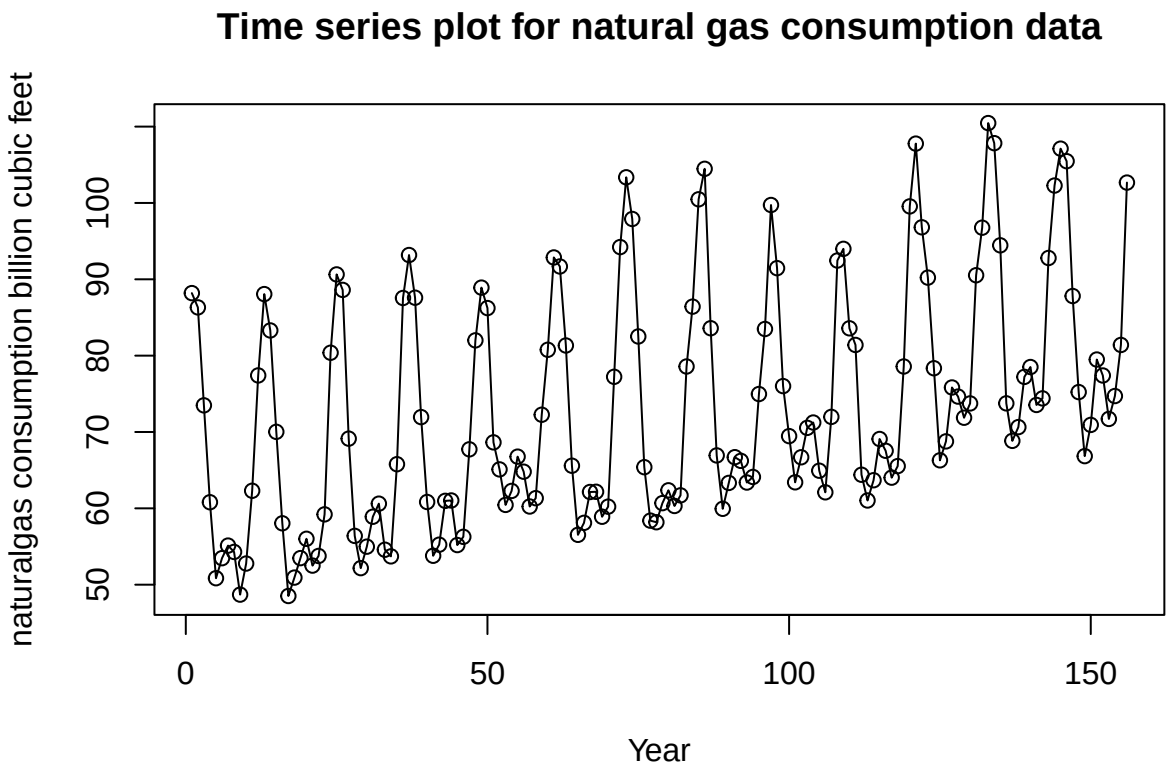
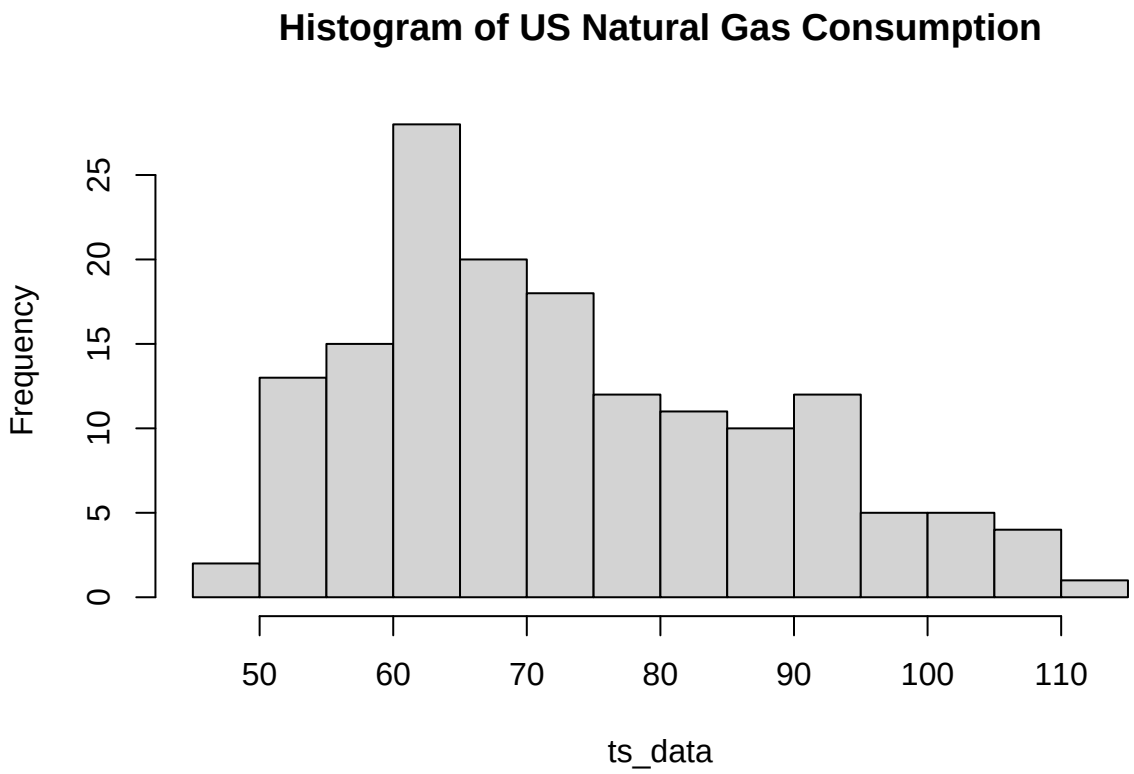
Time Series Analysis of The US Natural Gas Consumption

2024-01-04

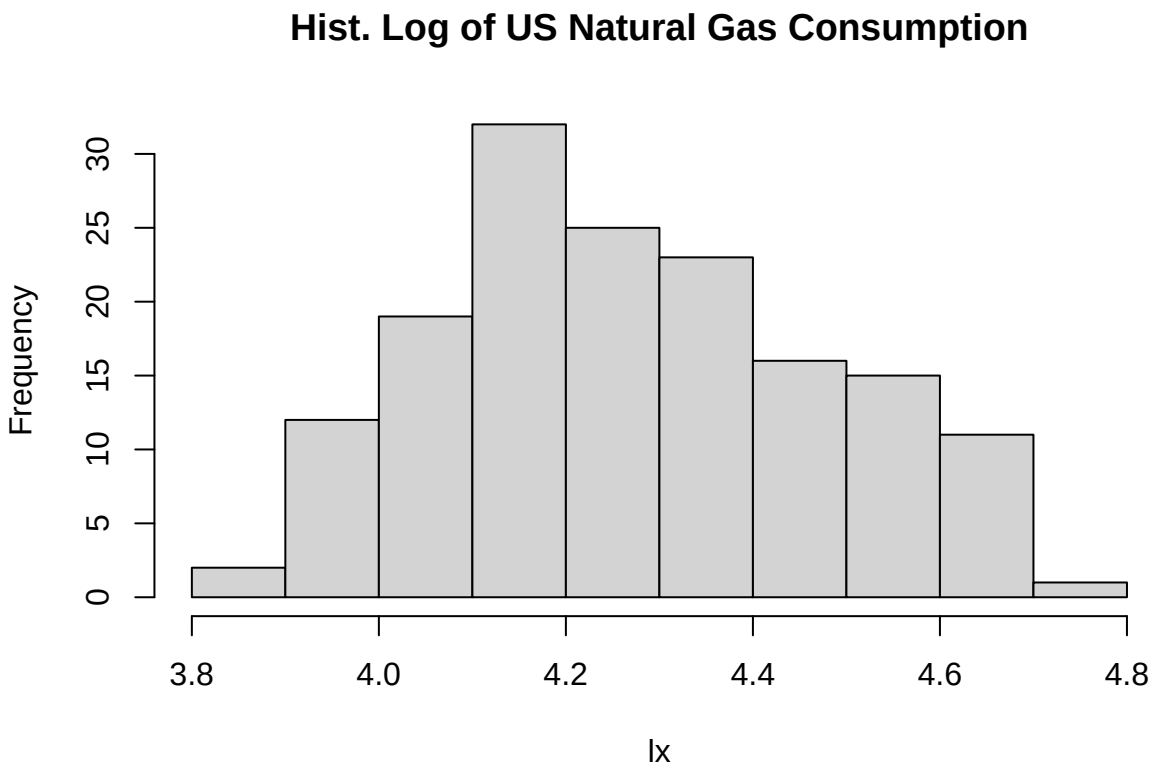
```r
ns_2 <- read.csv("C:/Users/PC/Downloads/ns-2.csv", header = F)

plot(ns_2$V2,type='o',ylab="naturalgas consumption billion cubic feet",xlab="Year",main="Time series pl
```



