INSTITUTE AND FACULTY OF ACTUARIES

EXAMINERS' REPORT

April 2019 Examinations

Subject CS2B - Risk Modelling and Survival Analysis
Core Principles

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

Mike Hammer
Chair of the Board of Examiners
July 2019

A. General comments on the *aims of this subject and how it is marked*

The aim of Risk Modelling and Survival Analysis (CS2) is to develop knowledge of and ability to apply statistical methods for risk modelling, time series analysis methods, stochastic processes (especially Markov chains and Markov jump processes), survival analysis (including regression methods applied to duration data) and graduation methods. It also includes a high-level introduction to machine learning. The exam paper aims at checking your understanding on both theory and application of the ideas to real data sets using R. We are not testing knowledge of the R program.

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

There was a different experience in paper B but the candidates provided sensible answers in terms of R code and interpreting its output. As expected the answers in were a lot more diverse as there are many ways to perform a single task in R.

C. Pass Mark

The Pass Mark for this exam was 55

Solutions for Subject CS2-B April 2019

Q1

(i)

```
> DriverZone <- c("North", "South", "West")
> DriverZone
```

[1] "North" "South" "West"

Note: Alternative and reasonable names for the states receive full marks. [3]

(ii)

```
> ZoneTransition <- matrix(c(0.3, 0.3, 0.4, 0.4, 0.4, 0.2,
0.5, 0.3, 0.2), nrow = 3, byrow = T, dimname =
list(DriverZone, DriverZone))
> ZoneTransition
```

	North	South	West
North	0.3	0.3	0.4
South	0.4	0.4	0.2
West	0.5	0.3	0.2

[3]

Notes:
(1) Row/column names are not necessary to get full marks.
(2) There are many ways to create a matrix in R. Any code that produces the correct matrix gets full marks.

(iii)

```
> install.packages("markovchain") # if not installed
> library(markovchain)
```

[3]

Notes:
(1) The installation of the package is not necessary to get full marks.
(2) Note that R is case sensitive and candidates were penalised for that. .

(iv)

```
> MCZone <- new("markovchain", states = DriverZone, byrow =
T, transitionMatrix = ZoneTransition, name = "Driver
Movement")
> MCZone
```

Driver Movement
A 3 - dimensional discrete Markov Chain defined by the following states:
North, South, West
The transition matrix (by rows) is defined as follows:

	North	South	West
North	0.3	0.3	0.4
South	0.4	0.4	0.2
West	0.5	0.3	0.2

[3]

(v) (a)

```
> MCZone^2
```

Driver Movement^2
A 3 - dimensional discrete Markov Chain defined by the following states:
North, South, West

The transition matrix (by rows) is defined as follows:

	North	South	West
North	0.41	0.33	0.26
South	0.38	0.34	0.28
West	0.37	0.33	0.30

[1]

So the required probability in 2 trips is 41% or 0.41 [1]

Notes:

(1) Alternatively, candidates who multiply matrices (using R) to get the required probability should get full marks.

(2) ½-1mark is deducted if the candidate does not specify the probability from the transition matrix /vector

(v) (b)

```
> MCZone^3
```

Driver Movement^3

A 3 - dimensional discrete Markov Chain defined by the following states:

North, South, West

The transition matrix (by rows) is defined as follows:

	North	South	West
North	0.385	0.333	0.282
South	0.390	0.334	0.276
West	0.393	0.333	0.274

[1]

So the required probability in 3 trips is 38.5% or 0.385 [1]

Notes:

(1) Alternatively, candidates who multiply matrices (using R) to get the required probability should get full marks.

(2) Half a mark is deducted if the candidate does not specify the probability from the transition matrix /vector

(vi)

```
> steadyStates(MCZone)
          North     South      West
[1,] 0.3888889 0.3333333
0.2777778
```

[4]

Notes:

(1) Alternative codes that yield the correct answer are accepted. For example raising appropriate matrix to large powers, or solving a linear system of equation as follows

A <- rbind(t(ZoneTransition) - diag(3), c(1,1,1))

b <- c(1,1,1,1)

qr.solve(A, b)

(2) If the candidate's output is a vector rather than a matrix then the candidate will need to raise the transition matrix to more than one large power to prove that it has reached the steady state. If the matrix has only been raised to one large power and the output is in vector format then at most half is awarded.

[Total 20]

This question was answered generally well with an average mark of about 15.

Q2

(i) (a)

```
> Exp_Vector <- rexp(1000,0.4)
> mean(Exp_Vector)  # or  summary(Exp_Vector)
> var(Exp_Vector)
```

The mean and variance will vary due to the random number generation. If the sample size was large enough, the mean and variance should be close the underlying distribution (exponential with parameter 0.4) as follows:

Mean = 2.5

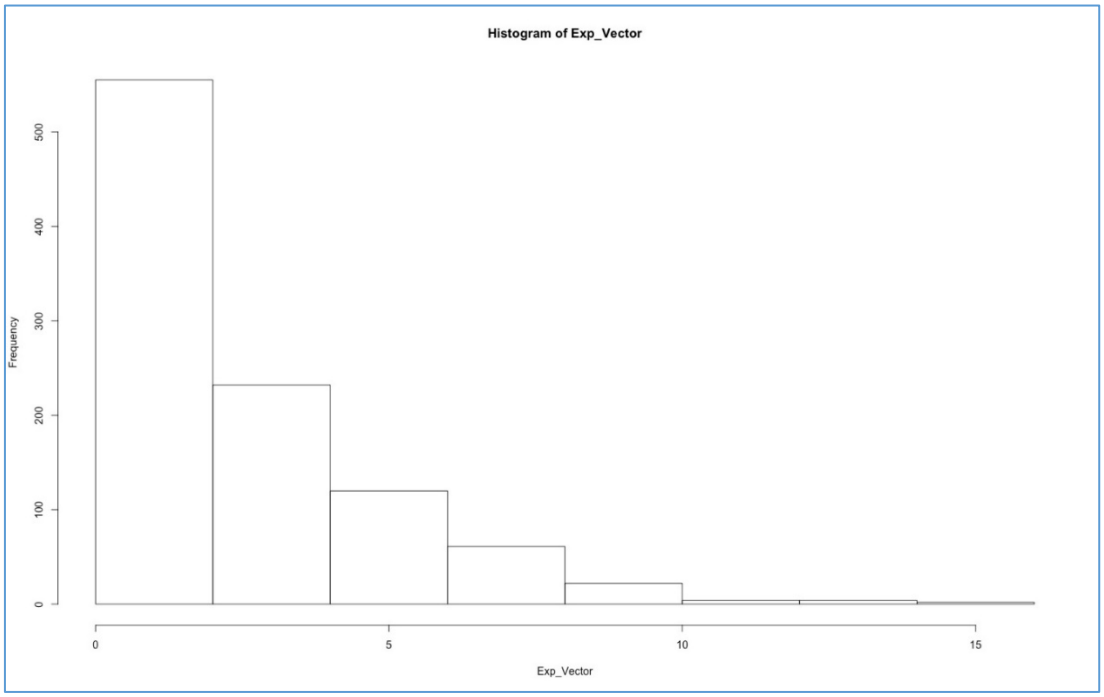
Variance = 6.25

[3]

The correct R code receives full marks.
Candidates are not required to paste their simulated sample here.

(i) (b)

```
> hist(Exp_Vector)
```

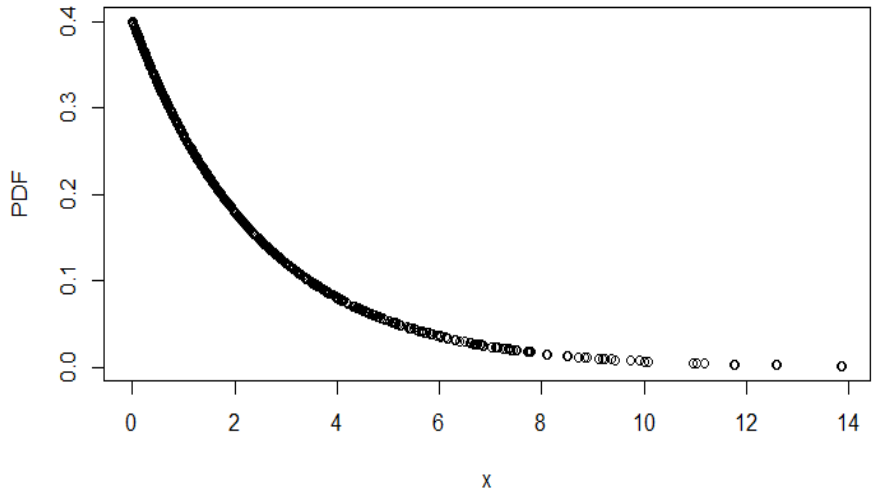


[3]

Note: Plotting the histogram with wrong axis labels receives no more than 2marks.

(i) (c) 1.

```
> x   <- sort(Exp_Vector)
> PDF <- dexp(x, 0.4)
> plot(x, PDF)
```

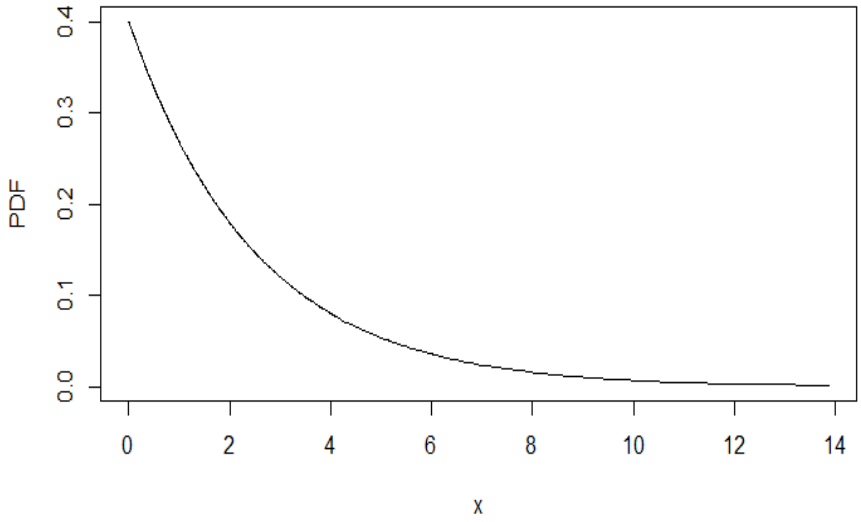


[2]

Note: Alternatively, plotting the theoretical exponential distribution with parameter 0.4 is acceptable for full marks.

