1. Simulate an ARMA(1,1)process with ϕ=0.6, θ=-0.4 and μ=100. Simulate 60 values but set aside the last 10 values to compare forecasts with actual values.

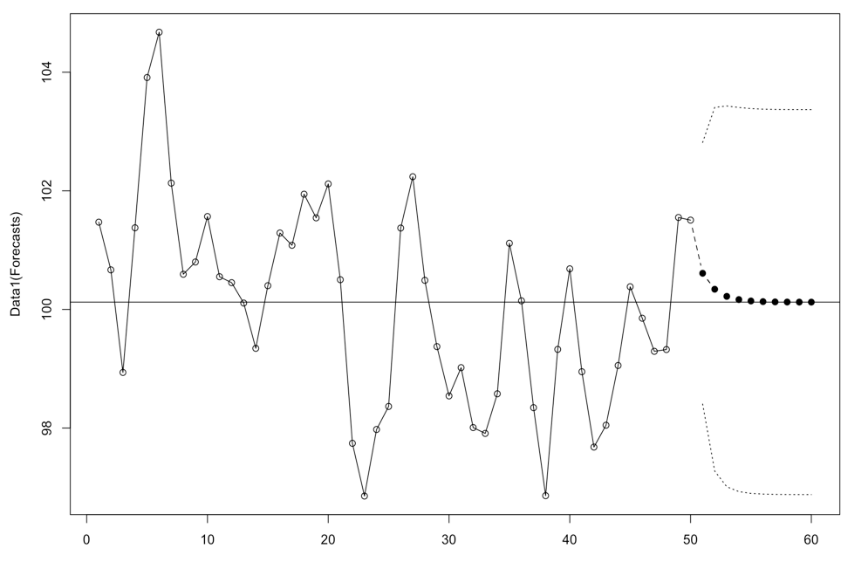
```
> set.seed(15716)
> data1=arima.sim(n=60,list(ar=0.6,ma=0.4))+100
> data1.2nd=window(data1,start=51)
```

a) Using the first 50 values of the series, find the values for the maximum likelihood estimate of ϕ and θ.

```
> ### Q1.a
> data1.1st=window(data1,end=50)
> model1=arima(data1.1st,order = c(1,0,1),method = "ML")
> model1
      Call:
      arima(x = data1.1st, order = c(1, 0, 1), method = "ML")
      Coefficients:
              ar1     ma1  intercept
           0.4468  0.5190   100.1231
      s.e.  0.1461  0.1169     0.4279
      sigma^2 estimated as 1.266:  log likelihood = -77.32,  aic = 160.64
```

b) Using the estimated model, forecast the next 10 values of the series. Plot series together with the 10 forecasts. Place a horizontal line at the estimate of the process mean.

```
> model1.FC=plot(model1,n.ahead=10,ylab = 'Data1(Forecasts)',pch=19)
> abline(h=coef(model1)[names(coef(model1))=='intercept'])
```



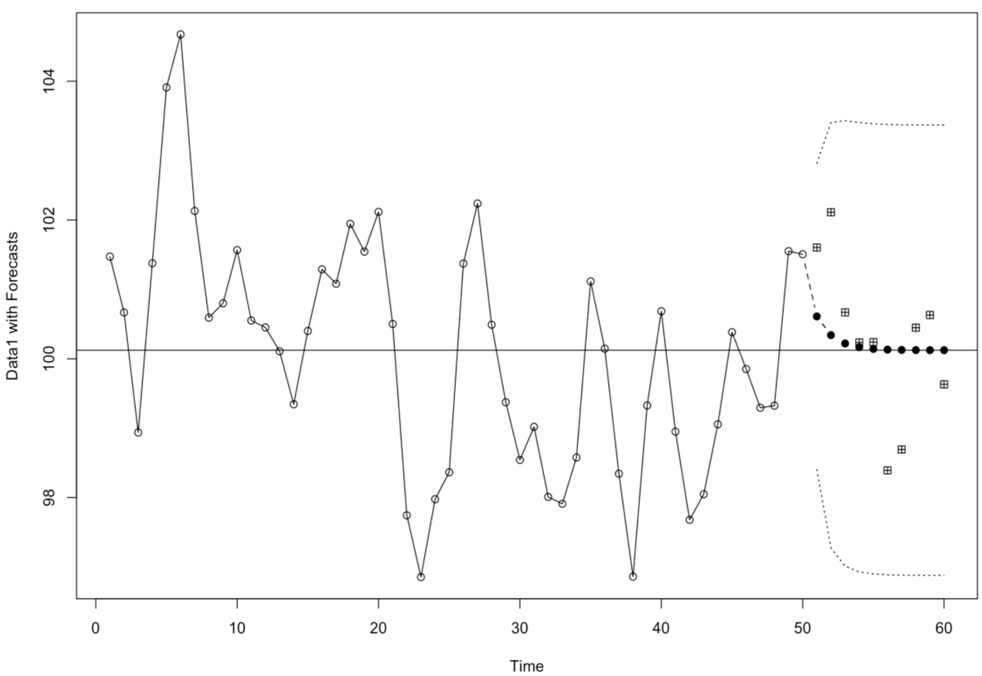

```
> model1.FC
      Time Series:
      Start = 51
      End = 60
      Frequency = 1
       [1] 100.6098 100.3406 100.2203 100.1665 100.1425 100.1318 100.1270 100.1248 100.1239
      [10] 100.1235
```

c) Compare 10 forecasts with the actual values you set aside.

```
> AC.FC=cbind(data1.2nd,model1.FC)
> AC.FC
        Time Series:
        Start = 51
        End = 60
        Frequency = 1
          data1.2nd      model1.FC
        51  101.60334     100.6098
        52  102.11194     100.3406
        53  100.66867     100.2203
        54  100.23541     100.1665
        55  100.24066     100.1425
        56   98.39140     100.1318
        57   98.69198     100.1270
        58  100.44735     100.1248
        59  100.63078     100.1239
        60   99.63098     100.1235
```

d) Plot the forecasts together with 95% forecast limits. Do the actual values fall within the forecast limits?