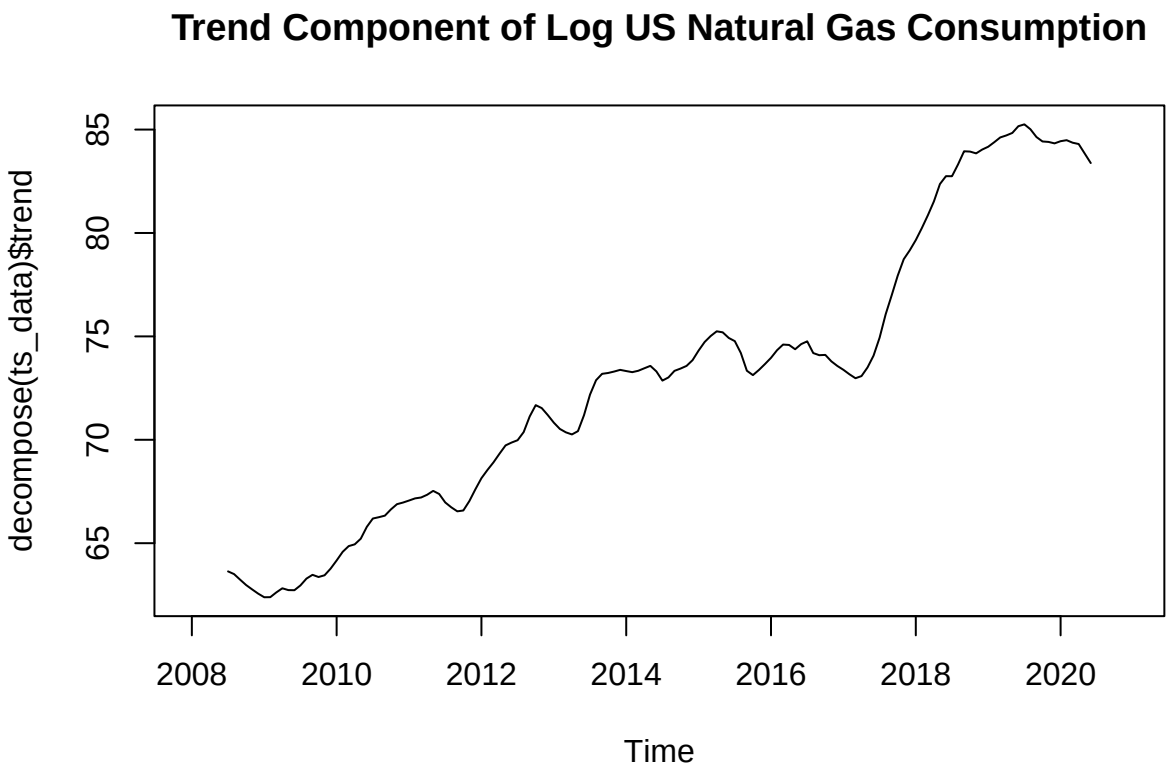

```r
ts_data = ts(ns_2$V2,start=c(2008,1), frequency = 12)
hist(ts_data, main="Histogram of US Natural Gas Consumption") #skewness
```



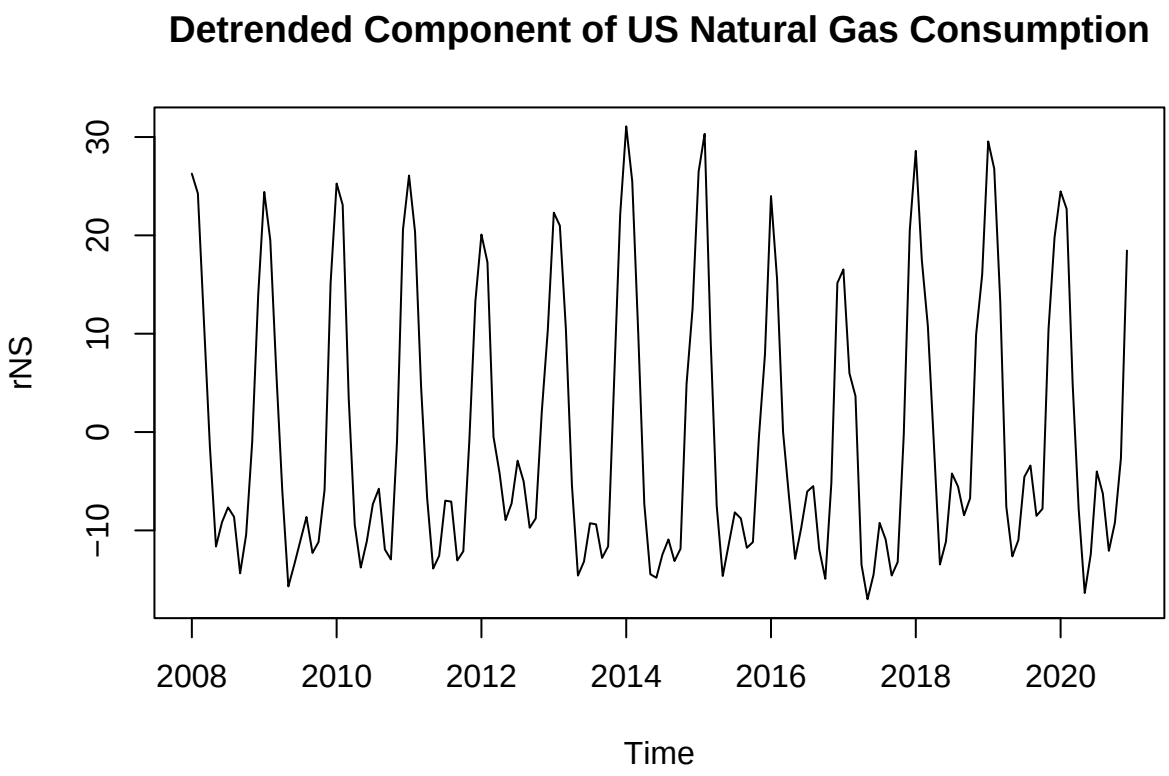

```
lx = log(ts_data) #Log to eliminate SKEWNESS
hist(lx, main="Hist. Log of US Natural Gas Consumption")
```



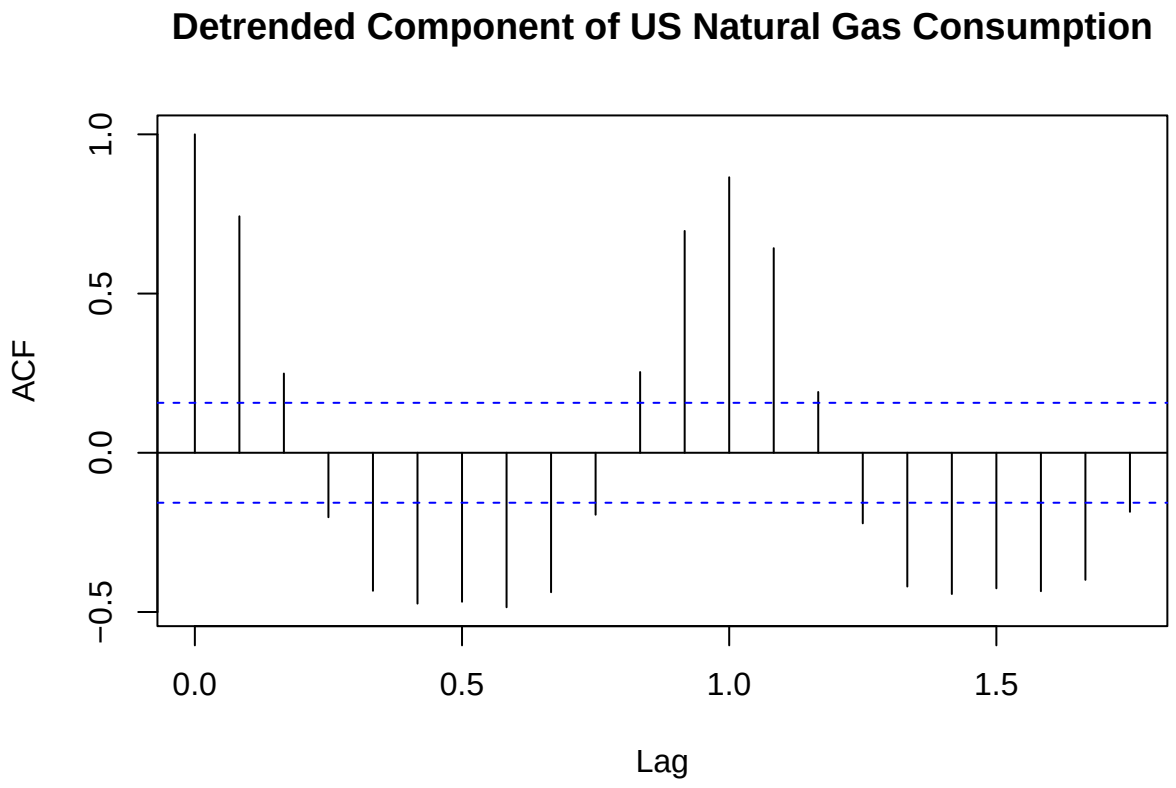

```
plot(decompose(ts_data)$trend, main="Trend Component of Log US Natural Gas Consumption")# Plot the tren
```




```r
# Since there is an obvious upward trend, we need to take that trend out
m1 <- lm(ts_data ~ time(ts_data))
#summary(m1)
rNS = ts(resid(m1),start=c(2008,1),freq=12)
plot(rNS, main= "Detrended Component of US Natural Gas Consumption")
```




```
acf(rNS, main= "Detrended Component of US Natural Gas Consumption")
```



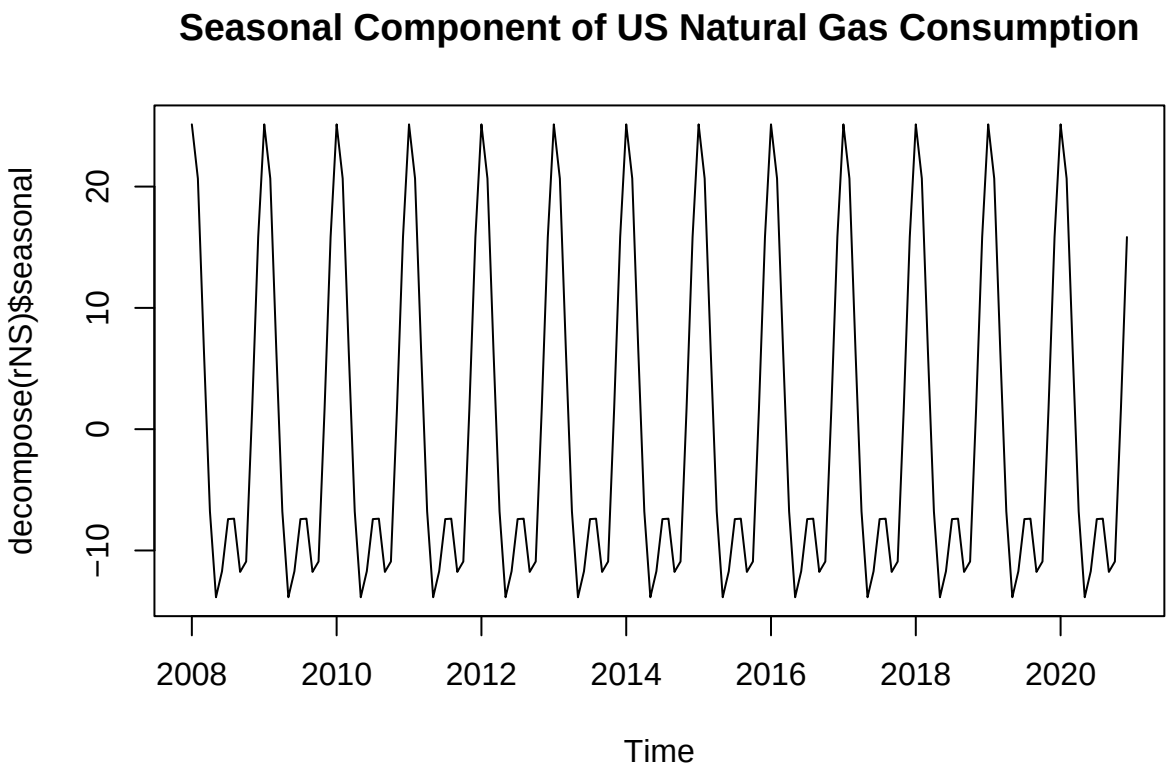

```
library(TSA)
```

```
##
## Attaching package: 'TSA'
```

```
## The following objects are masked from 'package:stats':
##
##     acf, arima
```

