INDIAN INSTITUTE OF TECHNOLOGY MADRAS

Department of Chemical Engineering

CH5350 : Applied Time Series Analysis
Solutions to Assignment #4

1

1.1 Given $H(q^{-1}) = \frac{1+0.4q^{-1}}{1+0.25q^{-2}}$

1.1.1 Wiener-Khinchin theorem

For the given transfer function acf coefficients are calculated using Yule-Walker equations Let

$$v[k] = H(q^{-1})e[k]$$

$$\implies v[k] + 0.25v[k-2] = e[k] + 0.4e[k-1] \tag{1}$$

Multiplying Equation 1 with $v[k-l]$ on both sides and taking expectation we get

$$\sigma_{vv}[l] + 0.25\sigma_{vv}[l-2] = \sigma_{ev}[l] + 0.4\sigma_{ev}[l-1] \tag{2}$$

Substituting $l = 0, 1, 2$ in Equation 2, we get

$$\sigma_{vv}[0] + 0.25\sigma_{vv}[2] = \sigma_{ev}[0] + 0.4\sigma_{ev}[-1] \tag{3}$$

$$\sigma_{vv}[1] + 0.25\sigma_{vv}[l] = \sigma_{ev}[1] + 0.4\sigma_{ev}[0] \tag{4}$$

$$\sigma_{vv}[2] + 0.25\sigma_{vv}[0] = \sigma_{ev}[2] + 0.4\sigma_{ev}[1] \tag{5}$$

We know $\sigma_{ve}[l] = 0 \ \ \forall \ \ l > 0$ Multiplying Equation 1 with $e[k-l]$ on both sides and taking expectation we get

$$\sigma_{ve}[l] + 0.25\sigma_{ve}[l-2] = \sigma_{ee}[l] + 0.4\sigma_{ee}[l-1] \tag{6}$$

Substituting $l = 0, 1$ in Equation 6, we get

$$\sigma_{ve}[0] + 0.25\sigma_{ve}[-2] = \sigma_{ee}[0] + 0.4\sigma_{ee}[1]$$
$$\sigma_{ve}[1] + 0.25\sigma_{ve}[-1] = \sigma_{ee}[1] + 0.4\sigma_{ee}[0]$$

Solving we get $\sigma_{ve}[0] = \sigma_e^2$ and $\sigma_{ve}[1] = 0.4\sigma_e^2$.
Substituting and solving Equation 5, we get $\sigma_v^2 = 1.5\sigma_e^2$, $\sigma_{vv}[1] = 0.32\sigma_e^2$ and $\sigma_{vv}[2] = -0.375\sigma_e^2$.
Extending we get

$$\sigma_{vv}[l] = \begin{cases} = (0.25)^l \sigma_{vv}[1] & \text{when } \ l \ \text{ is odd} \\ = (0.25)^l \sigma_{vv}[2] & \text{when } \ l \ \text{ is even} \end{cases}$$

Wiener-Khinchin theorem states that the

$$\Phi_{vv}(\omega) = \sum_{l=-\infty}^{\infty} \sigma_{vv}(l) \exp(-j\omega l)$$

1.1.2 Time series approach

Given

$$\frac{v[k]}{e[k]} = H(q^{-1}) = \frac{1 + 0.4q^{-1}}{1 + 0.25q^{-2}}$$

We know

$$\Phi_{vv}(\omega) = \left|H(e^{jw})\right|^2 \Phi_{ee}(\omega)$$

$$\Phi_{vv}(\omega) = \frac{1.16 + 0.8\cos\omega}{1.062 + 0.5\cos(2\omega)}\sigma_e^2$$

The spectral density calculated using both the methods are shown in Figures 1 and 2 (σ_e^2 is assumed to be 1).