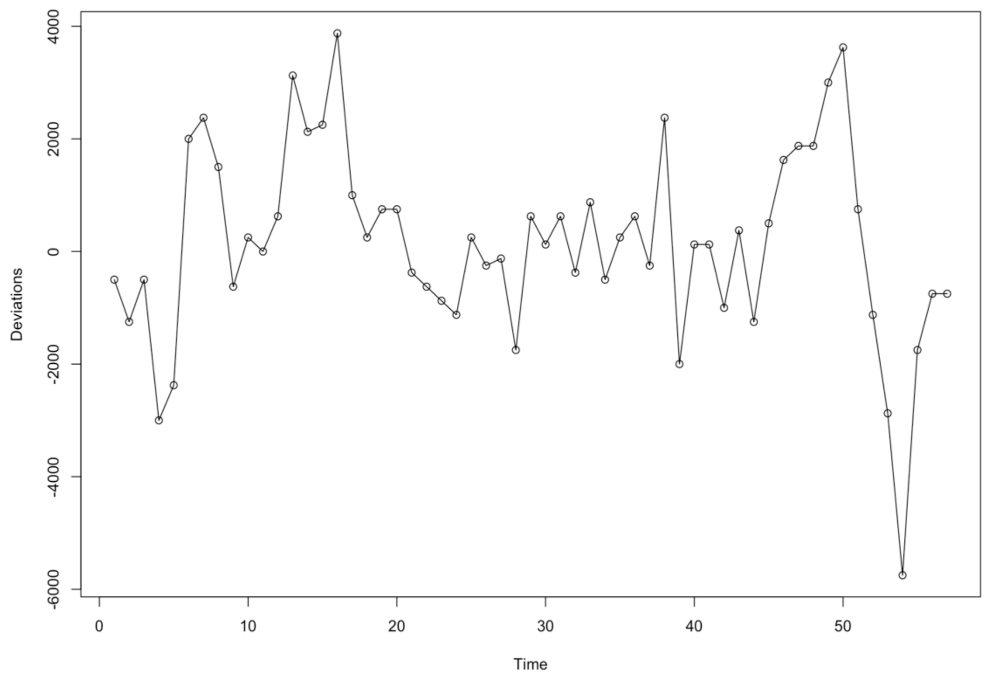
```
> plot(model1,n.ahead=10,ylab = 'Data1 with Forecasts',pch=19)
> points(data1.2nd,pch=12)
> abline(h=coef(model1)[names(coef(model1))=='intercept'])
```



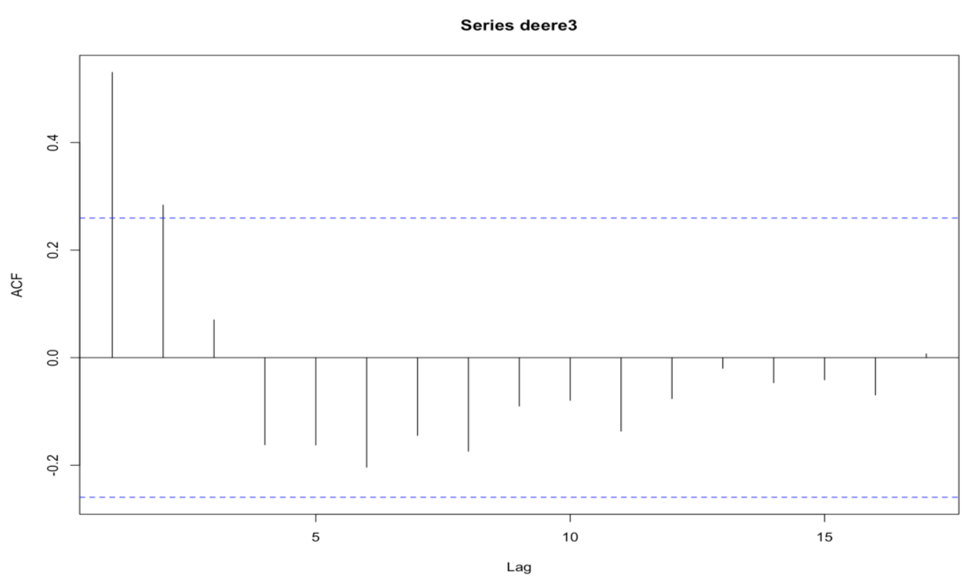
2. The data file deere3 contains 57 consecutive values from a complex machine tool process at Deere & Co. The values given are deviations from a target value in units of ten millions of an inch. The process employs a control mechanism that resets some of the parameters of the machine tool depending on the magnitude of deviation from target of the last item produced.

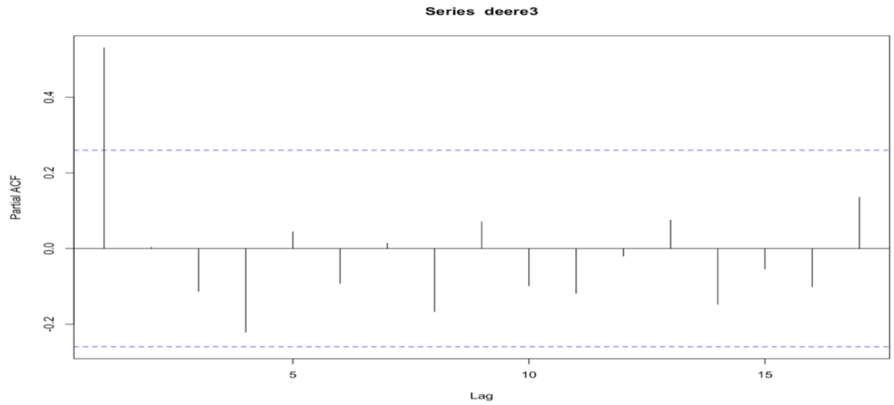
a) Make a time series plot for the data set. Comment on your observations.

```
> data(deere3)
> plot(deere3, type='o', ylab='Deviations')
```



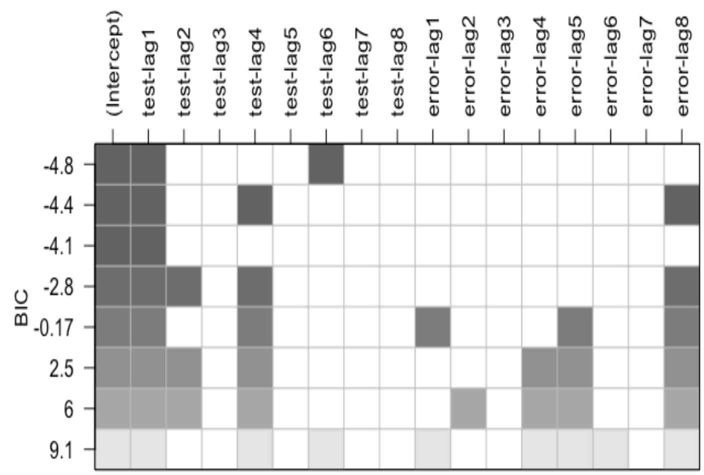
b) Plot the ACF and PACF of the data set. Comment on the plots





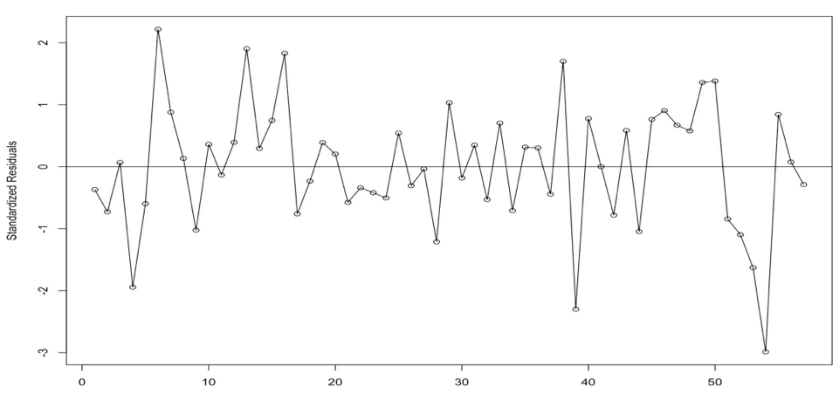
c) Identify the appropriate ARMA model to fit the data and give the fitted model.