(i) (c) 2.

```
> plot(x, PDF, type="l")
```



[2]
[Total 10]

(ii) (a)

```
> LNorm_Vector <- rlnorm(1000, meanlog = 0, sdlog = 1)
> mean(LNorm_Vector)
> var(LNorm_Vector)
```

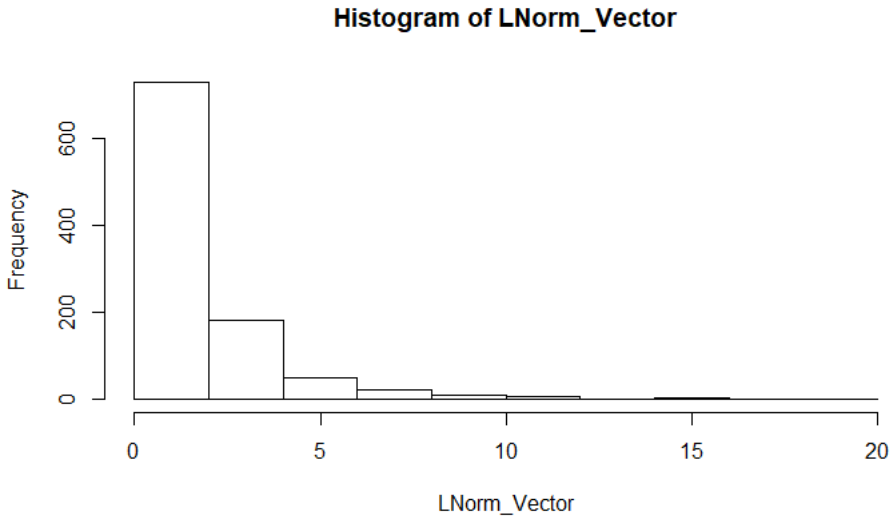
The mean and variance will vary due to the random number generation. If the sample size was large enough, the mean and variance should be close the underlying distribution (lognormal with parameters $\mu = 0$, $\sigma^2 = 1$) as follows:
Mean = 1.649
Variance = 4.6708

[3]

Note:
The correct R code receives full marks.
Candidates are not required to paste theirs simulated sample.

(ii) (b)

```
> hist(LNorm_Vector)
```

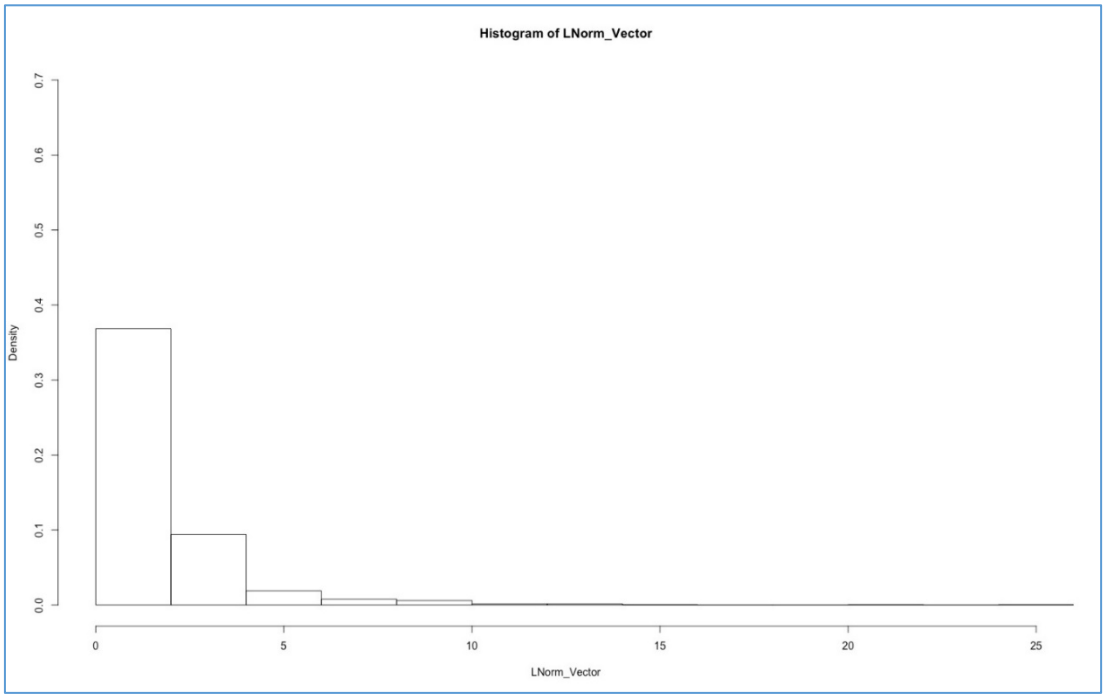


[3]

Note: Plotting the correct histogram but with wrong axis labels receive only 2marks.