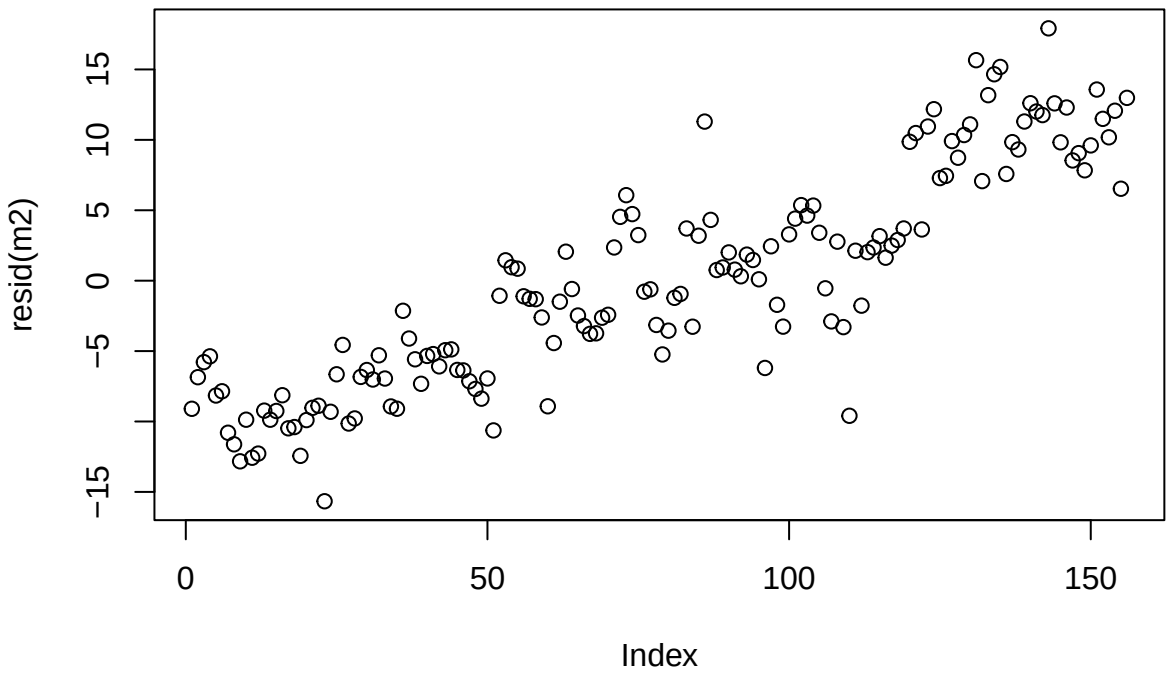
```
## The following object is masked from 'package:utils':
##
##     tar
```

```
# Plot the seasonal component
plot(decompose(rNS)$seasonal, main="Seasonal Component of US Natural Gas Consumption")
```

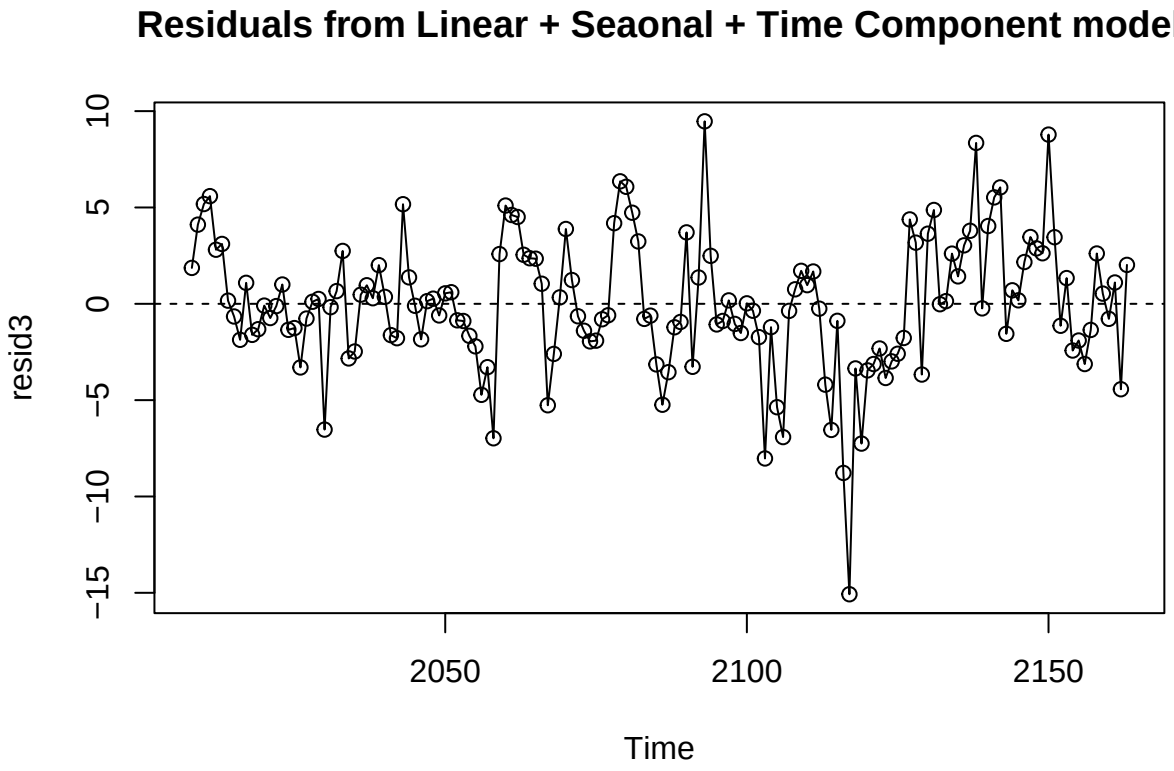



```
month=season(decompose(rNS)$seasonal)
```

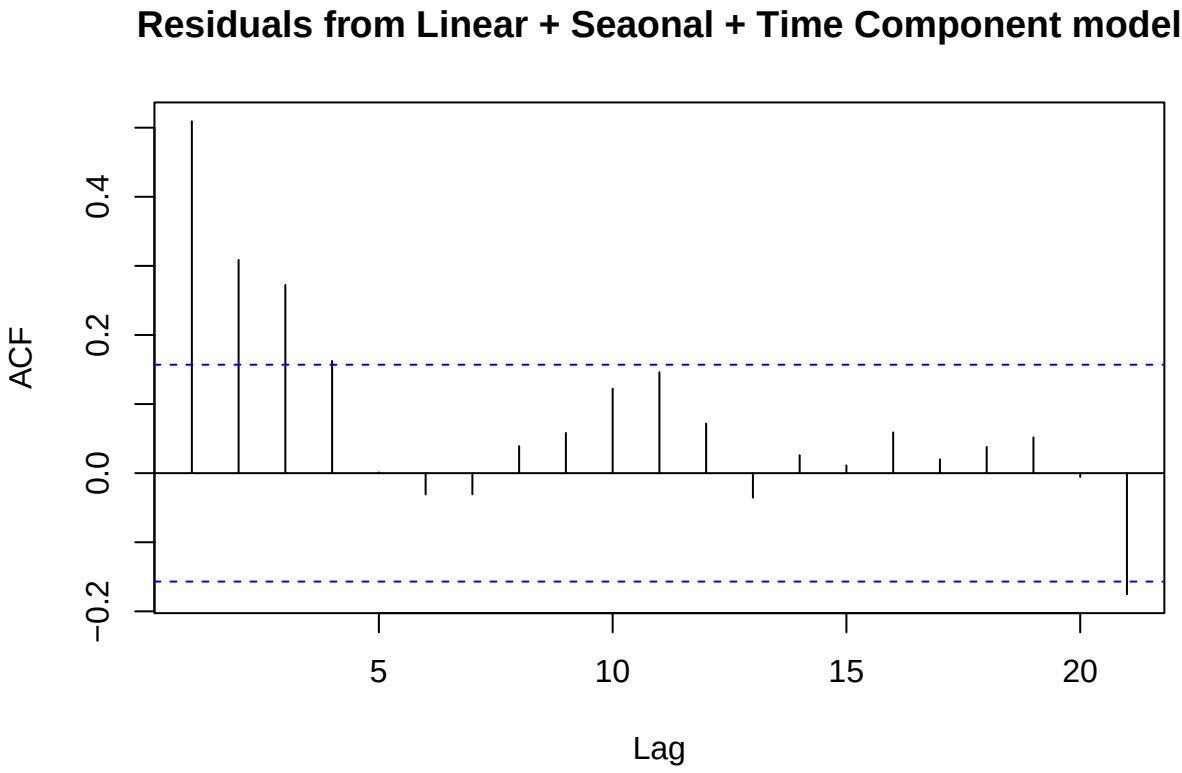
```
#plot Residuals from Linear Regression + Seaonal Component model
m2=lm(ts_data~season(ts_data))
plot(resid(m2), main="Residuals from Linear Regression + Seaonal Component model")
```