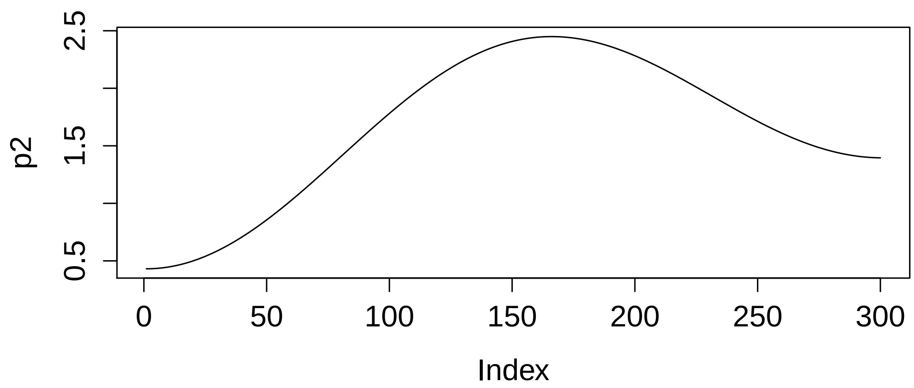


Figure 1: Spectral density obtained using Wiener-Khinchin theorem (maximum lag 200)

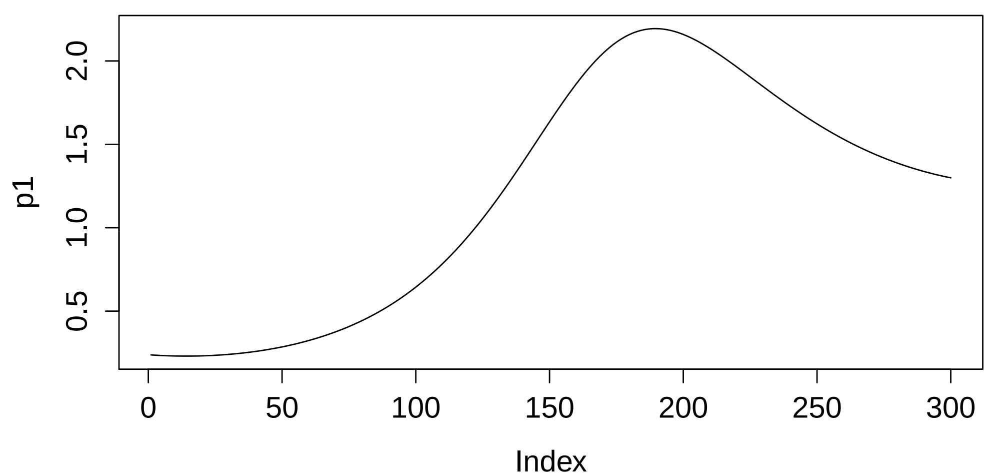


Figure 2: Spectral density obtained using time series modelling

In time series approach to calculate psd, we inherently assume $e[k]$ is deterministic signal. Hence we wrote the equation

$$\Phi_{vv}(\omega) = \left|H(e^{jw})\right|^2 \Phi_{ee}(\omega)$$

It is not advisable to use time series approach since $e[k]$ is random. Also, the model cannot be estimated correctly which gives error in estimating spectral density. Hence, Wiener-Khinchin method of calculating spectral density is advisable. But the problem is that ACVF estimates are noisy, hence spectral density estimate obtained using Wiener-Khinchin theorem is very noisy.

1.2

1500 samples of data is model is simulated and spectral densities are calculated using both the methods. The routine to estimate spectral density using Wiener-Khinchin theorem is given below

```r
# simulate given arima model
data1=arima.sim(n=1500,list(ar=c(0,0.25),ma=c(0.5)))
# Estimate acf of given data
autocorr=acf(data1,lag.max=200)
#extract coefficient values from list
autocorr=autocorr$acf
# Perform fourier transform
Fourier=fft(autocorr)
plot(abs(Fourier)^2)
```

Spectral density estimated using time-series approach is given below The process is modelled using arima command in R as
arima(data1, order = c(2, 0, 1))

Coefficients:

	ar1	ar2	ma1
	-0.0890	-0.2325	0.4644
s.e.	0.0745	0.0344	0.0064

AR(1) coefficient can be neglected as the confidence interval contain zero. The spectral density is estimated using the time series approach. The spectral density calculated using both the methods are shown in Figures 3 and 4. In previous example, we calculated acf upto lag of 200. If we change maximum lag to 50, the spectral density is shown in Figure 5. The spectral density estimation depends on maximum lag up to which acf is calculated. It is evident form Figures 3 and 5.