(ii) (c)

```
> hist(LNorm_Vector, freq = FALSE, xlim = c(0,25), ylim = c(0,0.7))
```

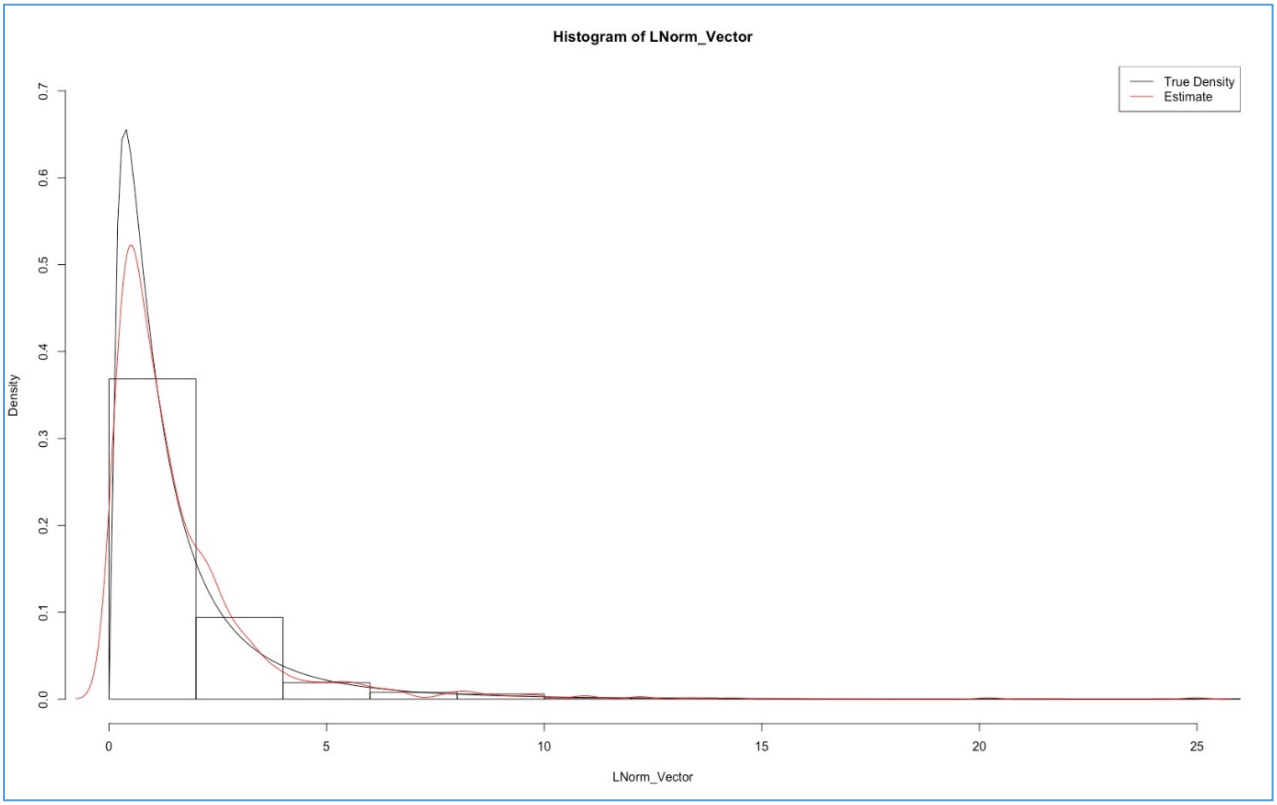


[2]

Note: No credit if the y-axis are frequencies instead of probabilities.

(ii) (d) and (e)

```
> lines(grid, dlnorm(grid,0,1),type="l",xlab="x",ylab="f(x)", col="black")
> lines(density(LNorm_Vector), col="red")
> legend("topright",c("True Density", "Estimate"),lty=1,col=c("black", "red"))
```



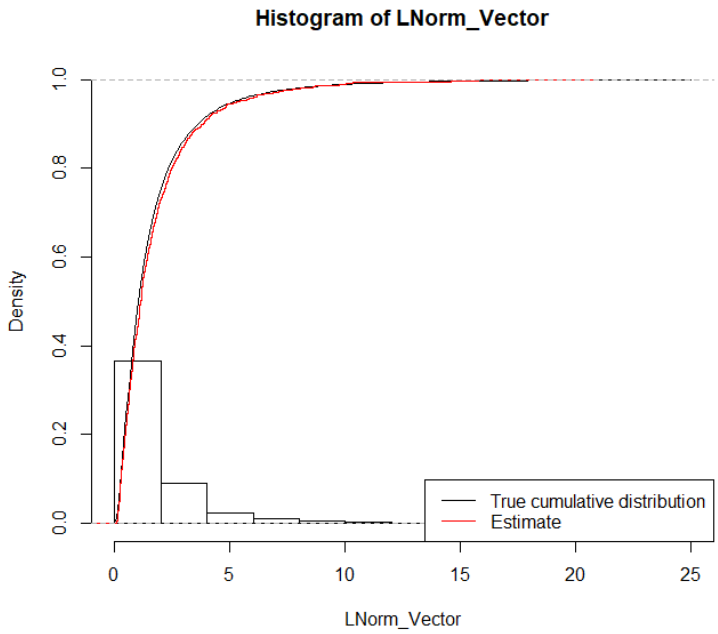
[4]
[Total 12]