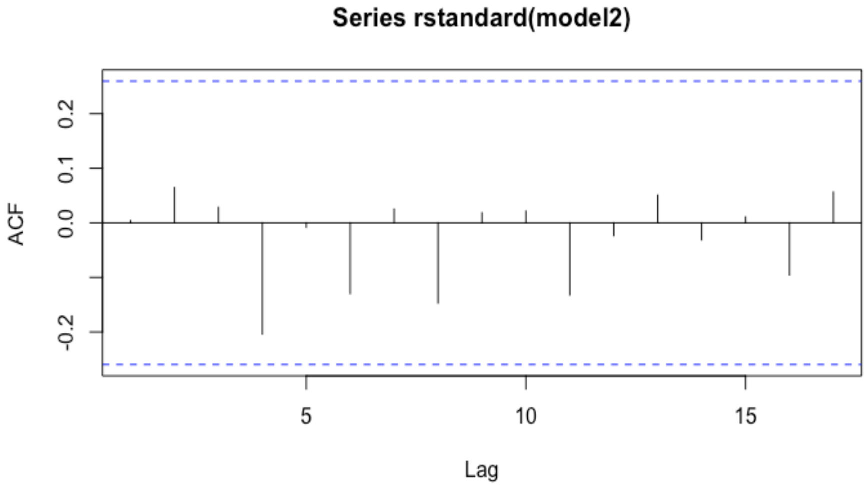
```
> eacf(deere3)
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x x o o o o o o o o o o  o  o
1 o o o o o o o o o o o o  o  o
2 o o o o o o o o o o o o  o  o
3 x o o o o o o o o o o o  o  o
4 o x o o o o o o o o o o  o  o
5 o x o x o o o o o o o o  o  o
6 o x o x o o o o o o o o  o  o
7 o x x o o o o o o o o o  o  o
```

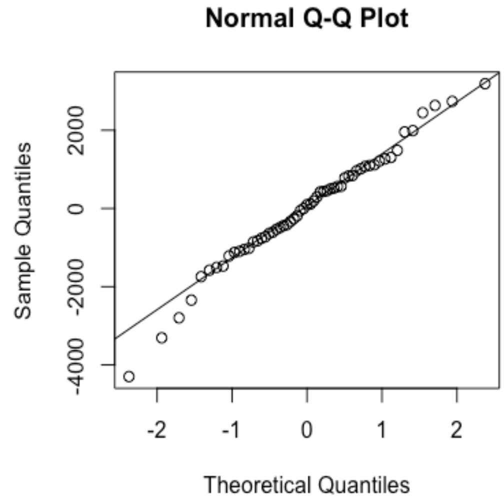



```
> res=armasubsets(y=deere3,nar=8,nma=8,y.name='test',ar.method='ols'); plot(res)
> arima(deere3,order = c(1,0,0),method="ML")
      ar1  intercept
   0.5256   124.3524
s.e.  0.1108   394.2320
sigma^2 estimated as 2069354:  log likelihood = -495.51,  aic = 995.02
```

d) Use appropriate plots of standard residuals to do model checking and comment on those plots.





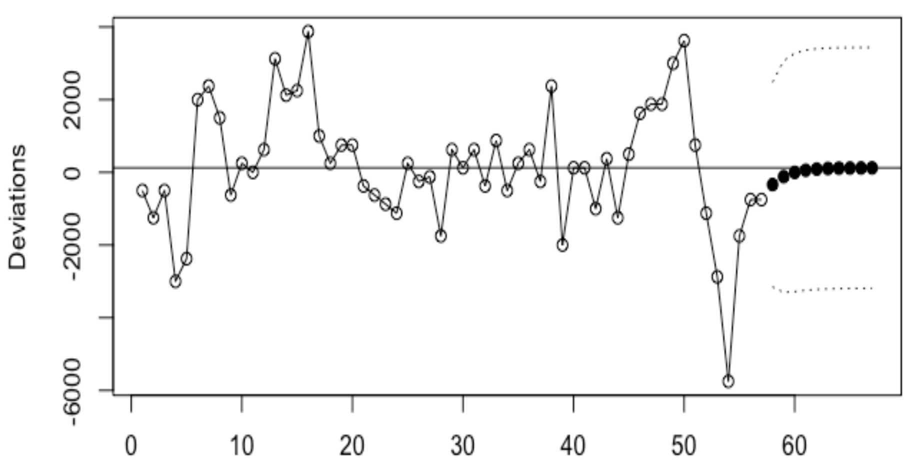


e) Use the fitted model to forecast the next 10 values.

```
> model2.FC=plot(model2,n.ahead=10)
> model2.FC$pred
    Time Series:
    Start = 58
    End = 67
    Frequency = 1
     [1] -335.145928 -117.120772   -2.538388   57.679997   89.327566  105.959839  114.700873
     [8]  119.294695  121.708962  122.977772
```

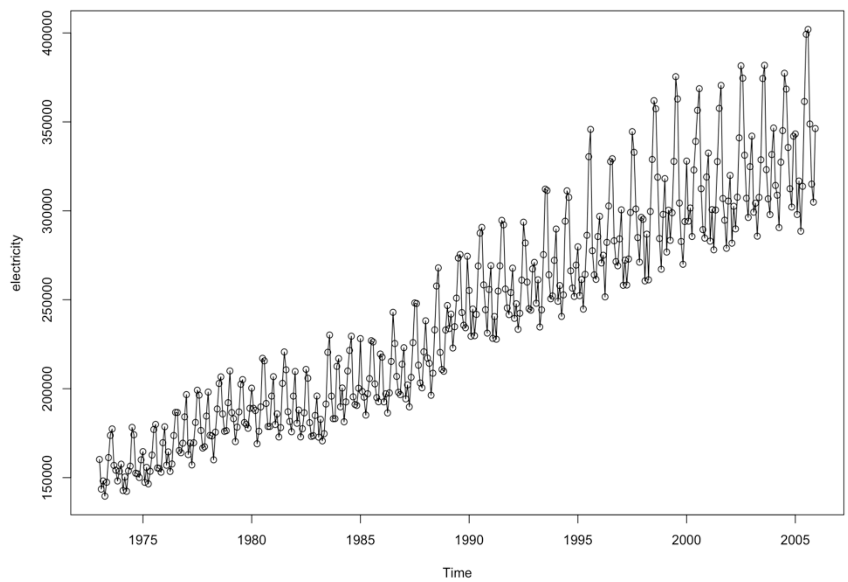
f) Plot the time series, the forecasts and 95% forecast limits, and interpret the results.

```
> plot(model2,n.ahead = 10,ylab = 'Deviations',xlab='Year',pch=19)
> abline(h=coef(model2)[names(coef(model2))=='intercept'])
```