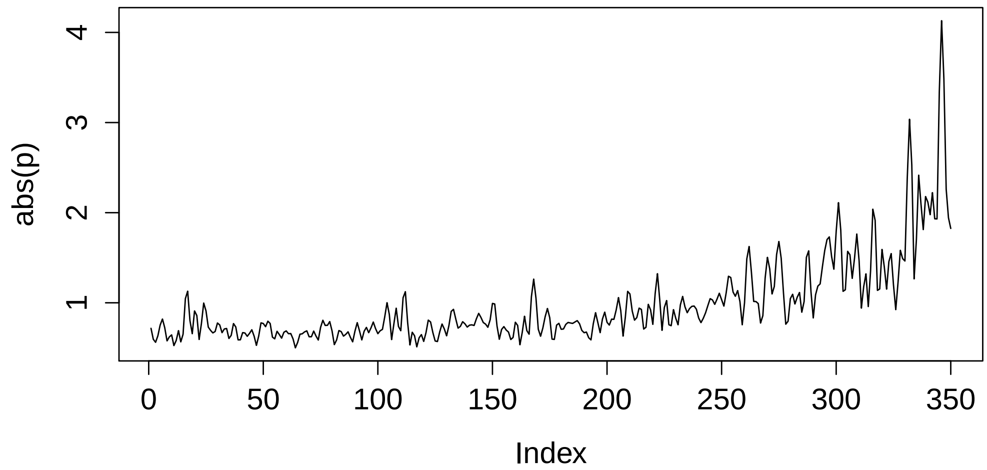


Figure 3: Spectral density obtained using Wiener-Khinchin theorem using ACF calculated upto 200 lags