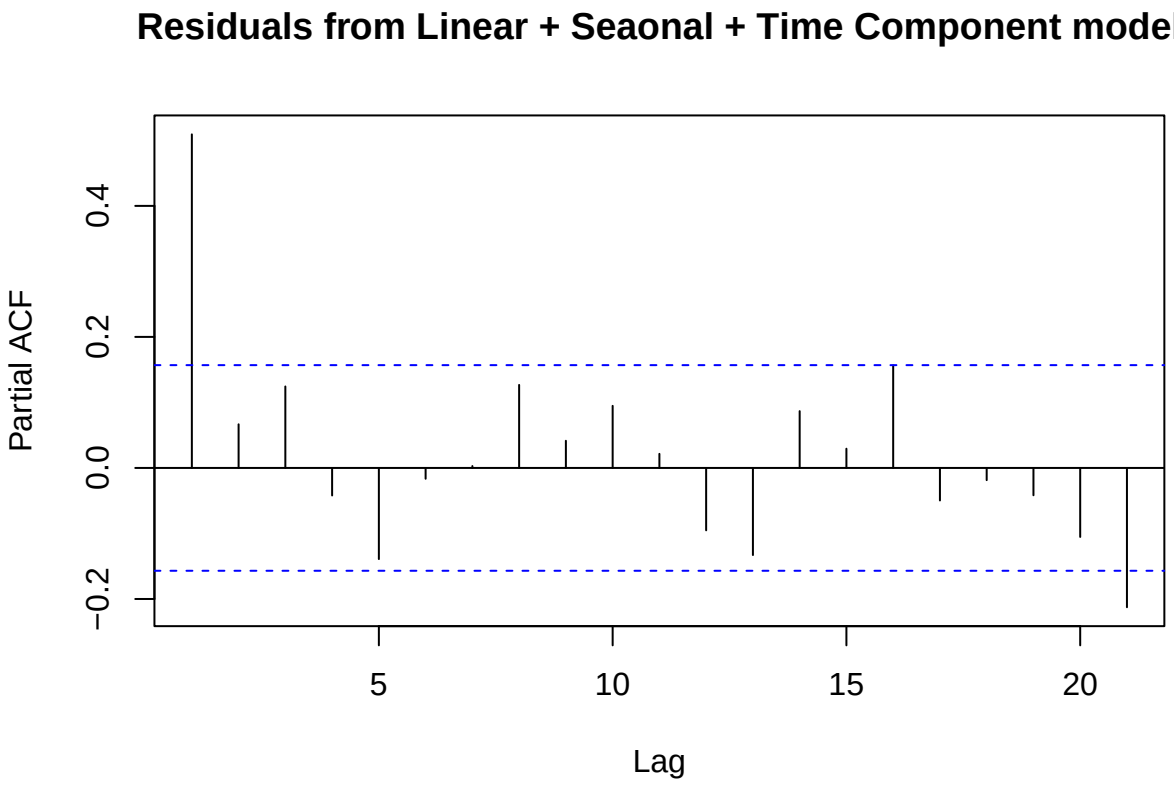

```r
library(TSA)
m3 = lm(ts_data~season(ts_data)+time(ts_data))
#summary(m3)
resid3 = ts(resid(m3),start=c(2008,1),freq=1)
lx3=residuals(m3)
#plot Residuals from Linear + Seaonal + Time Component model
plot(resid3, main="Residuals from Linear + Seaonal + Time Component model", type="o")
abline(h=0,lty=2)
```




```
acf(resid3,main="Residuals from Linear + Seaonal + Time Component model")
```




```
pacf(resid3,main="Residuals from Linear + Seaonal + Time Component model")
```




```
eacf(resid3)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x o o o o o o o  o  o  o
## 1 o x o o o o o o o o o  o  o  o
## 2 x x o o o o o o o o o  o  o  o
## 3 x x o x o o o o o o o  o  o  o
## 4 x x o o o o o o o o o  o  o  o
## 5 o o x x o o o o o o o  o  o  o
## 6 x o x o o o o o o o o  o  o  o
## 7 o o x o o x x o o o o  o  o  o
```

```
####tm and tm squared
tm=time(lx)
tm2=time(lx)^2
#summary of m3
summary(m3)
```

```
##
## Call:
## lm(formula = ts_data ~ season(ts_data) + time(ts_data))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -15.0737  -1.7759  -0.0979   2.3474   9.4701
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -3.582e+03  1.560e+02 -22.957  < 2e-16 ***
## season(ts_data)February   -4.267e+00  1.420e+00  -3.005  0.00314 **
## season(ts_data)March      -1.832e+01  1.420e+00 -12.899  < 2e-16 ***
## season(ts_data)April      -3.157e+01  1.420e+00 -22.229  < 2e-16 ***
## season(ts_data)May        -3.889e+01  1.420e+00 -27.383  < 2e-16 ***
## season(ts_data)June       -3.672e+01  1.420e+00 -25.853  < 2e-16 ***
## season(ts_data)July       -3.228e+01  1.421e+00 -22.724  < 2e-16 ***
## season(ts_data)August     -3.244e+01  1.421e+00 -22.835  < 2e-16 ***
## season(ts_data)September -3.698e+01  1.421e+00 -26.021  < 2e-16 ***
## season(ts_data)October    -3.602e+01  1.421e+00 -25.340  < 2e-16 ***
## season(ts_data)November   -2.394e+01  1.422e+00 -16.839  < 2e-16 ***
## season(ts_data)December   -9.273e+00  1.422e+00  -6.522 1.12e-09 ***
## time(ts_data)              1.827e+00  7.747e-02  23.581  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.621 on 143 degrees of freedom
## Multiple R-squared:  0.9474, Adjusted R-squared:  0.943
## F-statistic: 214.6 on 12 and 143 DF,  p-value: < 2.2e-16
```

```
#Quadratic Model
model4=lm(lx~month+tm+tm2)
summary(model4)
```

```
##
## Call:
## lm(formula = lx ~ month + tm + tm2)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.179305 -0.025452 -0.001957  0.032160  0.097432
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.397e+03  1.243e+03   1.123   0.2632
## monthFebruary  -4.603e-02  1.890e-02  -2.435   0.0161 *
## monthMarch     -2.112e-01  1.890e-02 -11.170  < 2e-16 ***
## monthApril     -3.932e-01  1.891e-02 -20.800  < 2e-16 ***
## monthMay       -5.111e-01  1.891e-02 -27.030  < 2e-16 ***
## monthJune      -4.746e-01  1.891e-02 -25.101  < 2e-16 ***
## monthJuly      -4.063e-01  1.891e-02 -21.486  < 2e-16 ***
## monthAugust    -4.075e-01  1.891e-02 -21.545  < 2e-16 ***
## monthSeptember -4.795e-01  1.892e-02 -25.351  < 2e-16 ***
## monthOctober   -4.633e-01  1.892e-02 -24.490  < 2e-16 ***
## monthNovember  -2.884e-01  1.892e-02 -15.238  < 2e-16 ***
## monthDecember  -1.062e-01  1.893e-02  -5.612 1.01e-07 ***
## tm             -1.408e+00  1.234e+00  -1.140   0.2560
## tm2             3.558e-04  3.064e-04   1.161   0.2475
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.04819 on 142 degrees of freedom
## Multiple R-squared:  0.9481, Adjusted R-squared:  0.9433
## F-statistic: 199.4 on 13 and 142 DF,  p-value: < 2.2e-16
```

```
#Linear model
m4 = lm(lx~month + tm)
summary(m4)
```

```
##
## Call:
## lm(formula = lx ~ month + tm)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.181885 -0.023889 -0.002782  0.031791  0.095402
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -47.114168   2.079550 -22.656  < 2e-16 ***
## monthFebruary   -0.046060   0.018927  -2.434   0.0162 *
## monthMarch      -0.211211   0.018927 -11.159  < 2e-16 ***
## monthApril      -0.393294   0.018928 -20.778  < 2e-16 ***
## monthMay        -0.511121   0.018930 -27.001  < 2e-16 ***
## monthJune       -0.474700   0.018931 -25.075  < 2e-16 ***
## monthJuly       -0.406379   0.018934 -21.463  < 2e-16 ***
## monthAugust     -0.407552   0.018936 -21.523  < 2e-16 ***
## monthSeptember  -0.479603   0.018939 -25.324  < 2e-16 ***
## monthOctober    -0.463382   0.018942 -24.463  < 2e-16 ***
## monthNovember   -0.288382   0.018946 -15.221  < 2e-16 ***
## monthDecember   -0.106226   0.018950  -5.606 1.04e-07 ***
## tm               0.025665   0.001033  24.856  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04825 on 143 degrees of freedom
## Multiple R-squared:  0.9476, Adjusted R-squared:  0.9432
## F-statistic: 215.4 on 12 and 143 DF,  p-value: < 2.2e-16
```

```
# Fit the models
model4 <- lm(lx~ month + time(lx) + I(time(lx)^2))
m4 <- lm(lx~ month + tm)

# Calculate the BIC values for each model (including the Linear + Seaonal + Time Component model)
bic_Quadratic <- BIC(model4)
bic_Linear <- BIC(m4)
bic_l.s.t <- BIC(m3)
bic_Quadratic
```

```
## [1] -442.3542
```