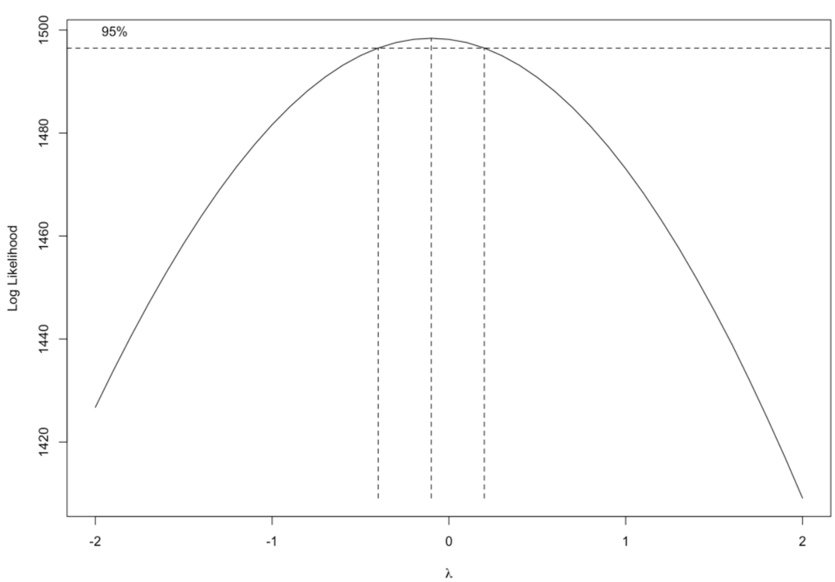


3. **Data file electricity contains the time series {Yt} of total monthly electricity generated in the United States in millions of kilowatt-hours.**

a) **Display and interpret the time series plot for the data set.**

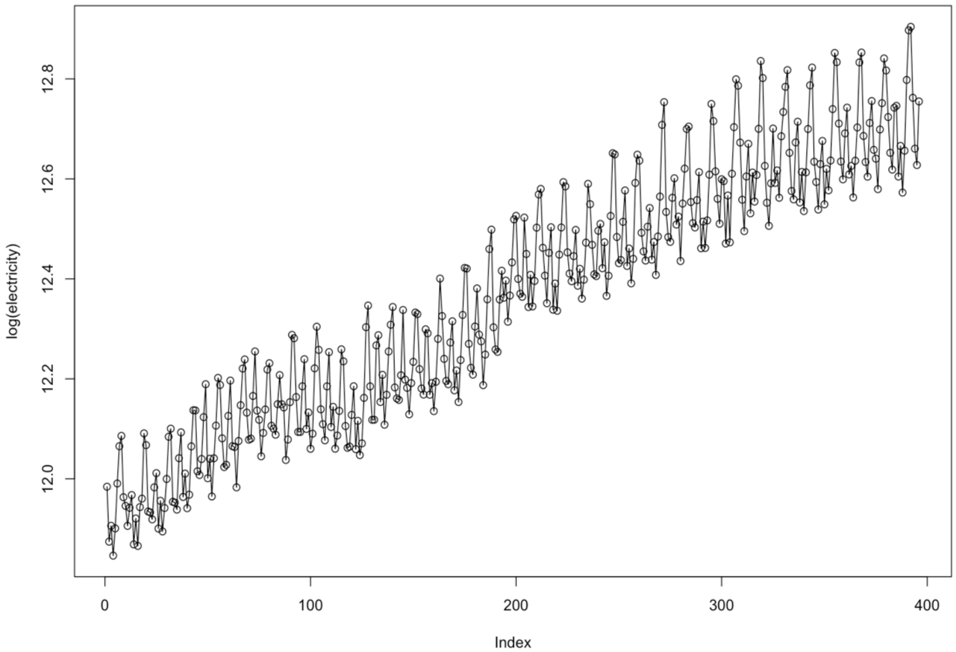


b) **Use software to produce a plot similar to Exhibit 5.11, on page 102, and determine the "best" value of λ for a power transformation of the data.**



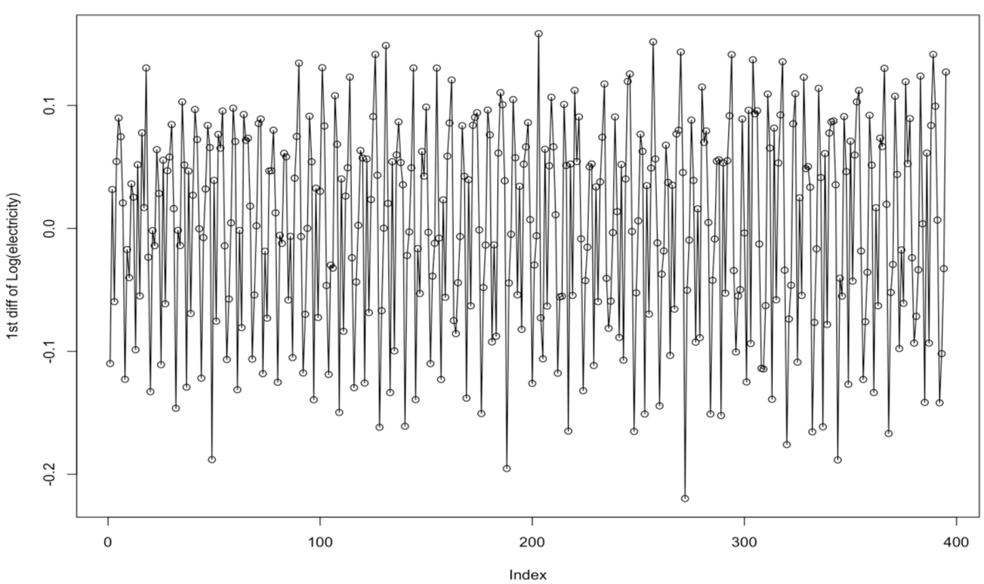
c) Produce a time series plot of the transformed data {Wt} and interpret the plot.

```
> W=log(as.vector(electricity))
> plot(W,type='o',ylab='log(electricity)')
```