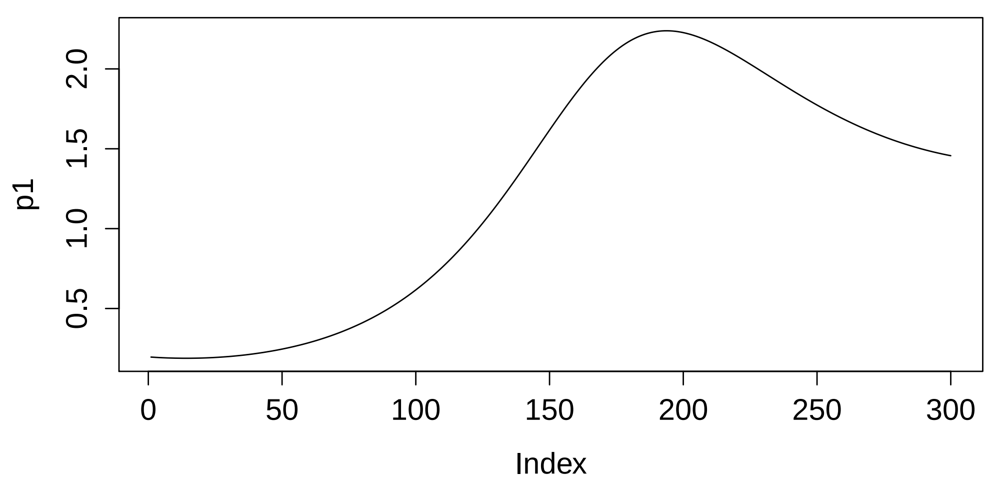


Figure 4: Spectral density obtained using times series modelling

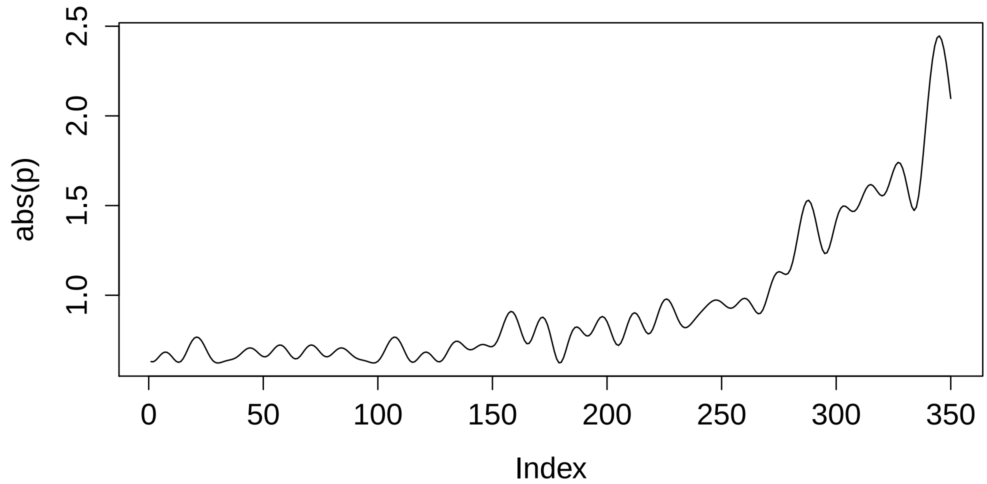


Figure 5: Spectral density obtained using Wiener-Khinchin theorem using ACF at 50 lags

2 Coherency

Given $y[k] = H(q^{-1})u[k] + v[k]$ We know coherence $k_{yu}(f)$ is given by

$$k_{yu}(f) = \frac{|\Phi_{yu}(f)|^2}{\Phi_{yy}(f)\Phi_{uu}(f)}$$

Give model is

$$\begin{aligned} y[k] &= H(q^{-1})u[k] + v[k] \\ \sigma_{yu}[l] &= H(q^{-1})\sigma_{uu}[l] + \cancel{\sigma_{vu}[l]} \\ \implies \Phi_{yu}(f) &= \left|H(e^{j\omega})\right|^2 \Phi_{uu}(f) \end{aligned}$$

We know

$$\Phi_{yy}(f) = \left|H(e^{j\omega})\right|^2 \Phi_{uu}(f) + \Phi_{vv}(f)$$

Hence, coherence

$$
\begin{aligned}
k_{yu}(f) &= \frac{\left|H(e^{j\omega})\right|^2 \Phi_{uu}^2(f)}{\left(\left|H(e^{j\omega})\right|^2 \Phi_{uu}(f) + \Phi_{vv}(f)\right)\Phi_{uu}(f)} \\
&= \frac{\left|H(e^{j\omega})\right|^2 \Phi_{uu}(f)}{\left(\left|H(e^{j\omega})\right|^2 \Phi_{uu}(f) + \Phi_{vv}(f)\right)} \\
&= \frac{1}{1 + \dfrac{\Phi_{vv}(f)}{\left|H(e^{j\omega})\right|^2 \Phi_{uu}(f)}} \\
&= \frac{1}{1 + \dfrac{1}{SNR(f)}}
\end{aligned}
$$

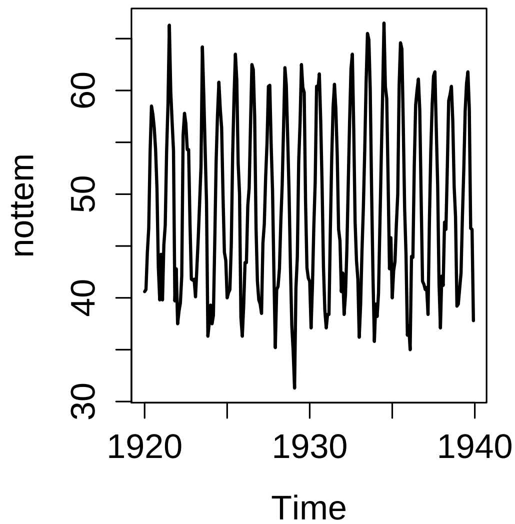
where $SNR(f) = \dfrac{\left|H(e^{j\omega})\right|^2 \Phi_{uu}(f)}{\Phi_{vv}(f)}$. Hence coherence is inversely proportional to $\Phi_{vv}(f)$, which means as the variance of $v[k]$ increases, coherence decreases.

