

INSTITUTE OF ACTUARIES OF INDIA

Subject CS1 – Actuarial Statistics (Paper B)

November 2024 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) We know the property that if $W \sim \text{Gamma}(\alpha, \lambda)$ then $2\lambda W$ has χ^2 distribution with 2α degrees of freedom.

We have $X_i \sim \chi^2$ with 2 degrees of freedom.

$$X_i / 2\lambda \sim \text{Gamma}(1, \lambda)$$

We further know that for a chi-square distribution, $\lambda = 1/2$

$$X_i \sim \text{Gamma}(1, 1/2)$$

Hence, $X_i \sim \text{Exp}(1/2)$

..... Gamma distribution with parameters $\alpha=1$ and λ is an exponential distribution with parameter λ

$$Y = \sum_{i=1}^n X_i$$

Hence, $Y \sim \text{Gamma}(n, 1/2)$
variable

..... Sum of 'n' exponential variables is a gamma

(3)

- ii) First copy the numbers in R and define a vector `u_ran`. Then using `qchisq` function convert these uniform random numbers into random numbers from a chi-square distribution with 2 degrees of freedom.

R Code and Output:

```
> u_ran <-c(0.07991847, 0.82064314, 0.33683219, 0.93005953, 0.31919393, 0.92695533,0.7
6263949, 0.51740370, 0.49224880, 0.46354694, 0.89832157, 0.21920729, 0.20471780, 0.190
55074, 0.69137537)
```

```
> print(u_ran)
```

```
[1] 0.07991847 0.82064314 0.33683219 0.93005953 0.31919393 0.92695533
```

```
[7] 0.76263949 0.51740370 0.49224880 0.46354694 0.89832157 0.21920729
```

```
[13] 0.20471780 0.19055074 0.69137537
```

```
> x <- qchisq(u_ran,2)
```

```
[1] 0.1665860 3.4367557 0.8214544 5.3202217 0.7689556 5.2333682 2.8763503
```

```
[8] 1.4571496 1.3555274 1.2455524 4.5718802 0.4948912 0.4581165 0.4228024
```

```
[15] 2.3512591
```

```
> y=sum(x)
```

```
> print(y)
```

```
[1] 30.98087
```

(3)

- iii) H_0 : Standard deviation of X is equal to 2.5.
 H_1 : Standard deviation of X is not equal to 2.5.

```
> n=length(x)
```

```
> sigma=2.5
```

```
> alpha=0.05
```

```
>
```

```
> statistic <- (n-1)*var(x)/sigma^2
```

```

> statistic
[1] 7.305598
>
> #critical value
> qchisq(alpha/2,n-1)
[1] 5.628726
> qchisq(alpha/2,n-1,lower=FALSE)
[1] 26.11895
>
> #p-value
> 2*(pchisq((n-1)*var(x)/sigma^2,df=n-1))
[1] 0.1554235

```

As the p-value is $> 5\%$, we do not have sufficient evidence to reject the null hypothesis and hence we can conclude that the standard deviation of X is equal to 2.5. (6)

- iv) We need to write a code to obtain sample for y with 1,000 simulations. Following code as given the question can be used for that:

R Code and Output:

```

> set.seed(47)
> y = 0*(1:1000)
> for(i in 1:1000){
+   y[i] = sum(rchisq(15,2))
+ }

> sum(y)
[1] 30508.66

```

(4)

- v) The distribution of Y when sample size = 15, is not perfectly symmetrical like a normal distribution. This is given because Y seems to accept only positive values as gamma distribution is defined only when $y > 0$.

However, when sample size = 10,000 this histogram is symmetrical and is closer to a normal distribution.

As n tends to infinity, using central limit theorem, the distribution of the sample approaches a normal distribution.

For a larger sample size of n (changed from 15 to 10,000) the central limit theorem ensures that the distribution of Y becomes approximately normal. (3)

- vi) By modifying the code in part (iv), we generate 10,000 values of x and calculate the sample mean and sample variance for each sample.

R Code and Output:

```

> set.seed(47)
> x_bar = 0*(1:1000)
> s_squared = 0*(1:1000)
> for(i in 1:1000){
+   x_bar[i] = mean(rchisq(10000,2))
+   s_squared[i] = var(rchisq(10000,2))

```

```
+ } (5)
```

```
vii) >
> print(mean(x_bar))
[1] 1.998798
```

Also, by visual inspection we can see that the mean of the sample means is close to 2.