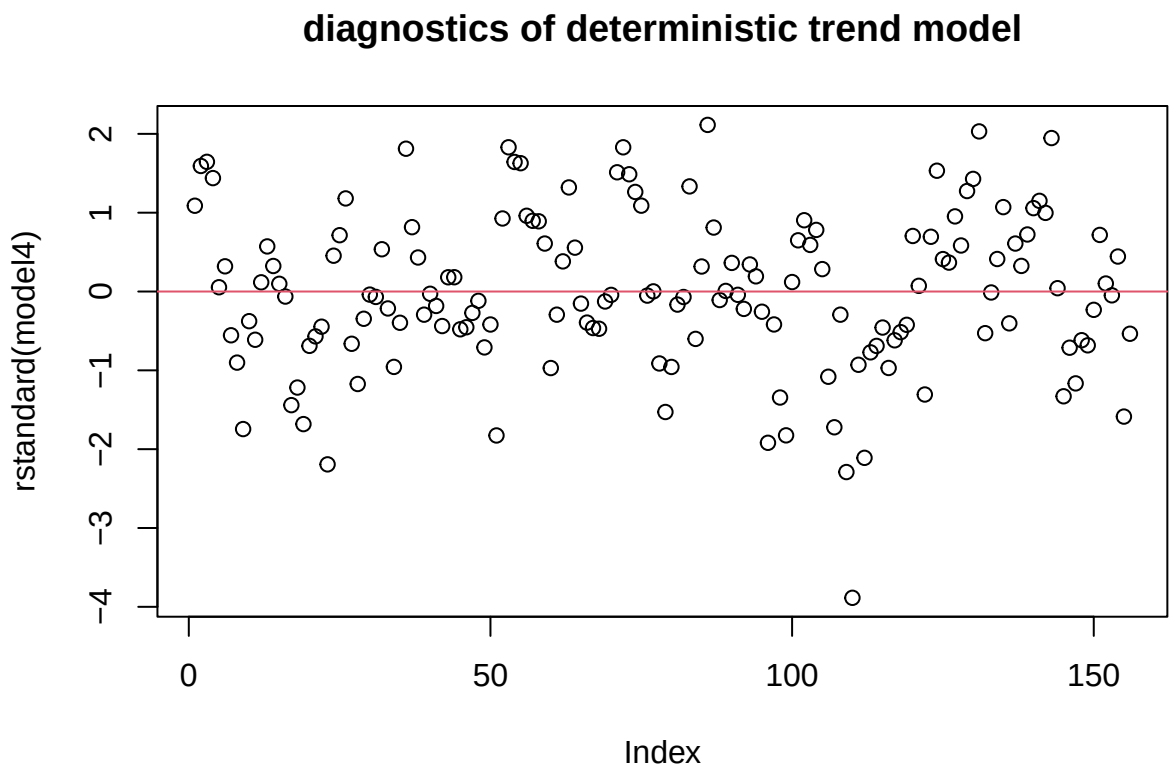
```
bic_Linear
```

```
## [1] -445.9296
```

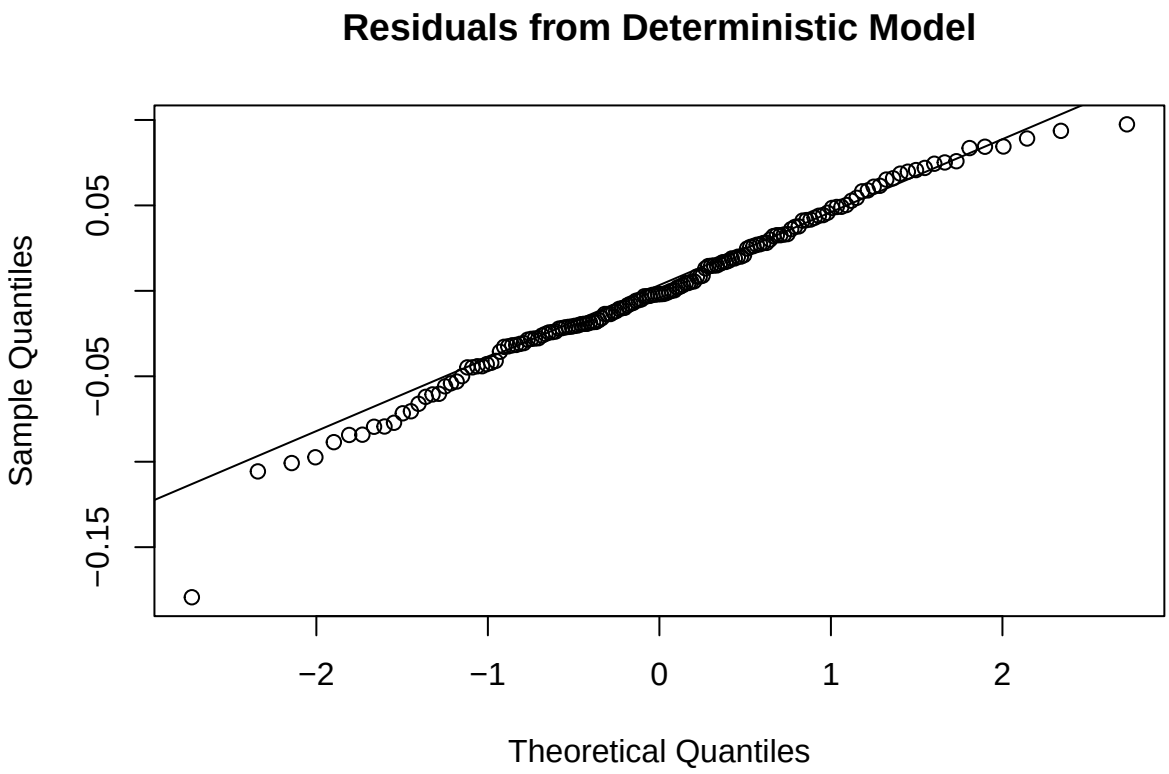
```
bic_l.s.t
```

```
## [1] 901.2647
```

```
#Based on BIC continue with the quadratic model
f.lx4=ts(fitted(model4),start=c(2008,1),freq=12)
#lines(f.lx4,col=2,lty=2)
lx4=residuals(model4)
# Check the diagnostics of this deterministic trend model
# 1) Residual plot (zero mean and homoscedasticity)
plot(rstandard(model4), main="diagnostics of deterministic trend model")
abline(h=0,col=2)
```




```
# 2) QQ plot (normality)
qqnorm(lx4, main= "Residuals from Deterministic Model")
qqline(lx4)
```




```
# 3) Shapiro-Wilk test (normality) and runs test (independence)
shapiro.test(lx4)
```

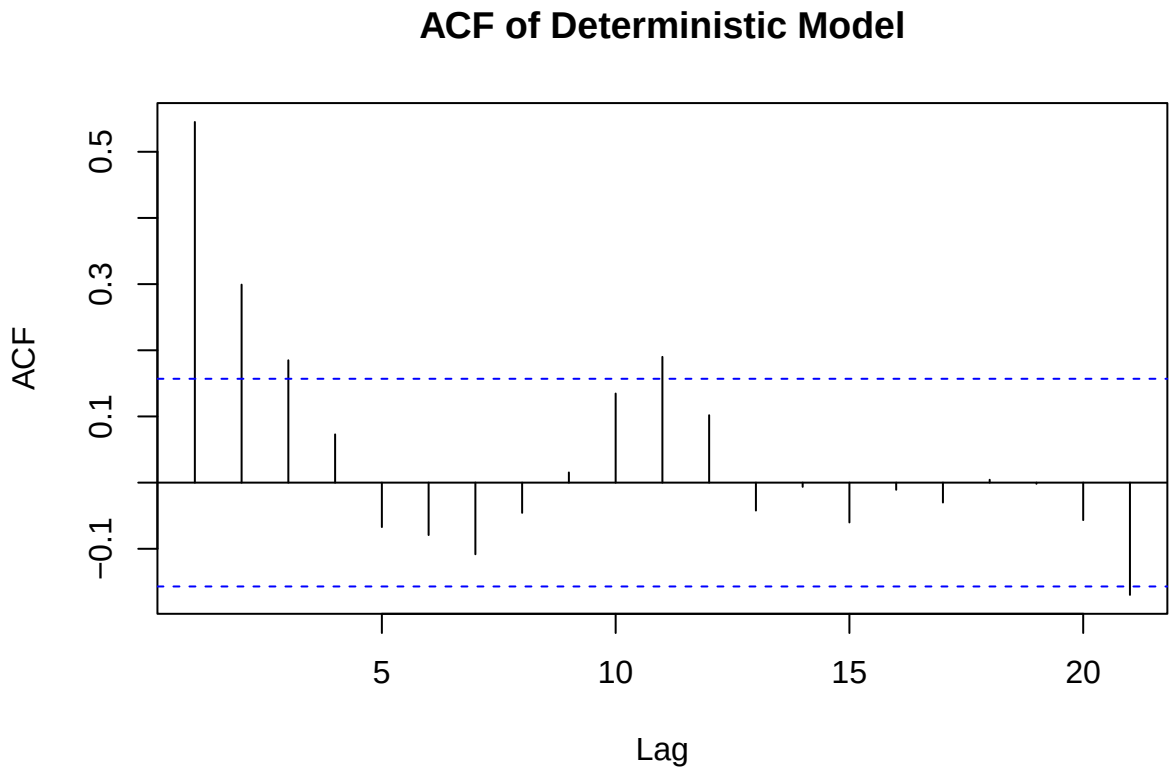
```
##
##  Shapiro-Wilk normality test
##
## data:  lx4
## W = 0.98448, p-value = 0.078
```

```
runs(lx4)
```