3 Modelling periodicities

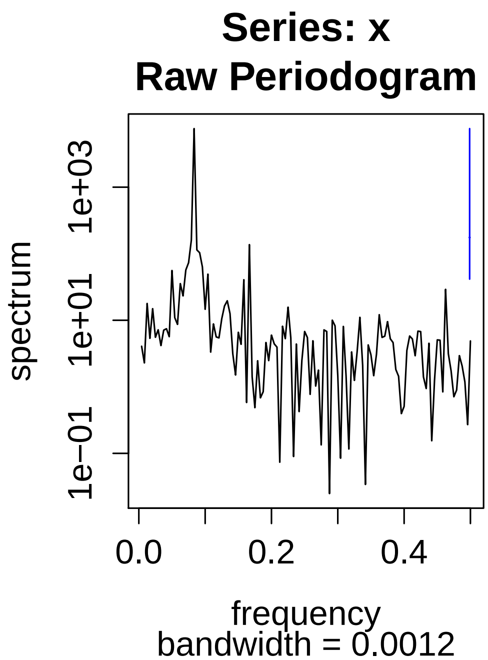
The R-code for modelling nottem data is shown below

```r
# Loading the data
 data('nottem')
 xk=ts(nottem)
 plot(xk,type='l')
 xk=xk-mean(xk)
 k=seq(0,239,1)
 spectrum(xk)
 # Period is 12 samples
 mod_per <- lm (xk ~ cos(2*pi*k/12) + sin(2*pi*k/12) - 1)
 # Extracting residuals
 resk=mod_per$residuals
 plot(resk,type='l')
 acf(resk)
 # Acf in significant at only one lag. Hence AR(1) model is fit for the data
 mod_ar=arima(resk,c(1,0,0))
 acf(mod_ar$residuals)
 pacf(mod_ar$residuals)
```

The data is shown in Figure 6a. The periodogram of the data is shown in Figure 6b. The



(a) nottem Data



(b) Spectrum of data

data is periodic with a period of 12 samples. To remove periodicities the data is fitted with

$$x[k] = A\cos(2 * pi * k/12) + B\sin(2 * pi * k/12)$$

The periodic signal is obtained as

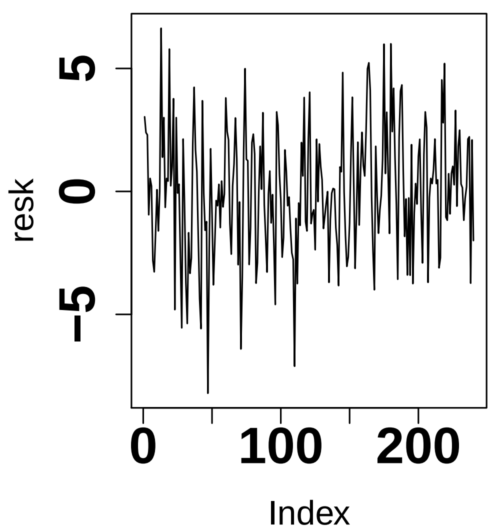
```
Call:
lm(formula = xk ~ cos(2 * pi * K/12) + sin(2 * pi * k/12) - 1)

Coefficients:
cos(2 * pi * K/12)  sin(2 * pi * k/12)
           -11.473              -1.391
```

