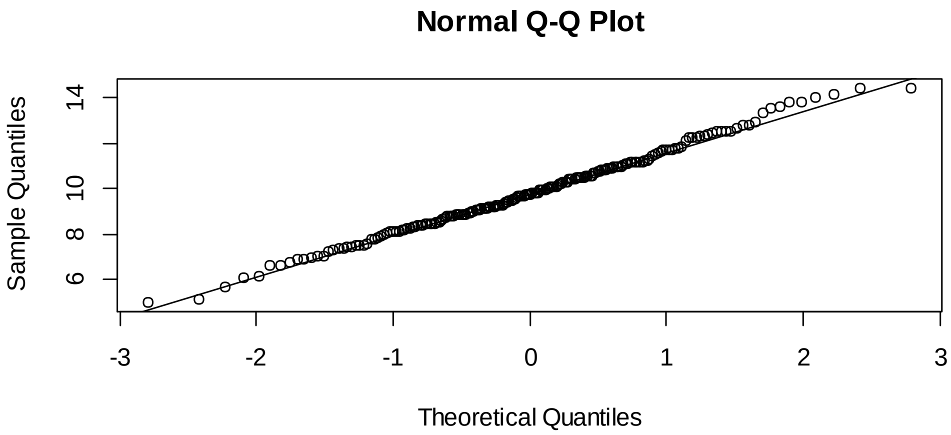
```
> lx=log(policydata$Amount[policydata$Amount>0])
> #Alternative, if dplyr installed
> #lx=log(filter(policydata,Amount >0)$Amount)

> mean(lx)
[1] 9.835205
> median(lx)
[1] 9.774659
> sd(lx)^2
[1] 3.425705
```

(4)

v)

```
> par(mfrow=c(2,1))
> hist(lx)
> qqnorm(lx)
> qqline(lx)
```





(3)

vi)

> # From Histogram and QQPlot it seems log amount closely follows normal distribution.
> # To add, the mean and median are very close indicating symmtery. One of the characterstics of Z.
> # Hence,Claim amount might be following log normal distribution.

(3)

vii)

```
> #Null Hypothesis : mu = 10 , alternate hypothesis mu >10
> t.test(lx,mu=10,alternative="greater", conf.level = .9)

        One Sample t-test

data:  lx
t = -1.2337, df = 191, p-value = 0.8906
alternative hypothesis: true mean is greater than 10
90 percent confidence interval:
 9.663428      Inf
sample estimates:
mean of x
```

```
 9.835205

> #Given p-value greater than 10% null hypothesis can not be rejected.
```

(4)

viii)

```
> ct=table(policydata$Claim,policydata$Cust_Exp)
> ct

    DS  SA  VD  VS
  0  63 306  20  69
  1  36  90   9  14
  2  16  14   6   0
  3   3   3   1   0
> # Null Hypothesis: No association between Policyholder's experience and Claim
> chisq.test(ct)

        Pearson's Chi-squared test

data:  ct
X-squared = 47.749, df = 9, p-value = 2.846e-07

Warning message:
In chisq.test(ct) : Chi-squared approximation may be incorrect
```

(3)

ix)

```
> #There are cells where the number of observations are less than 5.
```

(1)

x)

```
> policydata$Claim2=ifelse(policydata$Claim >2,2,policydata$Claim)
> policydata$Cust_Exp2=ifelse(policydata$Cust_Exp %in% c("DS","VD"),"DS","SA")
> ct2=table(policydata$Claim2,policydata$Cust_Exp2)
> ct2

    DS  SA
  0  83 375
  1  45 104
  2  26  17

> chisq.test(ct2)

        Pearson's Chi-squared test

data:  ct2
X-squared = 43.514, df = 2, p-value = 3.557e-10


> # There is a strong reason to reject null hypothesis.
> # Hence, it can concluded that policyholder's experience gets worse as
claim count increases
```

(5)

xi)

```
> summary(policydata$Amount)
```

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
     0       0       0   29501    3232 1848069
> policydata$large= ifelse(policydata$Amount >100000,1,0)
> x = sum(policydata$large)

> n = length(policydata$Amount[policydata$Amount>0])
> #Alternative, if dplyr installed
> #n = length(filter(policydata,Amount >0)$Amount)

> binom.test(x,n)

        Exact binomial test

data:  x and n
number of successes = 35, number of trials = 192, p-value < 2.2e-16
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.1303796 0.2442928
sample estimates:
probability of success
          0.1822917

>
> # Since upper bound of c.i is less that .25, it is unlikely that more
that
> #25% claims are large
```

(5)

[35 Marks]

Solution 3:

 # Sample mean and variance

```
Motorclaim = read.csv("Motorclaim.CSV")
Mean_Claim<-mean(Motorclaim$CLAIM)
Var_Claim<-var(Motorclaim$CLAIM)
```

i)

 # Method of moments estimate

 # Normal Distribution