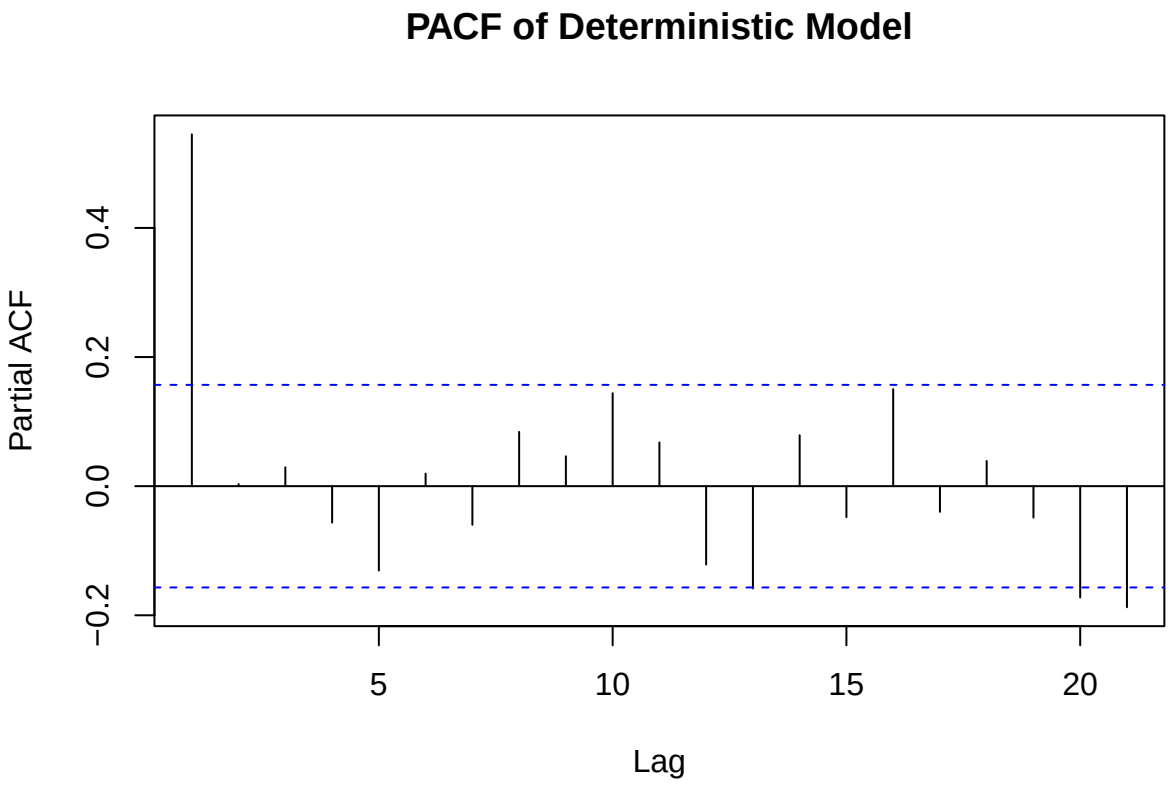
```
## $pvalue
## [1] 2.59e-09
##
## $observed.runs
## [1] 42
##
## $expected.runs
## [1] 78.79487
##
## $n1
## [1] 82
##
## $n2
## [1] 74
```

```
##
## $k
## [1] 0
```

```
# 4) ACF plot (independence)
acf(lx4,main= "ACF of Deterministic Model")
```




```
pacf(lx4, main= "PACF of Deterministic Model")
```




```
# d
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
adf.test(lx4)
```

```
## Warning in adf.test(lx4): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  lx4
## Dickey-Fuller = -4.7375, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```
pp.test(lx4)
```

```
## Warning in pp.test(lx4): p-value smaller than printed p-value
```

```
##
```

```
##  Phillips-Perron Unit Root Test
##
## data:  lx4
## Dickey-Fuller Z(alpha) = -71.102, Truncation lag parameter = 4, p-value
## = 0.01
## alternative hypothesis: stationary
```

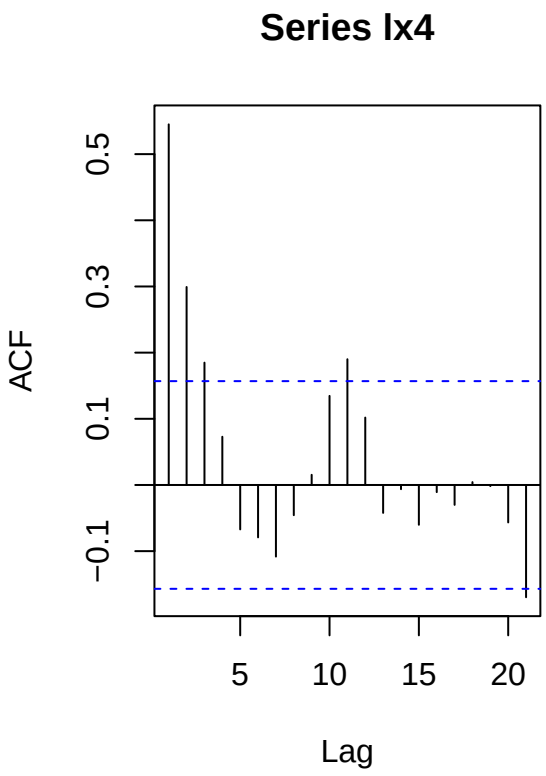
```
kpss.test(lx4)
```

```
## Warning in kpss.test(lx4): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  lx4
## KPSS Level = 0.069704, Truncation lag parameter = 4, p-value = 0.1
```

```
# P and q
par(mfrow=c(1,2))
acf(lx4)    # MA(3) pacf(1x4)   # AR(1) par(mfrow=c(1,1))
eacf(lx4)   # MA(3), AR(1), ARMA(2,2), ARMA(1,2)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x o o o o o o o x  o  o  o
## 1 o o o o o o o o o o o  o  o  o
## 2 x o o o o o o o o o o  o  o  o
## 3 x x o x o o o o o o o  o  o  o
## 4 x x o x o o o o o o o  o  o  o
## 5 o x x x o o o o o o o  o  o  o
## 6 x o o o o o o o o o o  o  o  o
## 7 x o x o o x o o o o o  o  o  o
```




```
library(forecast)
```

```
## Registered S3 methods overwritten by 'forecast':
##   method       from
##   fitted.Arima TSA
##   plot.Arima   TSA
```

```
auto.arima(lx4) # AR(1)
```

```
## Series: lx4
## ARIMA(1,0,0) with zero mean
##
## Coefficients:
##          ar1
##       0.5465
## s.e.  0.0669
##
## sigma^2 = 0.00149:  log likelihood = 286.66
## AIC=-569.33   AICc=-569.25   BIC=-563.23
```

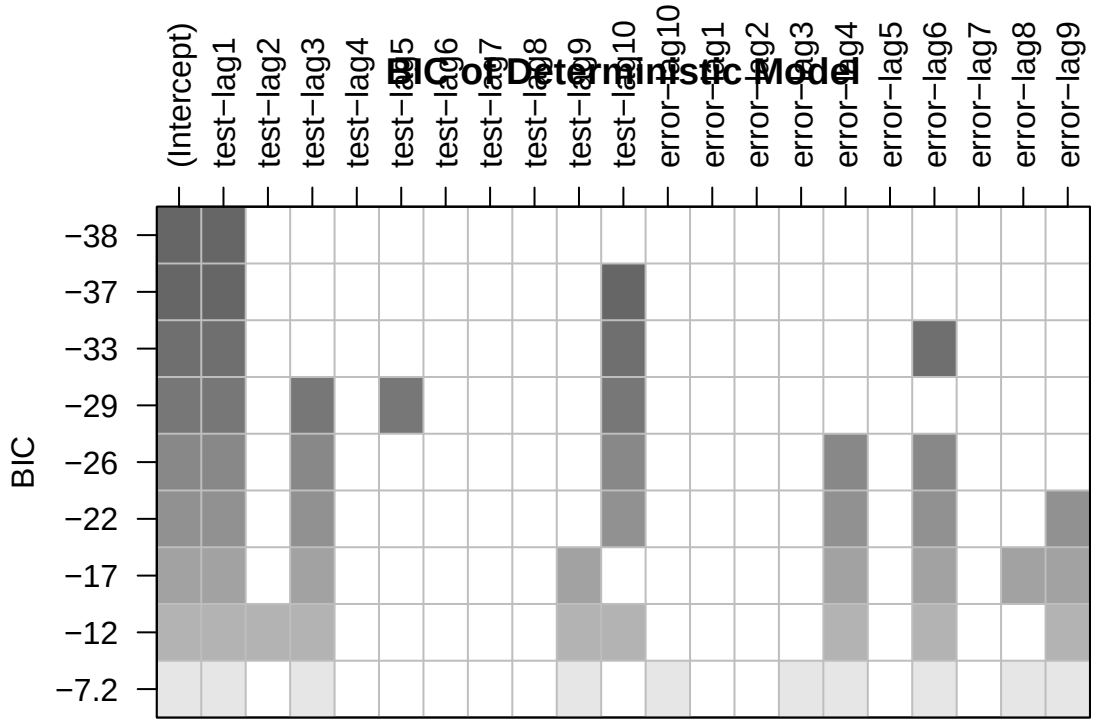
```
# Candidate models could be MA(3), AR(1), or ARMA(2,2)
ma3=Arima(lx4,order=c(0,0,3),include.mean=F) #
ar1=Arima(lx4,order=c(1,0,0),include.mean=F)
arma22=Arima(lx4,order=c(2,0,2),include.mean=F) #
```

```
library(TSA)
res=armasubsets(y=lx4,nar=10,nma=10,y.name='test',ar.method='ols')
```

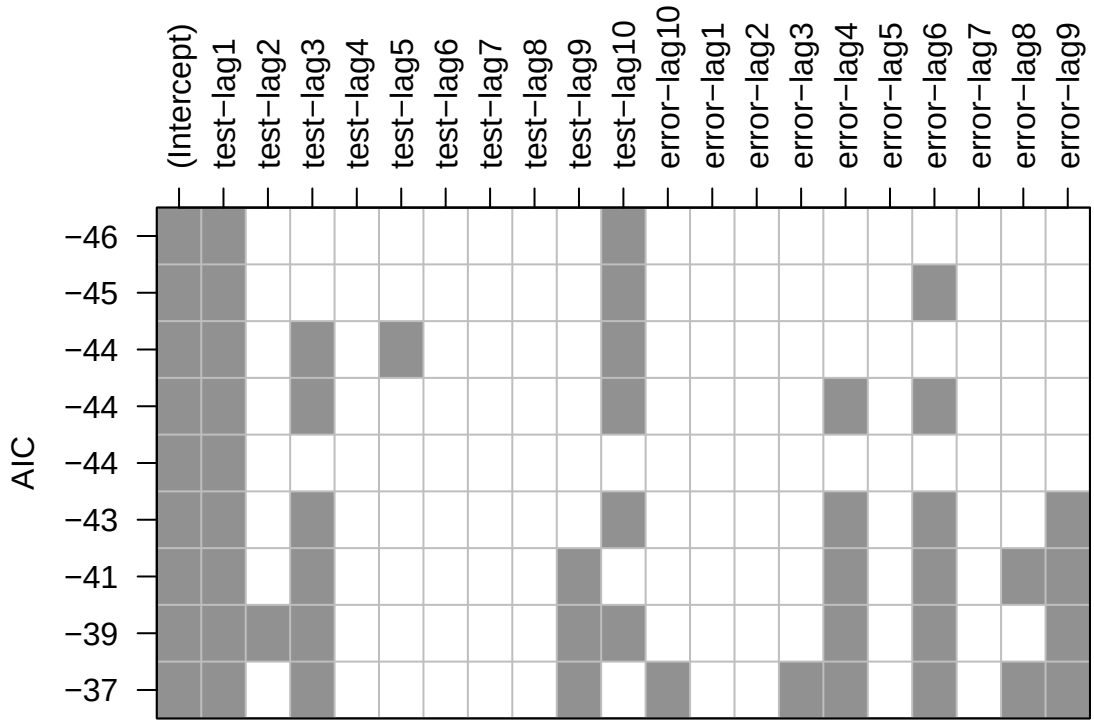
```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =
## force.in, : 9 linear dependencies found
```

```
## Reordering variables and trying again:
```