Note 1: The breakdown is: 2 marks for each correct line

Note 2: The term "cumulative density" used in this question is a bit confusing. Some candidates interpreted this a cumulative distribution function. Thus, an alternative acceptable solution to (ii)-a and (ii)-b is:

```
> hist(LNorm_Vector, freq = FALSE, xlim = c(0,25), ylim = c(0,1))
> grid = seq(0,25,0.1)
> lines(grid, plnorm(grid,0,1),type="l",xlab="x",ylab="f(x)", col="black")
> lines(ecdf(LNorm_Vector),col="red")
> legend("bottomright",c("True cumulative
distribution","Estimate"),lty=1,col=c("black", "red"))
```

The output is:



(iii) (a)

```
rpareto <- function(n,alpha,lambda) {
   rp <- lambda*( (1-runif(n))^(-1/alpha) -1 )
   rp
}
```

[4]

(iii) (b)

```
> LNorm_Vector = rpareto(1000, 3, 1)
> mean(LNorm_Vector)
> var(LNorm_Vector)
```

The mean and variance will vary due to the random number generation. If the sample size was large enough, the mean and variance should be close the underlying distribution (Pareto $\alpha = 3$, $\lambda = 1$) as follows:
Mean = 0.5
Variance = 0.75 [4]

[Total 8]
[Total 30]

Note: The correct R code receives full marks.

Candidates are not required to paste theirs simulated sample.

Note: Alternative solutions to (iii) are possible. For example,

```
rpareto <- function(alpha, lambda) {
   rp <- lambda*( (1-runif(1))^(-1/alpha) -1 )
   rp
}

LNorm_Vector = replicate( 1000, rpareto(3,1)  )
mean(LNorm_Vector)
var(LNorm_Vector)
```

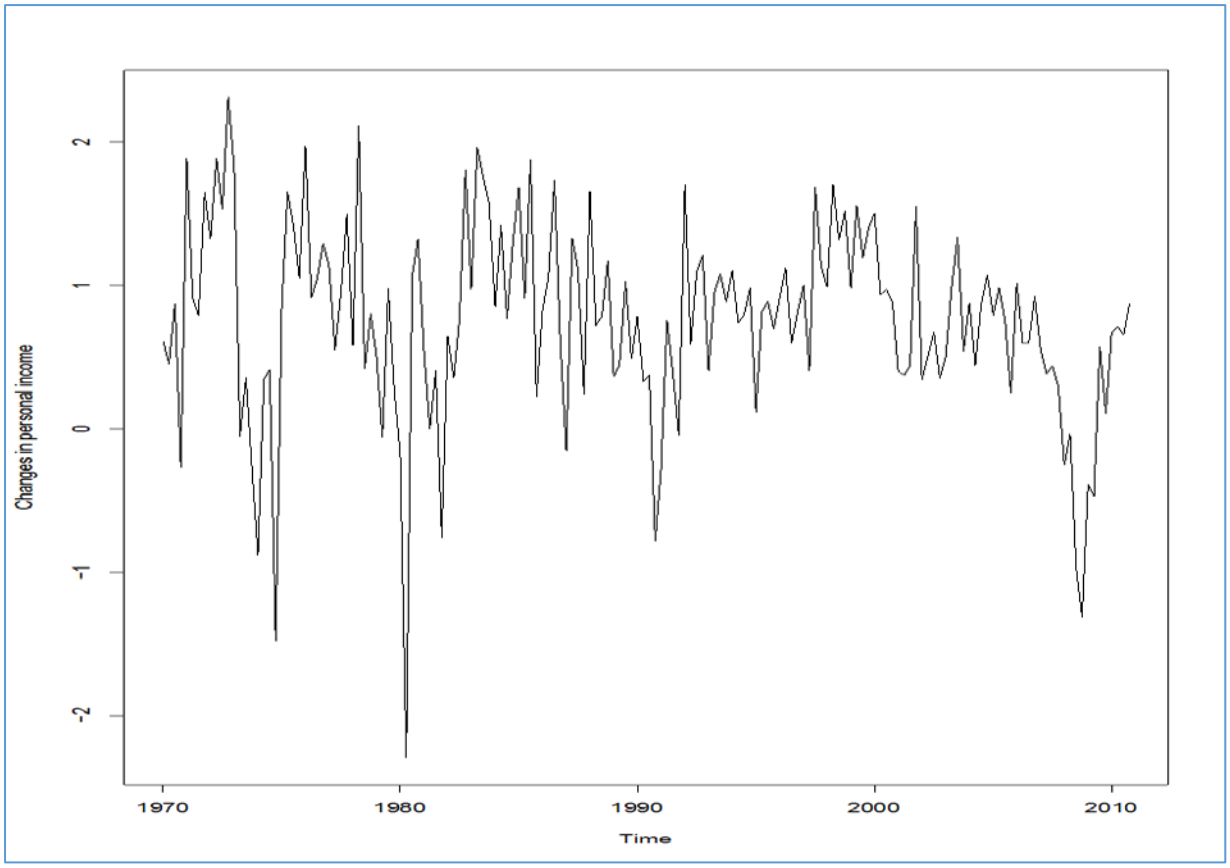
Note: Some candidates may use something equivalent to Pareto_Vector rather than LNorm_Vector. Marks should not be deducted for this.

This question was answered well by most candidates.