Population mean is 2 (number of degrees of freedom of the variable X_i which has chi-square distribution). Since $E(\bar{X}) = \mu$, we can conclude that \bar{X} is an unbiased estimator of population mean μ .

```
> print(mean(s_squared))
[1] 3.995956
```

Also, by visual inspection we can see that the mean of the sample variances is close to 4.

Population variance is 4 (2 times the number of degrees of freedom of the variable X_i which has chi-square distribution). Since $E(S^2) = \sigma^2$, we can conclude that S^2 is an unbiased estimator of population variance σ^2 . (4)

viii) Comment:

The plot of \bar{X} is indicative of normality. This is true as for large sample size, $\bar{X} \sim N(\mu, \sigma^2 / n)$.

However, plot of S^2 is relatively less normal as compared to plot of \bar{X} . (2)
[30]

Solution 2:

- i)
 - A. \bar{X} represents the total number of claims for every insurer (total of no of claims across all the years)
 - B. \bar{S}^2 is a quantity calculated by taking sum over all insurers of (total number of claims per insurer multiplied by proportion of total no of claims of that insurer to total) and divided by total number of cells minus 1. This quantity is used in the denominator while determining $V(m(\theta))$.
 - C. m or $E(m(\theta))$ is the expected value of average claims per policy based on the collateral data.
 - D. s or $E(s^2(\theta))$ is the expected value of the variance of claims per policy based on the collateral data.
 - E. v or $V(m(\theta))$ is the variance of average claims per policy based on the collateral data (5)

```
ii) > claims<-data.frame(
+ Year1 = c(14.2,58.6,123),
+ Year2 = c(15.8,63.1,132),
+ Year3 = c(22.7,81.0,161),
+ Year4 = c(19,64.2,133)
+ )
>
> nopols<-data.frame(
+ Year1 = c(163,4435,16184),
+ Year2 = c(189,4761,17443),
```

```

+ Year3 = c(252,5576,20102),
+ Year4 = c(199,4581,18000)
+)
>
> n <- ncol(claims) ## This stands for n
> N <- nrow(claims) ## This stands for N
> X <- claims/nopols ## This stands for Xij
> Xibar <- rowSums(claims) / rowSums(nopols) # Xibar
> Pibar <- rowSums(nopols) # Pibar
> Pbar <- sum(Pibar) # Pbar
> Pstar <- sum(Pibar * (1-Pibar/Pbar))/(N*n-1) # Pstar
> m <- sum(claims) / Pbar # E[m(θ)]
> print(m)
[1] 0.009659901
>
> s <- mean(rowSums(nopols *(X-Xibar)^2)/(n-1)) # E[s2(θ)]
> print(s)
[1] 0.002749923

> v <- (sum(rowSums(nopols*(X-m)^2))/(n*N-1)-s)/Pstar
> print(v)
[1] 0.0001793695

```

(3)

```

iii) > Zi <- Pibar / (Pibar + s/v)
> print(Zi)
[1] 0.9812655 0.9992084 0.9997863

```

(2)

```

iv) > Premi <- Zi * Xibar + (1-Zi) * m
> print(Premi)
[1] 0.087798326 0.013787873 0.007654237

```

```

> nopols_y5 <- c(5000,4800,4200)
> claims_y5 <- Premi * nopols_y5

```

```

> print(claims_y5)
[1] 438.99163 66.18179 32.14779

```

(3)

- v) Since the credibility premiums are high and close to 1 in case of all insurers, we are giving more importance to the direct data (related to that particular insurer) and ignoring the collateral data (data related to other insurers)

This is reasonable as there is wide variation in the claims amount for various insurers and hence more emphasis should be given on direct data.

(2)

[15]

Solution 3:

- i) `#a > Sports <- read.csv(Sports.csv)`
- ```

> Model1 <- lm(Sports$Points ~ Sports$X100m + Sports$X400m + Sports$X110m.hurdle + Sports
$High.jump + Sports$Long.jump + Sports$Pole.vault + Sports$Shot.put + Sports$Javeline + Sports
$Discus)
> summary(Model1)

```

Call:

```
lm(formula = Sports$Points ~ Sports$X100m + Sports$X400m + Sports$X110m.hurdle +
 Sports$High.jump + Sports$Long.jump + Sports$Pole.vault +
 Sports$Shot.put + Sports$Javeline + Sports$Discus)
```