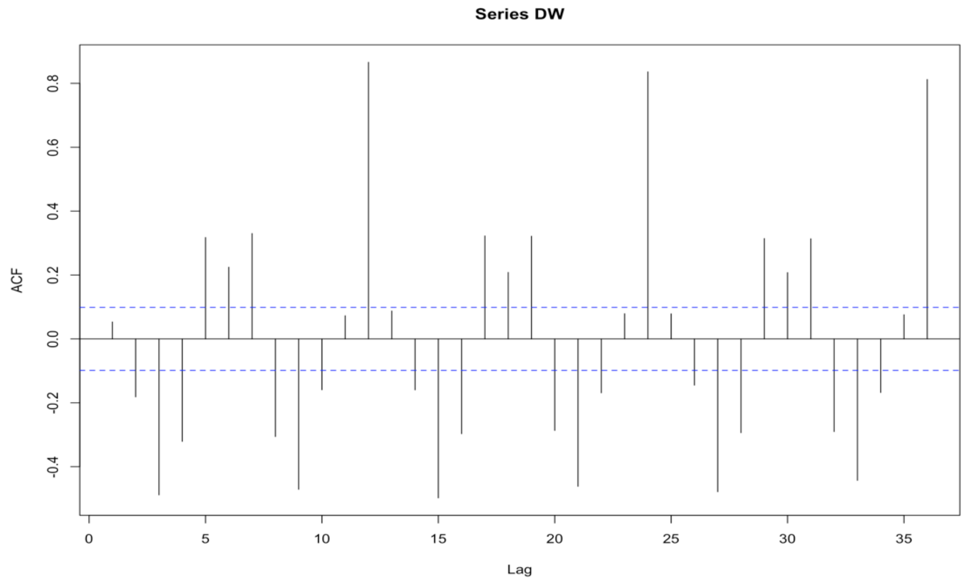


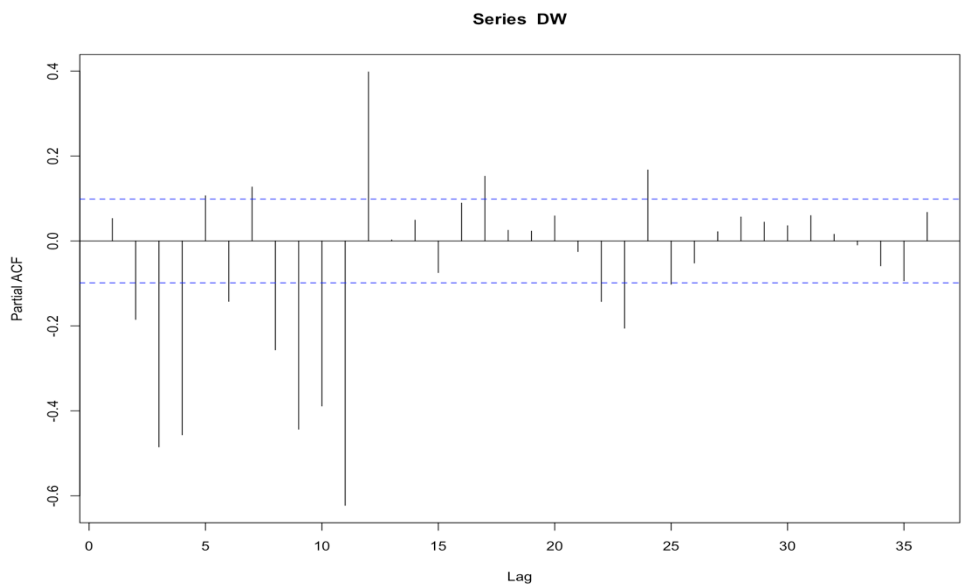
d) Make a time series plot of the the first difference of the transformed data {Wt}, {∇Wt}.

```
> DW=diff(W)
> plot(diff(W),type='o',ylab = '1st diff of Log(electricity)')
```



e) Caculate the sample ACF and PACF of {∇Wt}. Is the seasonality visible in this display? If so, what is the period s of the seasonality?

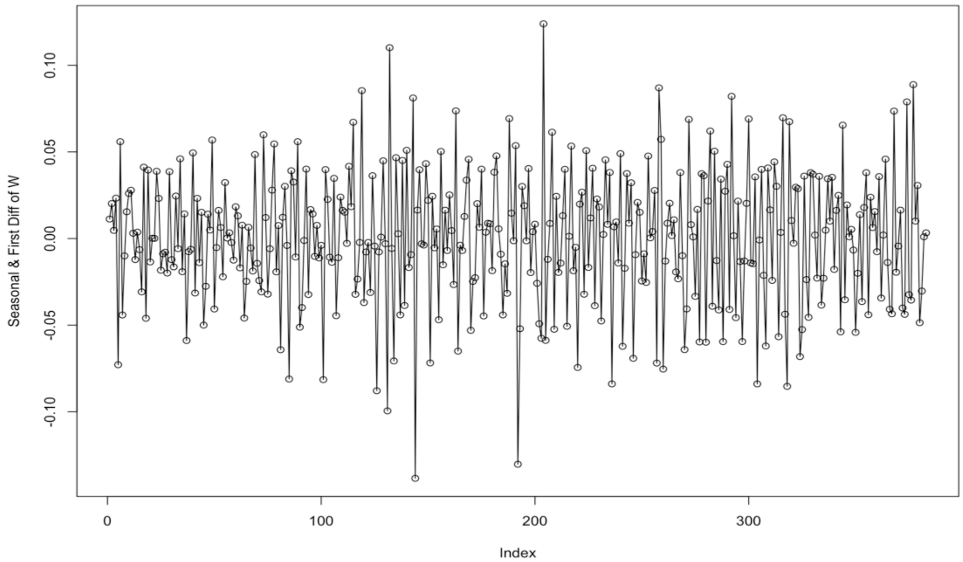




f) Plot the time series of seasonal difference and first difference of the transformed series, i.e. ∇s∇Wt.

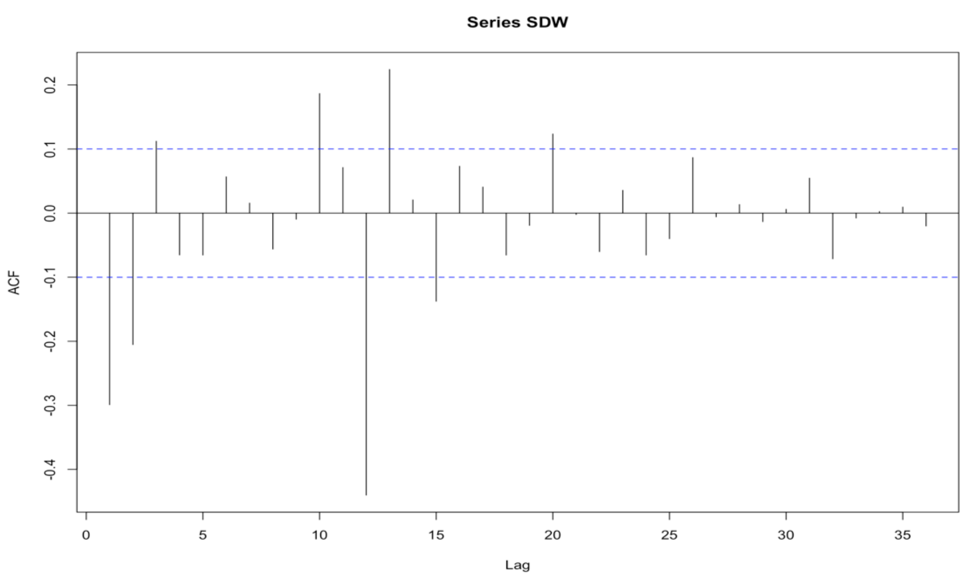
> SDW=diff(DW,lag = 12)

> plot(SDW,type = 'o',ylab = 'Seasonal & First Diff of W')