Q3

(i)

```
> plot(consumption, ylab = "Changes in personal income")
```

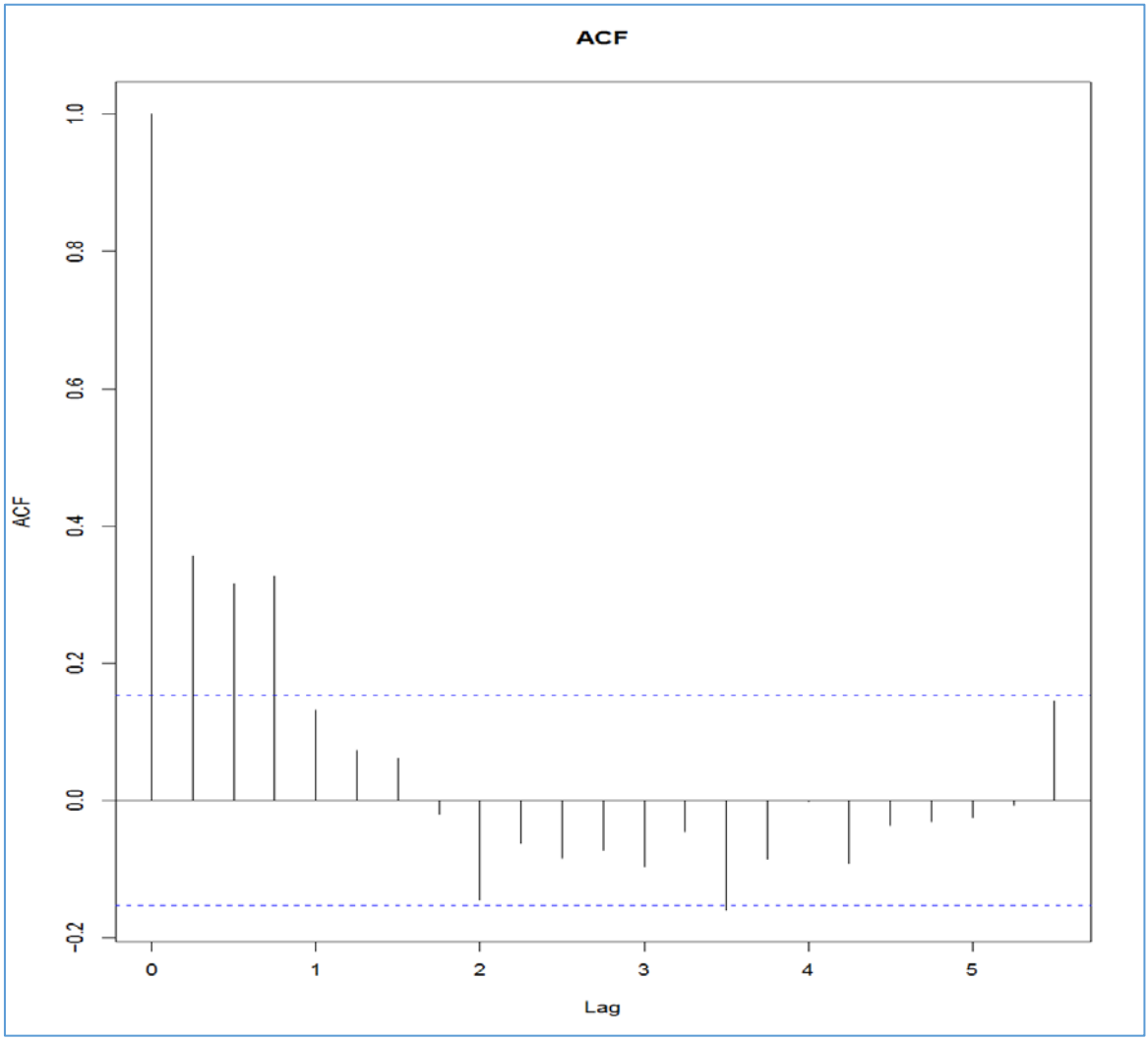


[4]

Notes:
(1) A graphic with wrong axis or unreasonable labels does not get full marks.
(2) There are many ways to produce graphics in R; any working code that produces the correct graphic is awarded full marks.