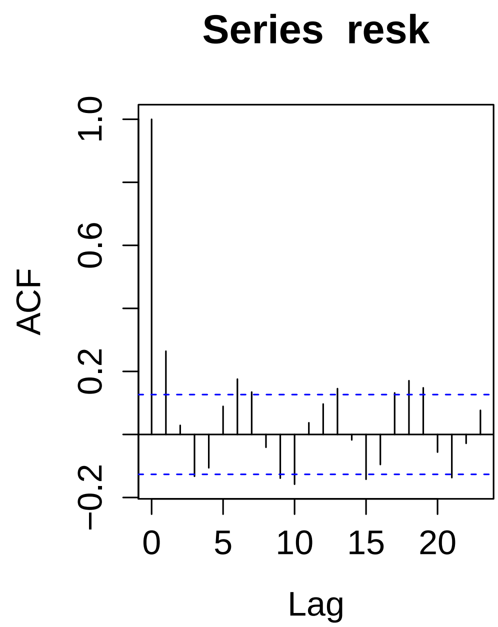
The residuals are shown in Figure 6a. The acf plot of the residuals is shown in the Figure 6b. From acf plot of residuals AR(1) model is chosen to fit for the process. The model is given by Call: arima(x = resk, order = c(1, 0, 0))

Coefficients: ar1 0.2655 ($\pm$0.064) s.e. 0.0623

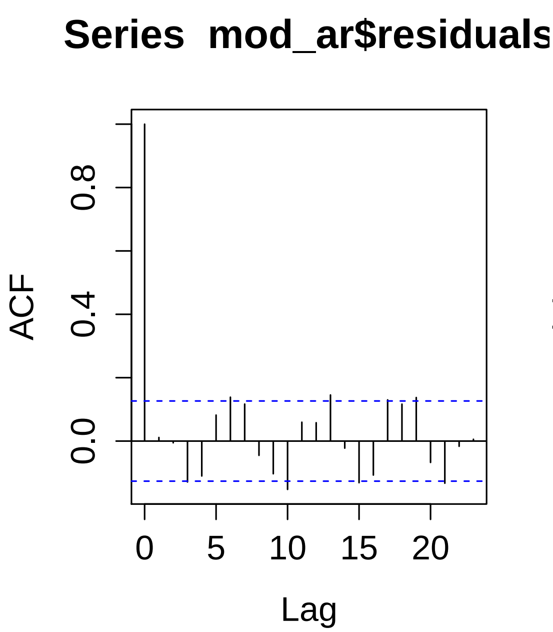
The acf and pacf plot of residuals is shown in Figures The final model is given by