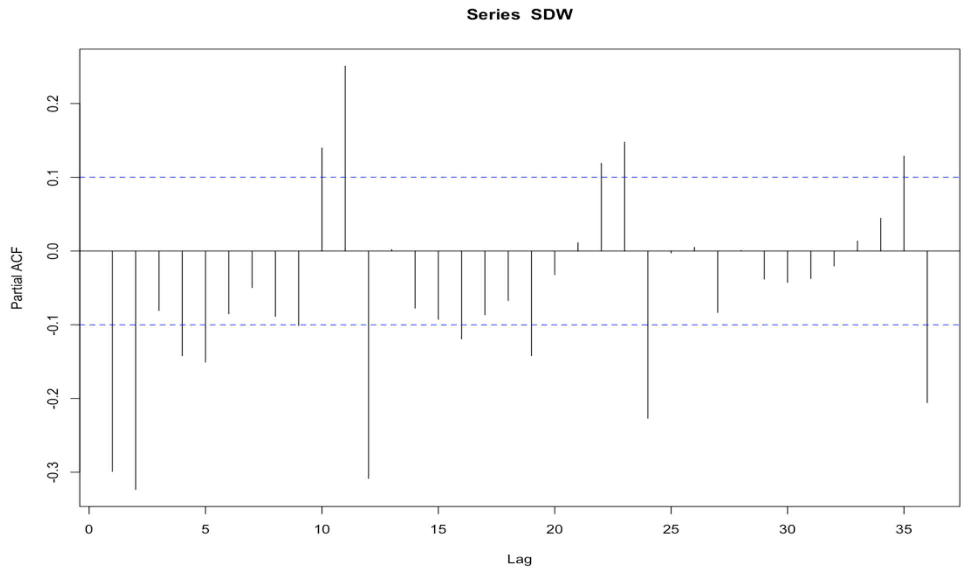


g) Display the sample ACF and PACF of ∇s∇Wt. Does a stationary model seem appropriate for ∇s∇Wt?

> acf(SDW,lag.max = 36)

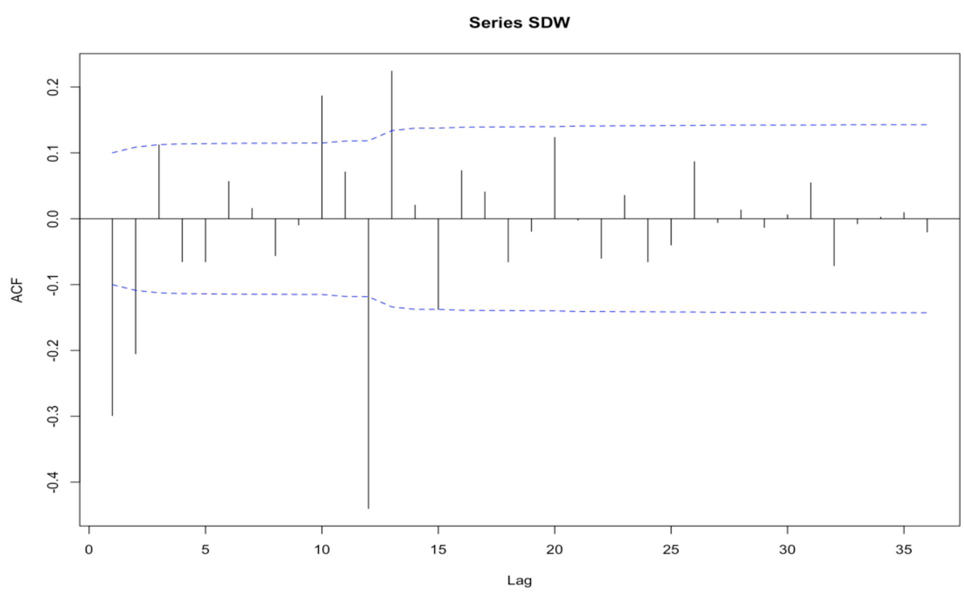


```
>pacf(SDW,lag.max = 36)
```



h) What model might you consider for the electricity series?

```
> acf(SDW,lag.max = 36,ci.type='ma')
```



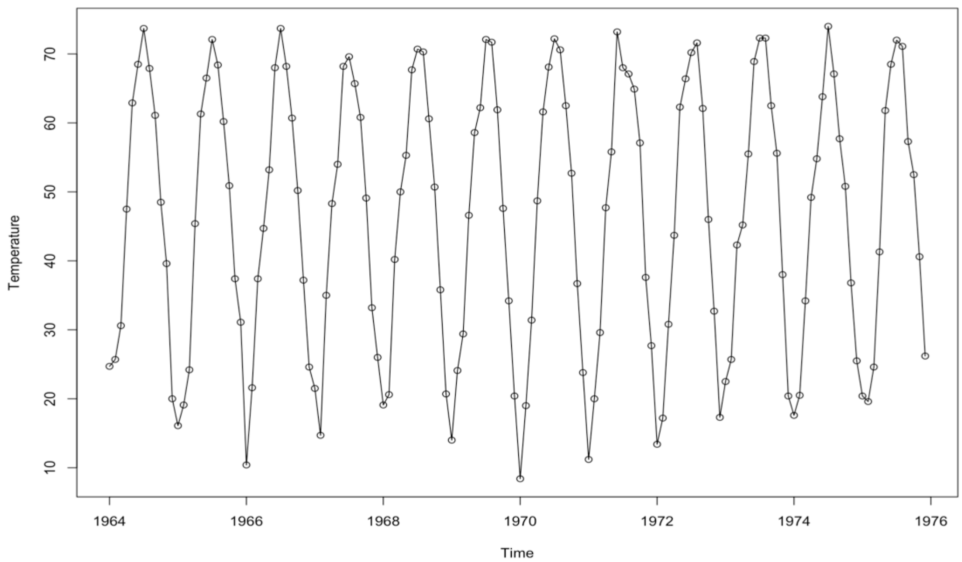
i) Estimate the model.

```
> Model3=arima(W,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=12))

> Model3

Call:

arima(x = W, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12))

Coefficients:

        ma1     sma1

     -0.5049  -0.8299

s.e.   0.0753   0.0319

sigma^2 estimated as 0.0007344:  log likelihood = 831.35,  aic = -1658.7
```

4. Our textbook uses data set tempdub several times. Make a complete R script based on the codes in oir textbook for analyzing the data set, including identifying the appropriate model to fit the data set. Estimating the model, and using the model for forecasting.