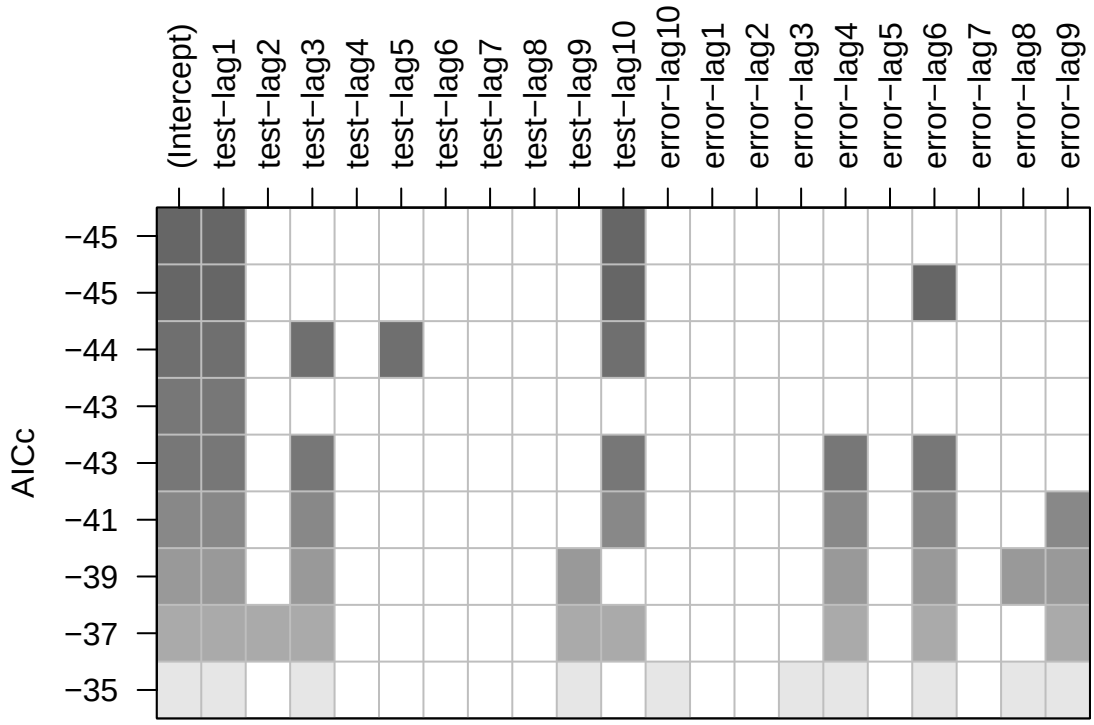
```
plot(res, main="BIC of Deterministic Model") # default is BIC
```



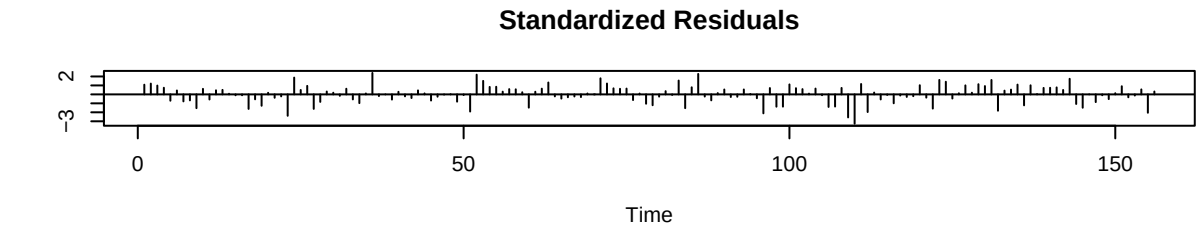

```
# AR(1) model is chosen based on BIC
plot(res,scale='AIC')
```

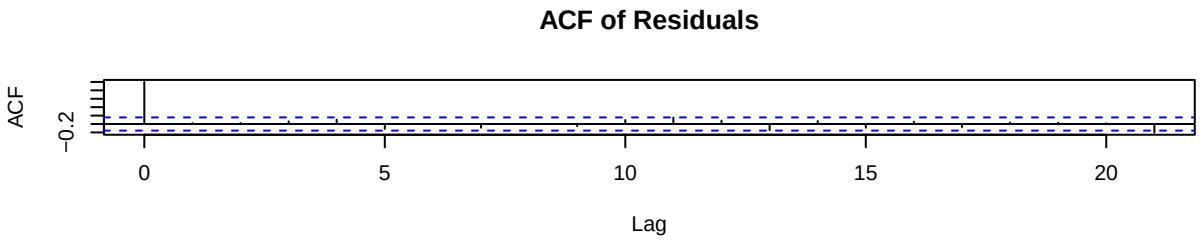


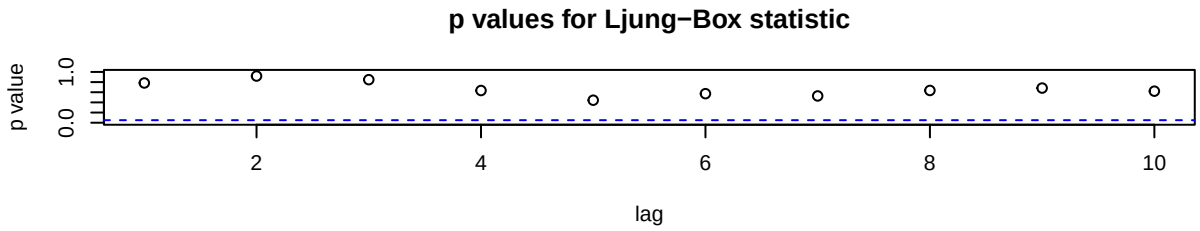

```
plot(res,scale='AICc') # more complicated models are suggested from AIC and AICc criterions.
```



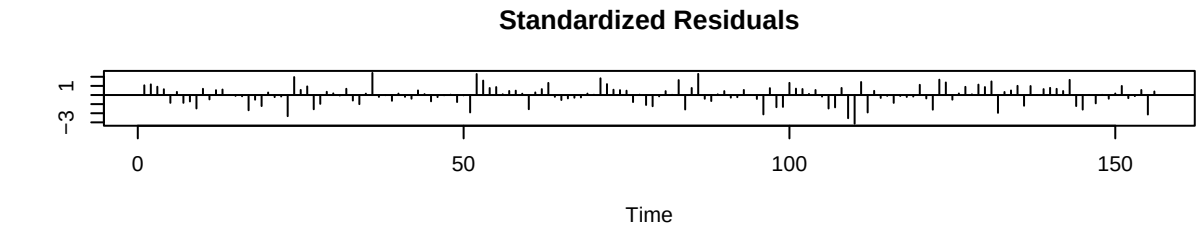

```
# Model diagnostics
library(TSA)
tsdiag(ma3)
```