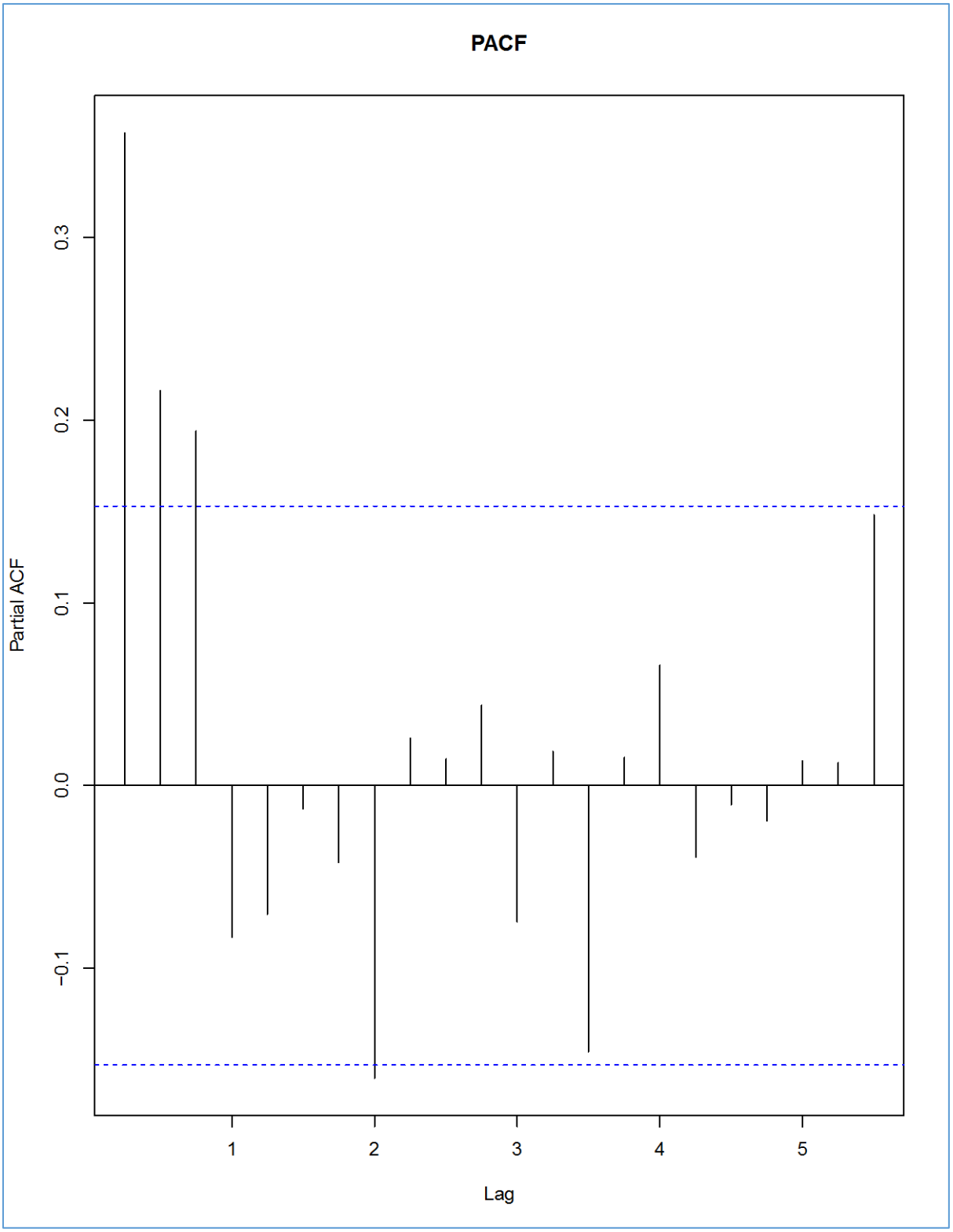
(ii) (a)

```
> acf(consumption, main = "ACF")
```



[4]

```
> pacf(consumption, main = "PACF")
```



[4]

(ii) (b)
The series appears stationary. There are no obvious trends or cycles in the graph of the series and it appears to have constant mean.

The blue dotted lines on the ACF and PACF indicate cut-offs for significance.
For a stationary time-series the ACF should decay to zero quickly and display no signs of oscillation.

The ACF looks to cut out at lag 3 and does not contain any periodic oscillation so this would indicate stationarity.

The PACF shows no significance past lag 3. This again, indicates stationarity.

[4, Max 3]

Note: 1 mark for each valid point, up to a max of 3.

(iii)
We note that for an AR(p) process the PACF is 0 for k>p.
In this instance the PACF seems to cut off at lag 3. A reasonable choice on this basis is an AR(3).

[2]

```
> model1 <- arima(x = consumption, order = c(3, 0, 0))
> model1

Call:
arima(x = consumption, order = c(3, 0, 0))

Coefficients:
         ar1     ar2     ar3  intercept
      0.2366  0.1603  0.1909     0.7533
s.e.  0.0763  0.0774  0.0759     0.1153

sigma^2 estimated as 0.3825:  log likelihood = -154.08,  aic = 318.16
```

[2]

In this case, the equation of the model is:

$$X_t = 0.7533 + 0.2366\, X_{t-1} + 0.1603\, X_{t-2} + 0.1909\, X_{t-3} + \varepsilon_t$$

[1]
[Total 5]

Notes:
(1) Any model of the form ARIMA(p,0,q) with reasonable justification for the choice of p and q receives marks.
(2) One could also let R choose the model by calling the auto.arima function. However, this approach is not awarded full credit if no further justification is provided.

(iv) (a)

forecast from the AR(3)

```
> forecast1 <- fitted(model1)
```

[2]

fit and forecast linear regression

```
> model3 <- lm(consumption ~ income, data = usconsumption)
```

[3]

```
> forecast3 <- fitted(model3)
```

[2]

Calculate RMSE

```
> n     <- length(forecast1)
> rmse1 <- sqrt( sum((forecast1 - consumption)^2)/n )
> rmse1
[1] 0.6185039
```

[2]

```
> rmse3 <- sqrt( sum((forecast3 - consumption)^2)/n )
> rmse3
[1] 0.6236113
```

[2]

(iv) (b)