a) Exhibit 1.7

```
> data("tempdub")
> plot(tempdub,ylab='Temperature',type='o')
```



b) Exhibit 3.3

```
> model.2=lm(tempdub~month.-1)
> summary(model.2)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
month.January	16.608	0.987	16.83	<2e-16	***
month.February	20.650	0.987	20.92	<2e-16	***
month.March	32.475	0.987	32.90	<2e-16	***
month.April	46.525	0.987	47.14	<2e-16	***
month.May	58.092	0.987	58.86	<2e-16	***
month.June	67.500	0.987	68.39	<2e-16	***
month.July	71.717	0.987	72.66	<2e-16	***
month.August	69.333	0.987	70.25	<2e-16	***
month.September	61.025	0.987	61.83	<2e-16	***
month.October	50.975	0.987	51.65	<2e-16	***
month.November	36.650	0.987	37.13	<2e-16	***
month.December	23.642	0.987	23.95	<2e-16	***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

c) Exhibit 3.4

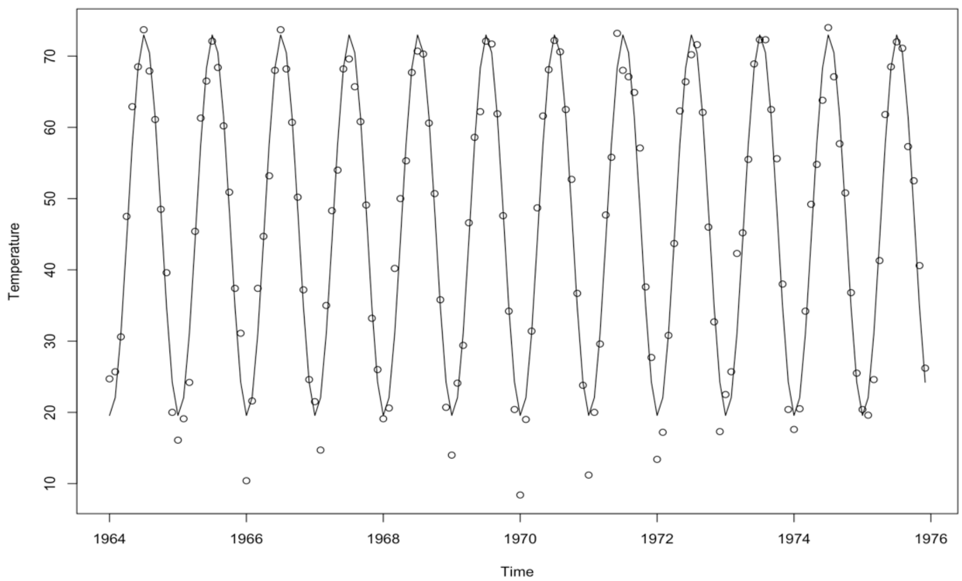
```
> model.3=lm(tempdub~month.)
> summary(model.3)
    Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
    (Intercept)       16.608     0.987  16.828  < 2e-16 ***
    month.February     4.042     1.396   2.896  0.00443 **
    month.March       15.867     1.396  11.368  < 2e-16 ***
    month.April       29.917     1.396  21.434  < 2e-16 ***
    month.May         41.483     1.396  29.721  < 2e-16 ***
    month.June        50.892     1.396  36.461  < 2e-16 ***
    month.July        55.108     1.396  39.482  < 2e-16 ***
    month.August      52.725     1.396  37.775  < 2e-16 ***
    month.September   44.417     1.396  31.822  < 2e-16 ***
    month.October     34.367     1.396  24.622  < 2e-16 ***
    month.November    20.042     1.396  14.359  < 2e-16 ***
    month.December     7.033     1.396   5.039 1.51e-06 ***
    ---
    Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

d) Exhibit 3.5

```
># an example of fitting a cosine curve at the fundamental frequency to the average monthly temp series.
> # cosine-sine trend
> har.=harmonic(tempdub,1)
> model.4=lm(tempdub~har.)
> summary(model.4)
    Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
    (Intercept)       46.2660     0.3088 149.816  < 2e-16 ***
    har.cos(2*pi*t)  -26.7079     0.4367 -61.154  < 2e-16 ***
    har.sin(2*pi*t)   -2.1697     0.4367  -4.968 1.93e-06 ***
    ---
    Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

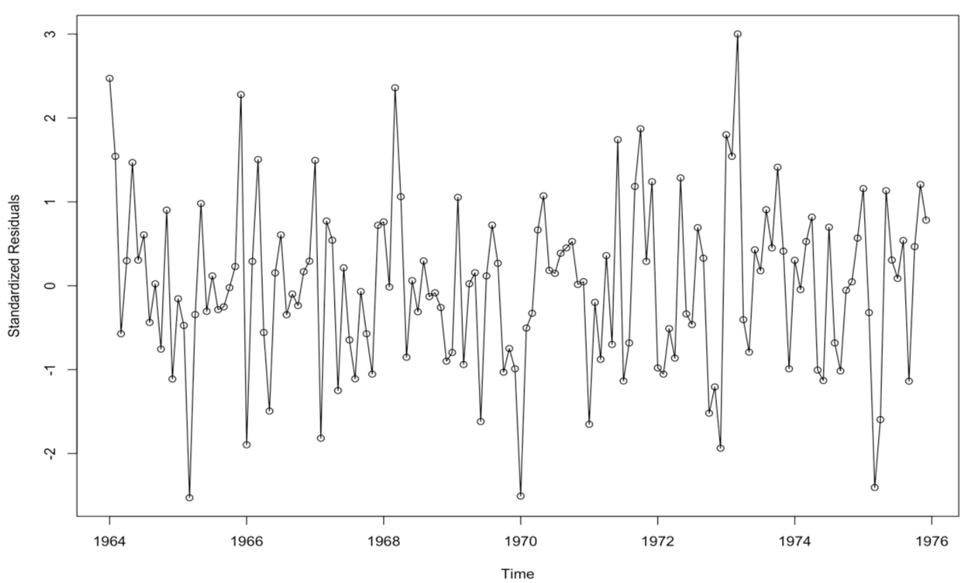
e) Exhibit 3.6

```
> # ylim ensures that the y axis range fits the raw data and the fitted values
> plot(ts(fitted(model.4),freq=12,start=c(1964,1)),
+     ylab='Temperature',type='l',ylim = range(c(fitted(model.4),tempdub)))
> points(tempdub)
```



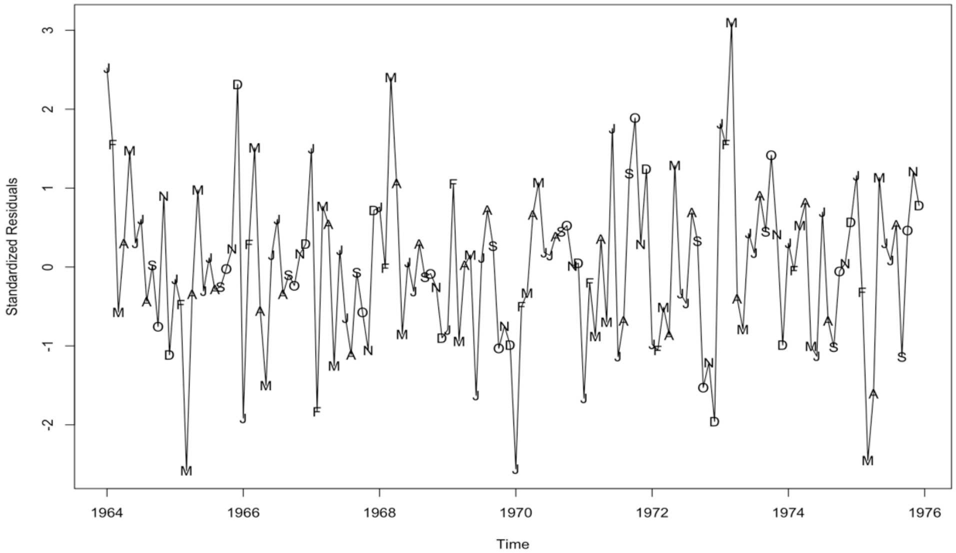
f) Exhibit 3.8