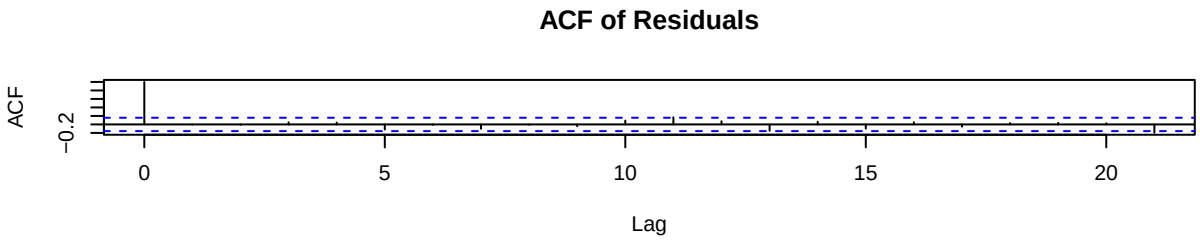

Standardized Residuals
2
−3
0
50
100
150
Time

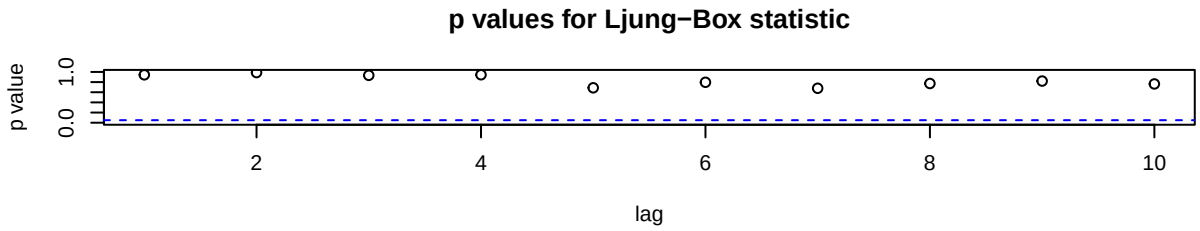

ACF of Residuals
ACF
1.0
−0.2
0
5
10
15
20
Lag


p values for Ljung−Box statistic
p value
1.0
0.0
2
4
6
8
10
lag

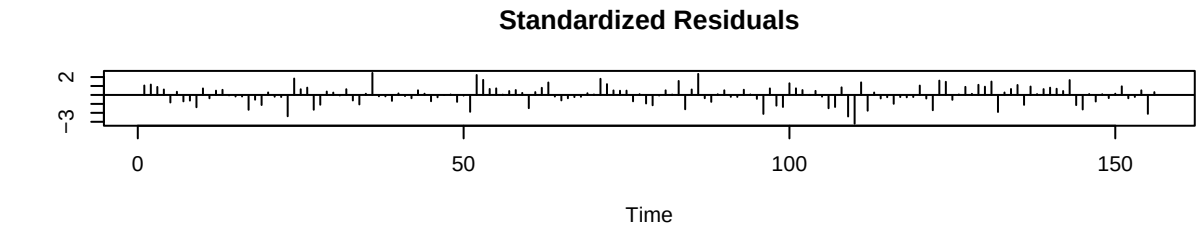
```
tsdiag(ar1)
```

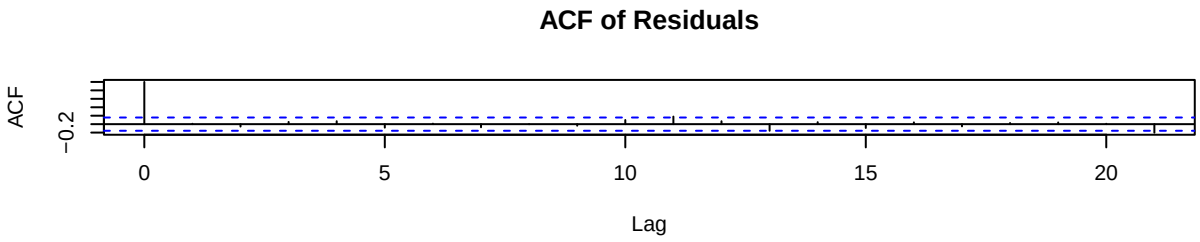


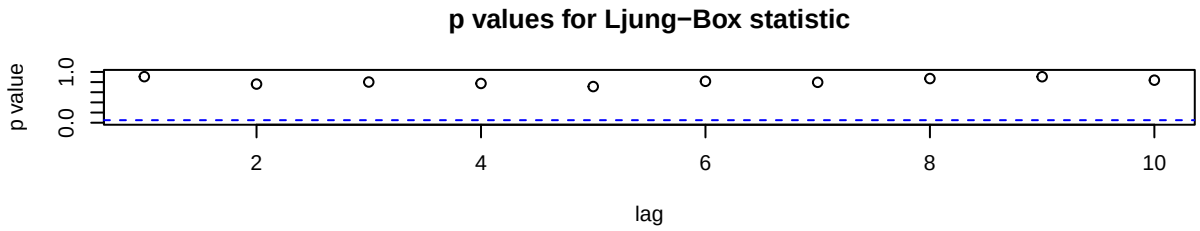




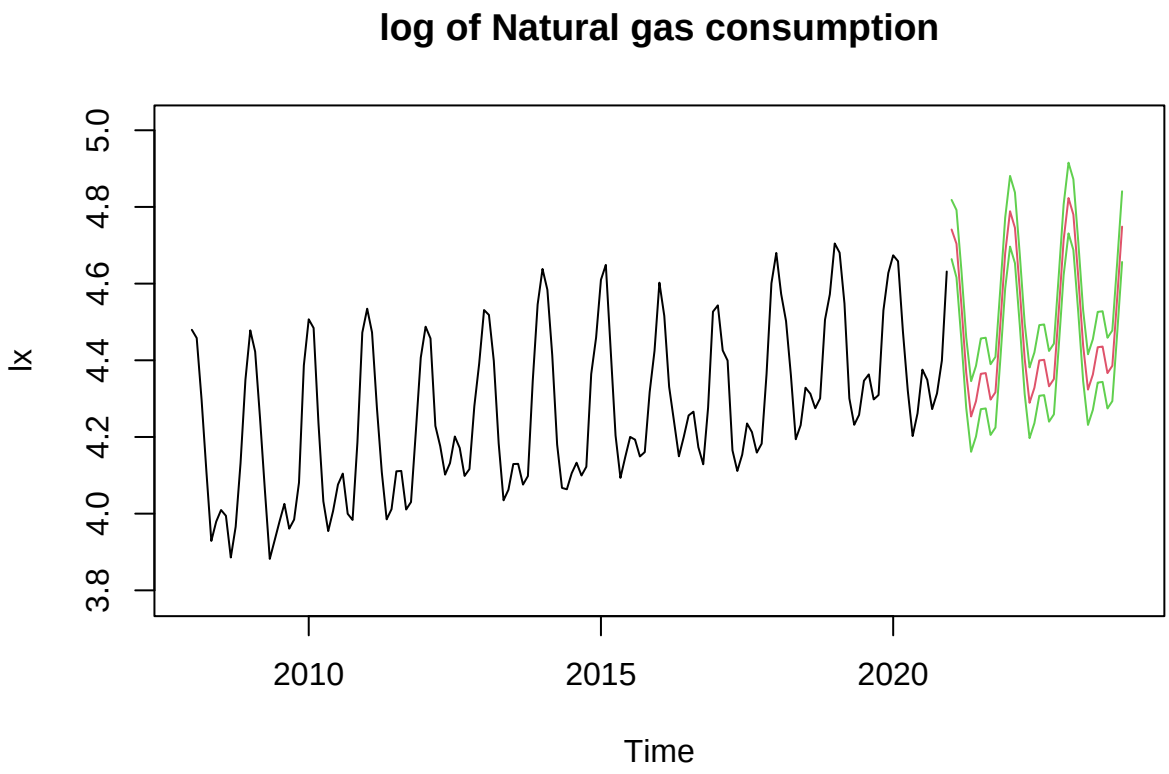

```
tsdiag(arma22)
```



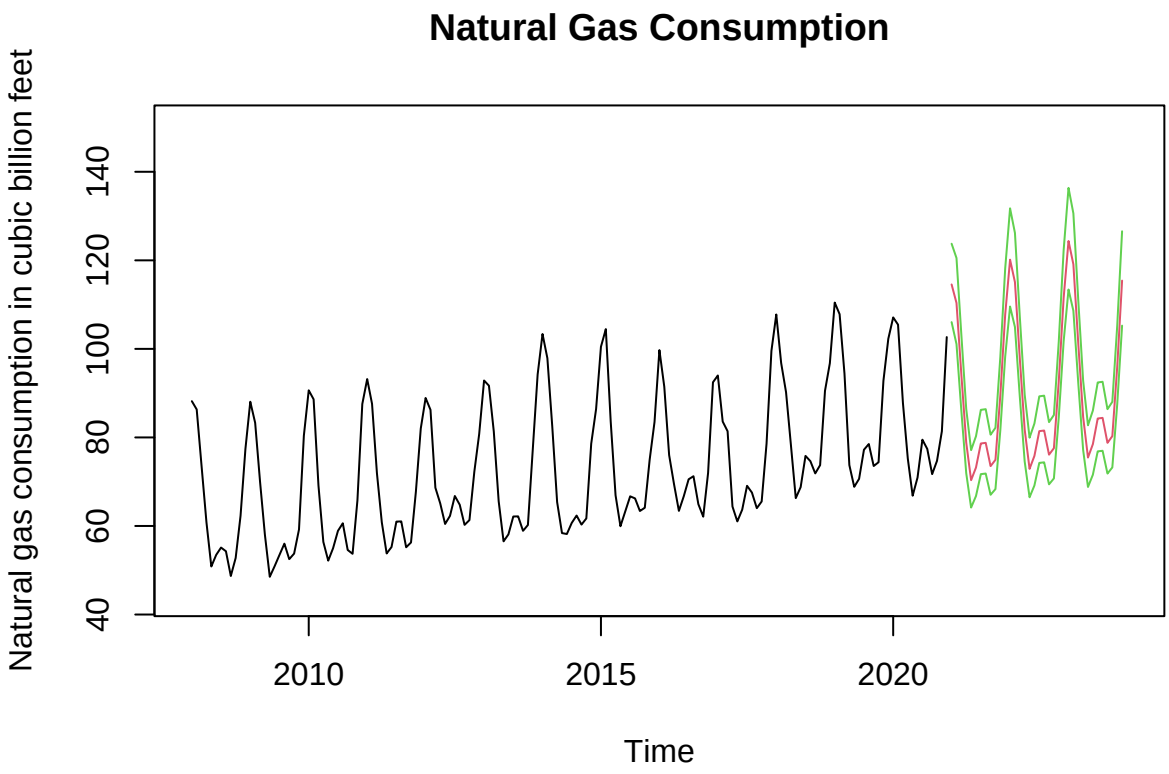





```r
##### Let's try forecasting for the next three years ##### based on the AR(1) model
newtm=seq(from=2021,to=2024.917,length=36)
newdata=data.frame(month=as.factor(month[1:36]),tm=newtm,tm2=newtm^2)
predxreg=predict(m4,newdata)  ## prediction of the deterministic trend portion
##### based on the AR(1) model
predx=predict(ar1,n.ahead=36) ## prediction of the stationary error portion
pr=predx$pred+predxreg
uci=pr+2*predx$se
lci=pr-2*predx$se
# To plot the predicted values as prediction intervals, code them as time series
pr=ts(pr,start=2021,freq=12)
uci=ts(uci,start=2021,freq=12)
lci=ts(lci,start=2021,freq=12)

ymin=min(c(as.vector(lci),lx))-.1
ymax=max(c(as.vector(uci),lx))+.1

plot(lx,xlim=c(2008,2024),ylim=c(ymin,ymax),main="log of Natural gas consumption")
lines(pr,col=2)
lines(uci,col=3)
lines(lci,col=3)
```




```
# In the original scale..
plot(exp(lx),ylab="Natural gas consumption in cubic billion feet",main="Natural Gas Consumption",xlim=c
lines(exp(pr),col=2)
lines(exp(uci),col=3)
lines(exp(lci),col=3)
```


Natural Gas Consumption
Natural gas consumption in cubic billion feet
40
60
80
100
120
140
2010
2015
2020
Time