Residuals:

```
Min 1Q Median 3Q Max
-97.106 -25.043 -7.748 33.856 119.528
```

Coefficients:

```
Estimate Std. Error t value Pr(>|t|)
(Intercept) 8632.632 1368.083 6.310 7.83e-06 ***
Sports$X100m -163.414 76.208 -2.144 0.046756 *
Sports$X400m -78.141 16.269 -4.803 0.000166 ***
Sports$X110m.hurdle -120.901 41.818 -2.891 0.010152 *
Sports$High.jump 810.196 211.532 3.830 0.001340 **
Sports$Long.jump 209.187 63.472 3.296 0.004269 **
Sports$Pole.vault 183.345 66.876 2.742 0.013912 *
Sports$Shot.put 90.349 28.200 3.204 0.005204 **
Sports$Javeline 17.757 2.975 5.969 1.52e-05 ***
Sports$Discus 10.986 5.781 1.900 0.074491 .
```

---

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

Residual standard error: 60.74 on 17 degrees of freedom

Multiple R-squared: 0.9794, Adjusted R-squared: 0.9685

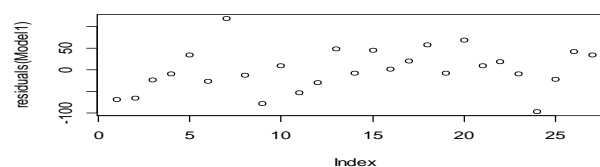
F-statistic: 89.91 on 9 and 17 DF, p-value: 1.491e-12

Using a general rule if the p-value is less than 0.05, then the concerned explanatory variable is considered to be significant. This is shown by \*, \*\*, \*\*\* signs in the R summary output .

So, based on the R-output, only points for Discus Throw are not significant. All other explanatory variables are considered to be significant.

(6)

### #b Plot of residuals for the model



(2)

ii) `> Model2_poisson<-glm(Sports$Points~Sports$X100m+Sports$X400m+Sports$X110m.hurdle+Sports$High.jump+Sports$Long.jump+Sports$Pole.vault+Sports$Shot.put+Sports$Javeline+Sports$Discus,family="poisson")`

`> summary(Model2_poisson)`

Call:

```
glm(formula = Sports$Points ~ Sports$X100m + Sports$X400m + Sports$X110m.hurdle +
 Sports$High.jump + Sports$Long.jump + Sports$Pole.vault +
 Sports$Shot.put + Sports$Javeline + Sports$Discus, family = "poisson")
```

Deviance Residuals:

| Min      | 1Q       | Median  | 3Q      | Max     |
|----------|----------|---------|---------|---------|
| -1.02319 | -0.25491 | 0.03473 | 0.36449 | 1.31328 |

Coefficients:

|                      | Estimate   | Std. Error | z value | Pr(> z )     |
|----------------------|------------|------------|---------|--------------|
| (Intercept)          | 9.0988649  | 0.2501151  | 36.379  | < 2e-16 ***  |
| Sports\$X100m        | -0.0211126 | 0.0139153  | -1.517  | 0.12921      |
| Sports\$X400m        | -0.0093734 | 0.0029665  | -3.160  | 0.00158 **   |
| Sports\$X110m.hurdle | -0.0161147 | 0.0076612  | -2.103  | 0.03543 *    |
| Sports\$High.jump    | 0.1009034  | 0.0387525  | 2.604   | 0.00922 **   |
| Sports\$Long.jump    | 0.0239972  | 0.0116951  | 2.052   | 0.04018 *    |
| Sports\$Pole.vault   | 0.0225249  | 0.0122296  | 1.842   | 0.06550 .    |
| Sports\$Shot.put     | 0.0111537  | 0.0051471  | 2.167   | 0.03024 *    |
| Sports\$Javeline     | 0.0021487  | 0.0005379  | 3.995   | 6.48e-05 *** |
| Sports\$Discus       | 0.0012340  | 0.0010555  | 1.169   | 0.24236      |

---

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

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 374.4158 on 26 degrees of freedom  
 Residual deviance: 8.3483 on 17 degrees of freedom  
 AIC: 321

Number of Fisher Scoring iterations: 3

(4)

**iii)** Scaled deviance can be used to compare only nested models.

Since, Model 1 and Model2\_Poisson are models with different distributional assumptions.

Model 1 assumes normal distribution and Model 2 assumes Poisson distribution.  
 They are not nested models.

Hence, scaled deviance cannot be used to compare the models in parts (i) and (ii).