```
> plot(y=rstandard(model.3),x=as.vector(time(tempdub)), xlab='Time',ylab='Standardized Residuals',type = 'o')
```



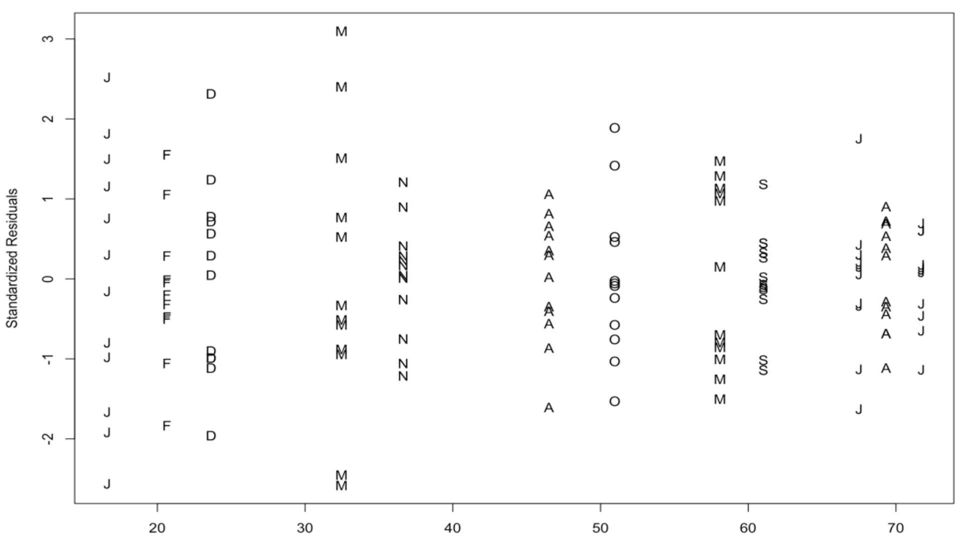
g) Exhibit 3.9

```
> plot(y=rstudent(model.3),x=as.vector(time(tempdub)),xlab='Time', ylab='Standardized Residuals', type='l')
> points(y=rstudent(model.3),x=as.vector(time(tempdub)), pch=as.vector(season(tempdub)))
```



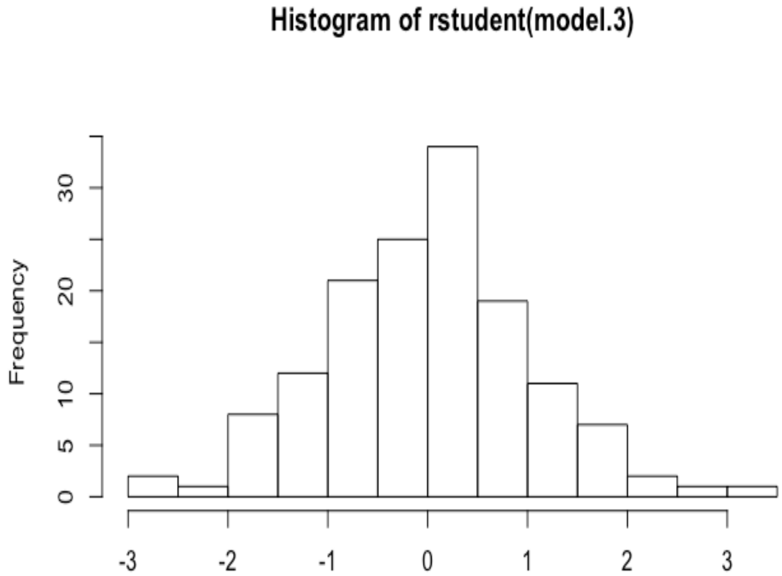
h) Exhibit 3.10

```
> plot(y=rstudent(model.3),x=as.vector(fitted(model.3)),xlab='Fitted Trend Values', ylab='Standardized
Residuals',   type='n')
> points(y=rstudent(model.3),x=as.vector(fitted(model.3)),pch=as.vector(season(tempdub)))
```



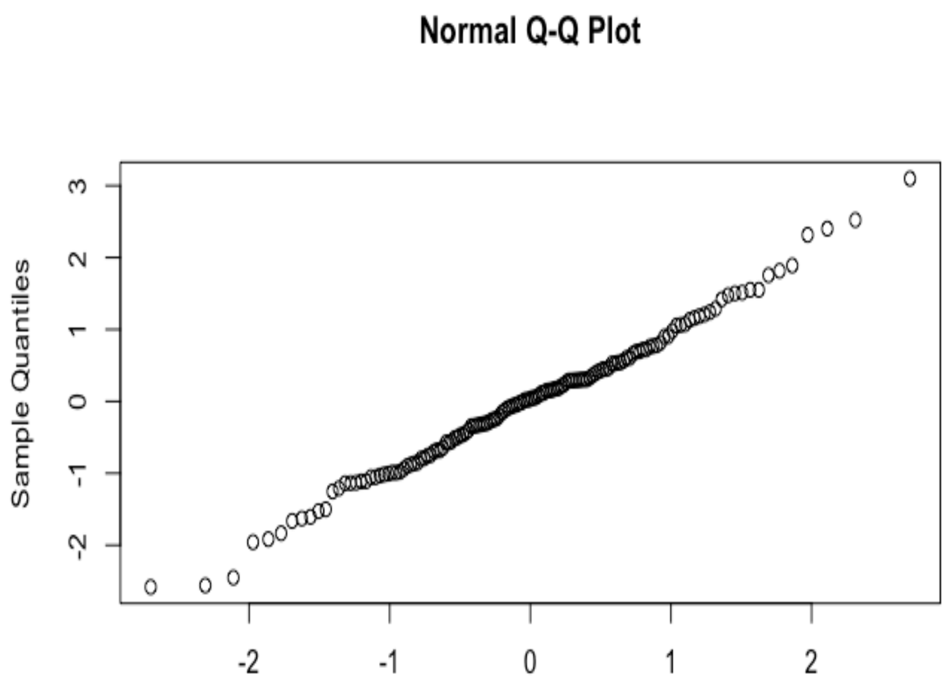
i) Exhibit 3.11

> hist(rstudent(model.3),xlab = 'Standardized Residuals')



j) Exhibit 3.12

> qqnorm(rstudent(model.3))



k) Exhibit 9.2

```
> tempdub1=ts(c(tempdub,rep(NA,24)),start=start(tempdub),freq=frequency(tempdub))
> har.=harmonic(tempdub,1)
> m5.tempdub=arima(tempdub,order=c(0,0,0),xreg = har.)
> newhar.=harmonic(ts(rep(1,24),start = c(1976,1),frequency = 12),1)
> plot(m5.tempdub,n.ahead=24,n1=c(1972,1),newxreg=newhar.,type = 'b',ylab = 'Temperature',xlab='Year')
```