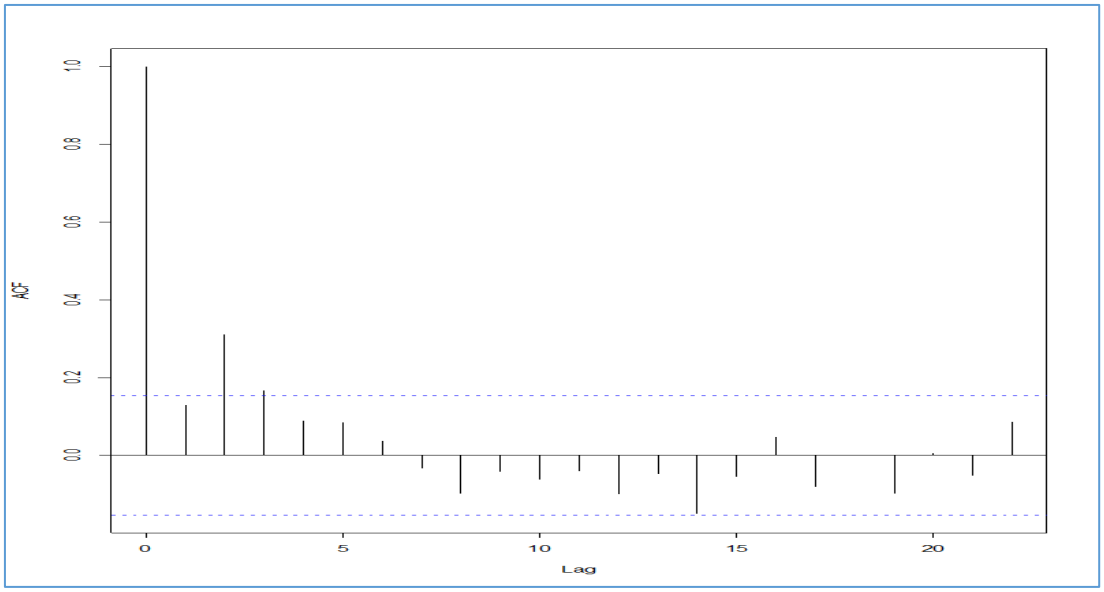
The RMSE for these two models are very similar. However, as a linear model has fewer parameters, it is likely that the linear regression model would be a more reliable model for forecasting. This is confirmed by the Akaike Information Criterion (AIC) as follows:

```
> AIC(model1)
[1] 318.1607
> AIC(model3)
[1] 316.5211
```

316.5211 < 318.1607 so we would prefer the linear regression

However, the residuals of the linear regression are not satisfactory, as shown by the autocorrelation and partial autocorrelation functions.

```
> acf(residuals(model3), main = "")
```



[4]
[Total 15]

Note: 1 mark for each valid point up to a max of 4marks

(v) (a)
It seems reasonable to consider a regression model with ARIMA errors as follows - which essentially blends the two models proposed in (iii) and (iv).

[3]

Note: Alternative extensions to the models fitted in (iii) and (iv), for example adding seasonality component, receive credits.

(v) (b)

```
> model4 <- arima(x=consumption, order = c(3, 0, 0), xreg = income)
```

[4]

Note: Credits awarded whenever the alternative suggested in (a) is fitted correctly.

(v) (c)

```
> AIC(model4)
[1] 300.58
```

Hence model4 is better in terms of AIC

```
> forecast4 <- fitted(model4)
```
[2]

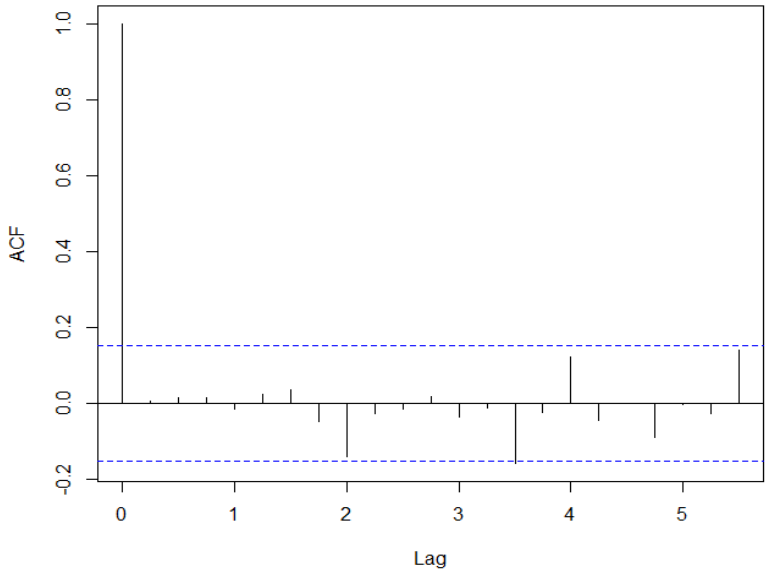
```
> rmse4     <- sqrt( sum((forecast4 - consumption)^2)/n )
```
[2]

```
> rmse4
[1] 0.582791
```

Hence, model4 fits the data better as measured by the RMSE [2]

The residuals from model4 looks better as shown below.

```
> residuals4  <-  residuals(model4)
> acf(residuals4,main="")
```



[2]

Note: 2 marks are awarded for each alternative/relevant comment (especially about the residuals) up to a maximum of 8 marks.

Many candidates struggled especially in 3(iv) and 3(v).

[Total 50]
[Paper Total 100]

END OF EXAMINERS' REPORT