(2)

**iv)** An equivalent model to a linear regression model will be a GLM with normal distribution.

```
> Model2_normal<-glm(Sports$Points~Sports$X100m+Sports$X400m+Sports$X110m.hurdle+Sports$High.jump+Sports$Long.jump+Sports$Pole.vault+Sports$Shot.put+Sports$Javeline+Sports$Discus,family=gaussian())
> summary(Model2_normal)
```

Call:

```
glm(formula = Sports$Points ~ Sports$X100m + Sports$X400m + Sports$X110m.hurdle + Sports$High.jump + Sports$Long.jump + Sports$Pole.vault + Sports$Shot.put + Sports$Javeline + Sports$Discus, family = gaussian())
```

Deviance Residuals:

| Min     | 1Q      | Median | 3Q     | Max     |
|---------|---------|--------|--------|---------|
| -97.106 | -25.043 | -7.748 | 33.856 | 119.528 |

Coefficients:

|             | Estimate | Std. Error | t value | Pr(> t )     |
|-------------|----------|------------|---------|--------------|
| (Intercept) | 8632.632 | 1368.083   | 6.310   | 7.83e-06 *** |

```

Sports$X100m -163.414 76.208 -2.144 0.046756 *
Sports$X400m -78.141 16.269 -4.803 0.000166 ***
Sports$X110m.hurdle -120.901 41.818 -2.891 0.010152 *
Sports$High.jump 810.196 211.532 3.830 0.001340 **
Sports$Long.jump 209.187 63.472 3.296 0.004269 **
Sports$Pole.vault 183.345 66.876 2.742 0.013912 *
Sports$Shot.put 90.349 28.200 3.204 0.005204 **
Sports$Javeline 17.757 2.975 5.969 1.52e-05 ***
Sports$Discus 10.986 5.781 1.900 0.074491 .

```

---

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

(Dispersion parameter for gaussian family taken to be 3689.4)

Null deviance: 3048179 on 26 degrees of freedom  
Residual deviance: 62720 on 17 degrees of freedom  
AIC: 307.89

Number of Fisher Scoring iterations: 2

(4)

- v) Model 1 in part (i) and Model 2 (Normal) in part (iv) are exact equivalents of each other. The same can be seen from the estimates of coefficients in the R summary output.

So, for comparing the fit of Model 1 and Model 2 (Poisson), Model 2 (Normal) can be used as a proxy for Model 1. And then, the AIC of Model 2 (Poisson) can be compared with the AIC of Model 2 (Normal).

AIC (Model 2 Normal) = 307.89

AIC (Model 2 Poisson) = 321

Smaller the AIC, better is the fit. So, Model 2 Normal is a better fit as compared to Model 2 Poisson.

Consequently, the linear multiple regression model in part (i) is a better fit to the data as compared to the GLM fitted in part (ii).

(2)

- vi) 

```

> run<-data.frame(Sports$X100m,Sports$X400m,Sports$X110m.hurdle)
> pr_run<-prcomp(run,scale. = TRUE)
>
> summary(pr_run)
Importance of components:
 PC1 PC2 PC3
Standard deviation 1.492 0.6680 0.5726
Proportion of Variance 0.742 0.1487 0.1093
Cumulative Proportion 0.742 0.8907 1.0000

>
> jump<-data.frame(Sports$High.jump,Sports$Long.jump,Sports$Pole.vault)
> pr_jump<-prcomp(jump,scale. = TRUE)
>
> summary(pr_jump)
Importance of components:
 PC1 PC2 PC3

```

```
Standard deviation 1.2588 1.0307 0.5941
Proportion of Variance 0.5282 0.3541 0.1176
Cumulative Proportion 0.5282 0.8824 1.0000
```

```
>
> throw<-data.frame(Sports$Shot.put,Sports$Javeline,Sports$Discus)
> pr_throw<-prcomp(throw,scale. = TRUE)
>
> summary(pr_throw)
Importance of components:
 PC1 PC2 PC3
Standard deviation 1.4141 0.8727 0.48853
Proportion of Variance 0.6665 0.2539 0.07955
Cumulative Proportion 0.6665 0.9204 1.00000
```

For Run Sports Category, 74.2% variance is captured by PC1.  
 For Jumping Sports Category, 52.82% variance is captured by PC1.  
 For Throw Sports Category, 66.65% variance is captured by PC1. (7)

**vii)** Considering the summary R output generated in part (vi),

In case of Run Sports Category, all three PCs cumulatively capture at least 90% of the total variance of the data.

In case of Jump Sports Category, all three PCs cumulatively capture at least 90% of the total variance of the data.

In case of Throw Sports Category, first two PCs cumulatively capture 92.04% (at least 90%) of the total variance of the data.