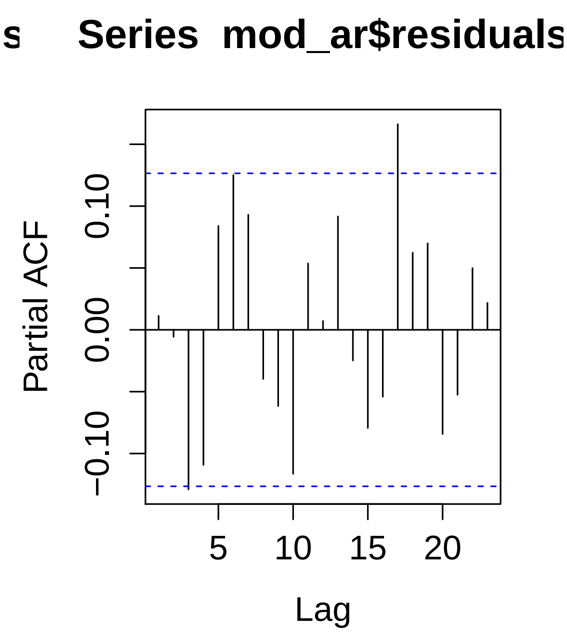
(a) Residual plot



(b) ACF plot of residuals



(a) ACF plot of residuals



(b) PACF plot of residuals

$$x[k] = -11.473cos(2*pi*K/12) - 1.391sin(2*pi*k/12) + v[k]$$

$$\text{where} \;\; v[k] = 0.2655v[k-1] + e[k]$$

4 Variability of sample mean

Sample mean

$$\bar{x} = \frac{1}{N}(x[0] + x[1] + ... + x[N-1])$$

Variance of sample mean is given by

$$\begin{aligned}
\text{var}(\bar{x}) &= \frac{1}{N^2}\left(\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \;\text{cov}\,(x[i]x[j])\right) \\
&= \frac{1}{N^2}\left(\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \;\sigma_{xx}[|i-j|]\right) \\
&= \frac{1}{N^2}\left(N\sigma_{xx}[0] + \sum_{i=0}^{N-1}\sum_{j=0,i\neq j}^{N-1} \;\sigma_{xx}[|i-j|]\right) \\
&= \frac{1}{N^2}\left(N\sigma_{xx}[0] + \sum_{i=0}^{N-1}\sum_{j=0,i\neq j}^{N-1} \;(N-l)\,\sigma_{xx}[l]\right)
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{N^2}\left(N\sigma_{xx}[0] + \sum_{l=-(N-1),l\neq 0}^{N-1} \;(N-l)\,\sigma_{xx}[l]\right) \\
&= \frac{1}{N^2}\left(N\sigma_{xx}[0] + 2\sum_{l=1}^{N-1} \;(N-|l|)\,\sigma_{xx}[l]\right) \\
&= \frac{1}{N}\left(\sigma_{xx}[0] + 2\sum_{l=1}^{N-1} \;\left(1-\frac{|l|}{N}\right)\sigma_{xx}[l]\right)
\end{aligned}$$

Given MA(1) process $x[k] = e[k] + 0.4e[k-1]$. The routine to calculate variance of sample mean by means of Monte-Carlo simulations is given below

```
# Intialise sample mean values to zero
mea=rep(0,1000);
#Outer loop for doing Monte-Carlo simulations
for (k1 in 1:1000){
```

```
  e=rnorm(1000)
  x=rep(0,1000)
# Inner loop for generating MA(1) data
for (k in 1:999){
  x[k+1]=e[k+1]+0.4*e[k]
}
# Calculate sample mean value
mea[k1]=mean(x)
}
auto=acf(x,type='covariance')
auto=auto$acf
# Calculate variance of sample mean using derived formula
# ACVF values at remaining lags neglected
varxbar=0.001*(auto[1]+2*(0.999*auto[2]+0.998*auto[3]))
# Calculate variance of sample mean using in built command in R
varx=var(mea)

```

The variance of sample mean estimates obtained using in built command in R is 0.0018. The variance of sample mean obtained using derived formula is 0.0021. Hence variance of sample mean estimates found to be identical.

5 Fisher's information matrix

Given exponential distribution

$$f(y) = \lambda e^{-\lambda y}$$

For the estimator to be efficient the function

$$\hat{\theta} = \frac{s(\theta)}{I(\theta)} + \theta$$

is to be independent of theta.

5.1 $\hat{\theta} = \lambda$

The log-likelihood function is given by

$$L = \ln f(y) = \ln \lambda - \lambda y$$

Assuming the parameter to be estimated is $\theta = \lambda$, the score function is obtained as

$$S(\lambda) = \frac{\partial L}{\partial \theta} = \frac{1}{\lambda} - y$$

The information matrix is obtained as

$$\begin{aligned} I(\lambda) &= -E(\frac{\partial^2 L}{\partial \lambda^2}) \\ &= \frac{1}{\lambda^2} \end{aligned}$$

Here $\hat{\theta}$ is obtained as

$$\hat{\theta} = \lambda^2 y$$

which is a function of λ. Hence there exists no efficient estimator to estimate λ.

5.2 $\hat{\theta} = \frac{1}{\lambda}$

The log-likelihood function is given by

$$L = \ln f(y) = \ln \lambda - \lambda y$$

Assuming the parameter to be estimated is $\theta = \frac{1}{\lambda}$, the score function is obtained as

$$S(\frac{1}{\lambda}) = \frac{\partial L}{\partial \theta} = -\lambda + \lambda^2 y$$

The information matrix is obtained as

$$\begin{aligned} I(\lambda) &= -E(\frac{\partial^2 L}{\partial \lambda^2}) \\ &= \lambda^2 \end{aligned}$$

Here $\hat{\theta}$ is obtained as

$$\hat{\theta} = y$$

which is a not a function of $\frac{1}{\lambda}$. Hence there exists an efficient estimator to estimate $\frac{1}{\lambda}$.