```
Normal_mu <- Mean_Claim
Normal_sigma <- sqrt(Var_Claim)

Normal_mu
[1] 6357.314

Normal_sigma
[1] 6986.523
```

 # Log Normal Distribution

```
LogNormal_sigma<- sqrt(log(1+Var_Claim/Mean_Claim^2))
LogNormal_mu<-log(Mean_Claim)-LogNormal_sigma^2/2
```

LogNormal_sigma
[1] 0.8899276

LogNormal_mu
[1] 8.361376

 # Exponential Distribution

Exp_lamda <- 1/Mean_Claim

Exp_lamda
[1] 0.0001572991

 # Gamma Distribution

Gamma_lamda<-Mean_Claim/Var_Claim
Gamma_alpha<-Gamma_lamda*Mean_Claim

Gamma_lamda
[1] 0.0001302421

Gamma_alpha
[1] 0.82799

(8)

ii)
 # Histogram

hist(Motorclaim$CLAIM,breaks = 35,freq = FALSE)

#Superimpose Normal distribution

curve(dnorm(x,mean = Normal_mu,sd = Normal_sigma),from = min(Motorclaim$CLAIM), to = max(Motorclaim$CLAIM), add = TRUE, col= "blue")

#Superimpose Log Normal distribution

curve(dlnorm(x,meanlog = LogNormal_mu,sdlog = LogNormal_sigma),from = min(Motorclaim$CLAIM), to = max(Motorclaim$CLAIM), add = TRUE, col= "green")

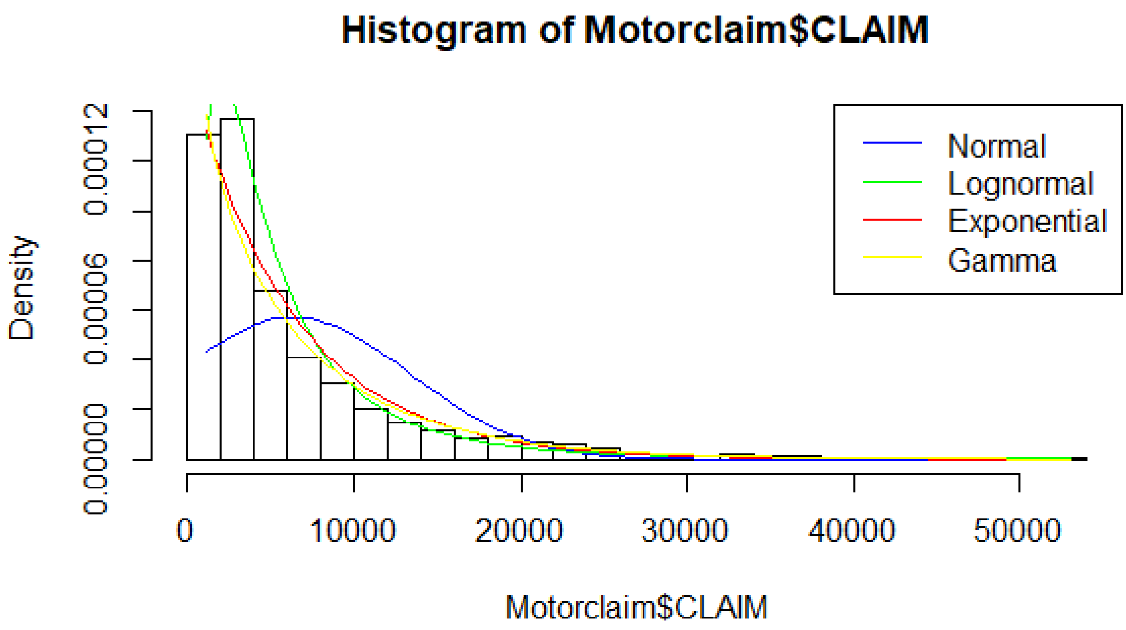
#Superimpose Exponential distribution

curve(dexp(x,rate = Exp_lamda),from = min(Motorclaim$CLAIM), to = max(Motorclaim$CLAIM), add = TRUE, col= "red")

#Superimpose Gamma distribution

curve(dgamma(x,shape = Gamma_alpha,rate = Gamma_lamda),from = min(Motorclaim$CLAIM), to = max(Motorclaim$CLAIM), add = TRUE, col= "yellow")

legend("topright",legend = c("Normal", "Lognormal", "Exponential", "Gamma"),lty = 1, col = c("blue","green","red","yellow"))



(8)

iii)

 # Quantiles

 # Actual Claim Data

quantile(Motorclaim$CLAIM,c(0.05,0.25,0.5,0.75,0.95))