So, if PCs capturing at least 90% of the variance is a criterion for reducing the dimensionality of the data set, then Throw Sports Category satisfies it. For Throw Sports, PC1 and PC2 can be retained and PC3 can be dropped thus reducing the dimensionality of the data set. In case of Run Sports and Jumping Sports, all three PCs need to be retained and thus the dimensionality of the data set will not be reduced. (3)

**viii)** `> Model3<-lm(Sports$Points~Sports$Pole.vault)`

$H_0$ : Beta coefficient is equal to 0  
 $H_1$ : Beta coefficient is not equal to 0.

```
> confint(Model3,level=0.95)
 2.5 % 97.5 %
(Intercept) 5654.3838 10895.1124
Sports$Pole.vault -573.3559 508.9227
```

As the 95% confidence interval for beta (-573.3559, 508.9277) contains the value 0, we do not have sufficient evidence to reject the null hypothesis at 5% level of significance. Hence, based on this test, one can conclude that there is no correlation between pole vault score and winning points. (5)

**ix)** `> Model4<-lm(Sports$X110m.hurdle~Sports$X100m)`  
`> summary(Model4)`

Call:  
lm(formula = Sports\$X110m.hurdle ~ Sports\$X100m)

Residuals:  
Min 1Q Median 3Q Max  
-0.48075 -0.28936 -0.05353 0.21428 0.76203

Coefficients:  
Estimate Std. Error t value Pr(>|t|)  
(Intercept) 2.2036 2.7252 0.809 0.426379  
Sports\$X100m 1.1183 0.2478 4.512 0.000132 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.356 on 25 degrees of freedom  
Multiple R-squared: 0.4489, Adjusted R-squared: 0.4268  
F-statistic: 20.36 on 1 and 25 DF, p-value: 0.0001319

```
> score<-c(10.68,10.42,11.68,11.62,10.54)
> hurdle_score<-2.2036+1.1183*score
> hurdle_score
[1] 14.14704 13.85629 15.26534 15.19825 13.99048
```

(5)  
[40]

#### Solution 4:

i) 

```
> exam.success = matrix(c(132,120,27,51),ncol=2,nrow=2)
> exam.success
[,1] [,2]
[1,] 132 27
[2,] 120 51
```

Data is properly getting displayed.

$H_0$ : tutorial attendance and exam success are independent, against  
 $H_1$ : tutorial attendance and exam success are not independent

(1)

ii) 

```
> chisq.test(exam.success)$expected
[,1] [,2]
[1,] 121.4182 37.58182
[2,] 130.5818 40.41818
```

(2)

iii) 

```
> chisq.test(exam.success)
```

Pearson's Chi-squared test with Yates' continuity correction

```
data: exam.success
X-squared = 6.8349, df = 1, p-value = 0.008939
```

The p-value is significant (e.g. at the 1% level), since  $0.008939 < 0.01$  – therefore there is evidence to reject the null hypothesis and we conclude that tutorial attendance and exam success are not independent.

(3)

iv) `> fisher.test(exam.success)`

#### Fisher's Exact Test for Count Data

```
data: exam.success
p-value = 0.006544
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.190372 3.671876
sample estimates:
odds ratio
 2.073216
```

The p-value is significant (e.g. at the 1% level),  $0.006544 < 0.01$  – therefore there is evidence to reject the null hypothesis and we conclude that tutorial attendance and exam success are not independent. Conclusion under Fisher's exact test is similar to conclusion under contingency table test. (3)

v) (a) Fisher's test is an exact test whereas chi-square test is an approximation

(b) Fisher's test is suitable for  $2 \times 2$  datasets whereas chi-square test can be used for  $N \times N$  datasets. (2)

vi)  $H_0$ : the proportion of students passing the exam is 60% ( $p = 0.60$ )  
 $H_1$ : the proportion of students passing the exam is not equal to 60% ( $p \neq 0.60$ )

```
> x=132+120
> n=x+27+51
> binom.test(x,n,conf.level = 0.95)
```

#### Exact binomial test

```
data: x and n
number of successes = 252, number of trials = 330, p-value <
2.2e-16
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.7140288 0.8084419
sample estimates:
probability of success
 0.7636364
```

The p-value is  $< 2.2e-16$  which is definitely less than 5% and hence we have sufficient evidence to reject the null hypothesis and hence we can conclude that the proportion of students passing the examination is not equal to 60%. (4)

[15]

\*\*\*\*\*