```
      5%       25%       50%       75%       95% 
 1324.561  1934.876  3631.070  7870.028 21246.913
```

 # Normal Distribution

qnorm(c(0.05,0.25,0.5,0.75,0.95),mean = Normal_mu,sd = Normal_sigma)

[1] -5134.494 1644.976 6357.314 11069.653 17849.123

 # Log Normal Distribution

qlnorm(c(0.05,0.25,0.5,0.75,0.95),meanlog = LogNormal_mu,sdlog = LogNormal_sigma)

[1]   989.8714 2347.5526 4278.5767 7798.0014 18493.5327

 # Exponential Distribution

qexp(c(0.05,0.25,0.5,0.75,0.95),rate = Exp_lamda)

[1]   326.0876 1828.8853 4406.5544 8813.1089 19044.8114

 # Gamma Distribution

```
qgamma(c(0.05,0.25,0.5,0.75,0.95),shape = Gamma_alpha,rate = Gamma_lamda)

[1]   193.6261  1479.4200  4053.4299  8797.0450 20369.6614
```

(5)

iv) From the histogram and superimposed plots it is clear that normal distribution is not good fit to the data.

The other three plots are getting superimposed more or less similar to the data. From the quantiles it is observed that lower value(5th percentile) of lognormal is closed to actual value and higher values(95th percentile) of gamma distribution is closed to actual value

The best fitting distribution among Lognormal, exponential & Gamma can not be decided basis of observations from (ii) & (iii). Further statistical tests need to be carried out to confirm best fit

(4)

v)

```
# Simulation from Gamma distribution

set.seed(2022)
Sim_samples <- rgamma(20000,Gamma_alpha,Gamma_lamda)

head(Sim_samples,10)
[1]  9505.735311  1376.831631   458.302589  3189.065594     5.340363  5821.017458
 [7] 11122.004509  5372.490004 43002.362493  3557.086406
```

(2)

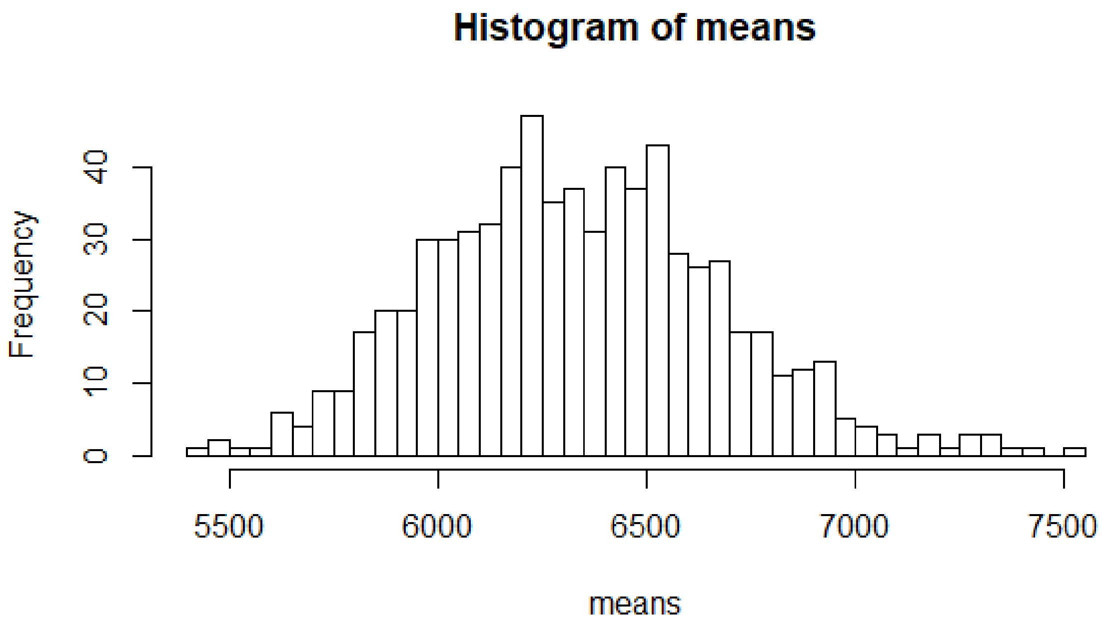
vi)

```
# Generating 700 random samples of size 400 and computing sample means

means<-c()
set.seed(2022)
for (i in 1:700){
selected_data_point<-sample(1:20000,400,FALSE)
random_sample<- Sim_samples[selected_data_point]
sample_mean<-mean(random_sample)
means<-c(means,sample_mean)
}
```

(5)

vii)

```
# Histogram of the sample means
hist(means,breaks = 40)
```



Comment:
The distribution of sample means tend to follow normal distribution however the actual data comes from gamma distribution. Central Limit Theorem states that the sample means tend to follow normal distribution as the sample size increases. The distribution of sample means will be closer to normal distribution by increasing the sample size from its current level of 400.

(3)

[35 Marks]

****