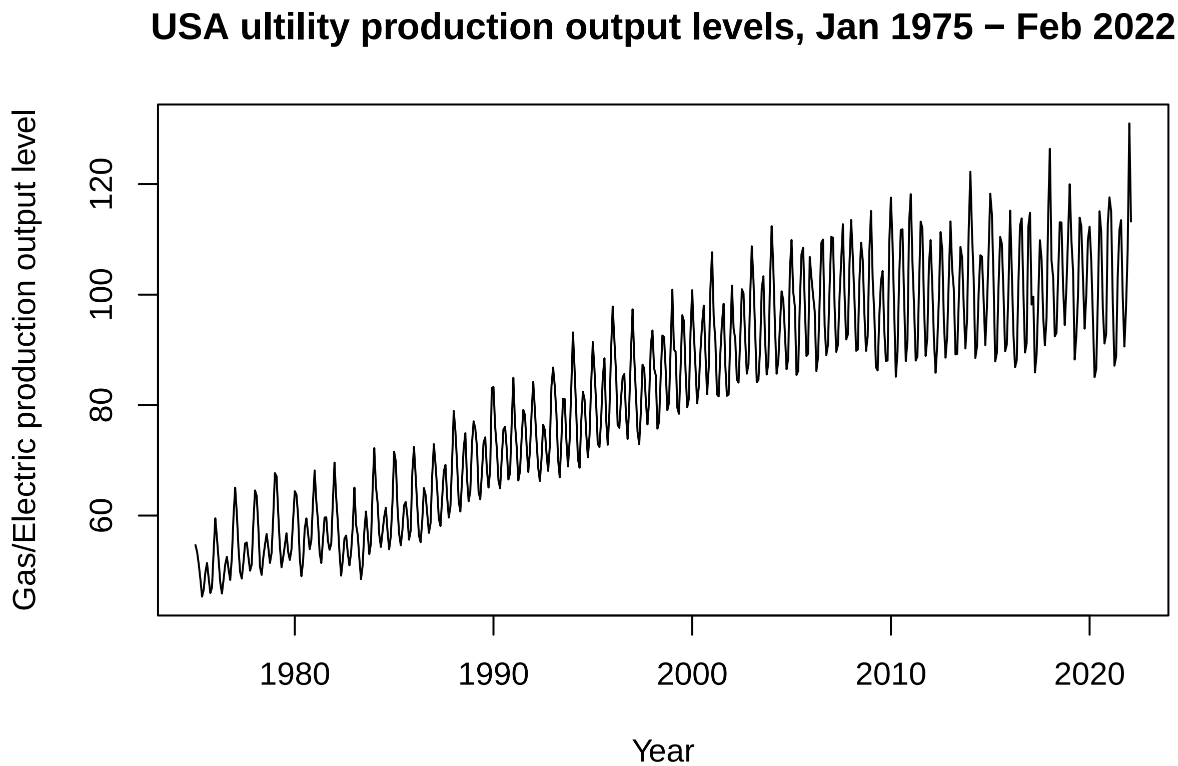
STAT 5550 project

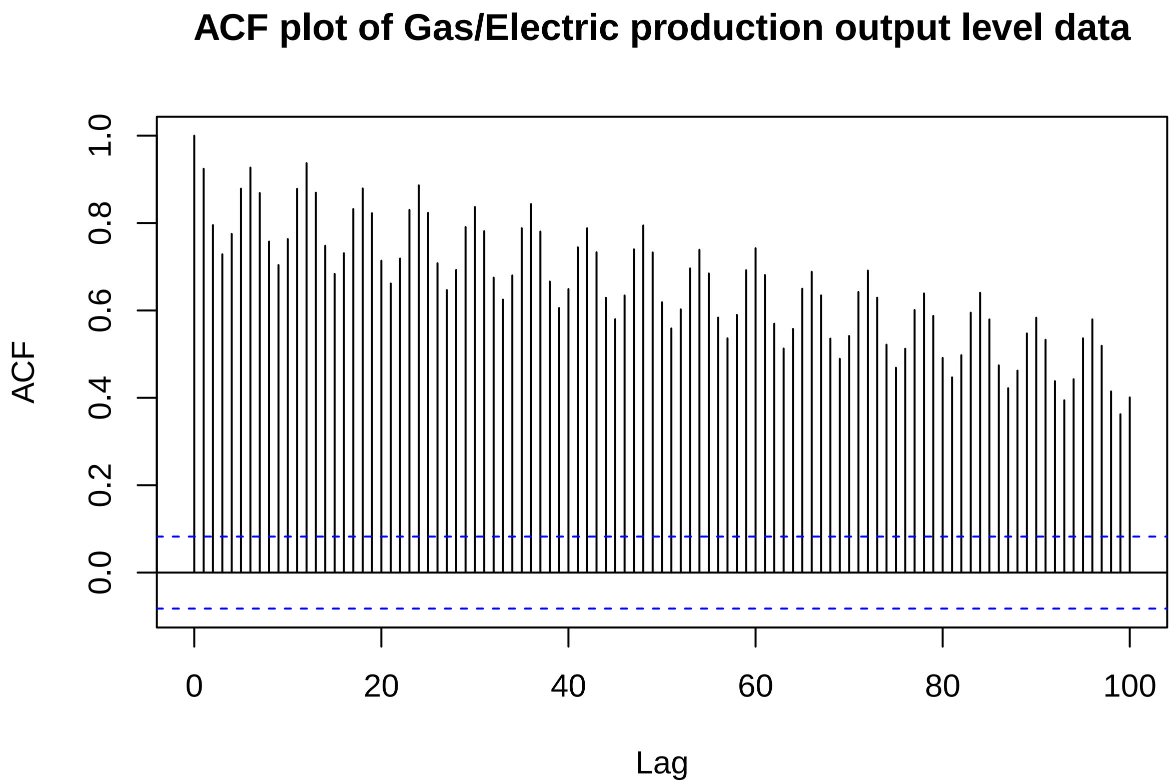
James Laughead

4/13/2022

```r
project.data <- read.csv("IPG2211A2N.csv")
df <- (project.data$IPG2211A2N)
df <- df[433:998]

df.ts = ts(df, frequency = 12, start=c(1975, 1))
plot.ts(df.ts, xlab = "Year", ylab = "Gas/Electric production output level", main = "USA ultility produc
```



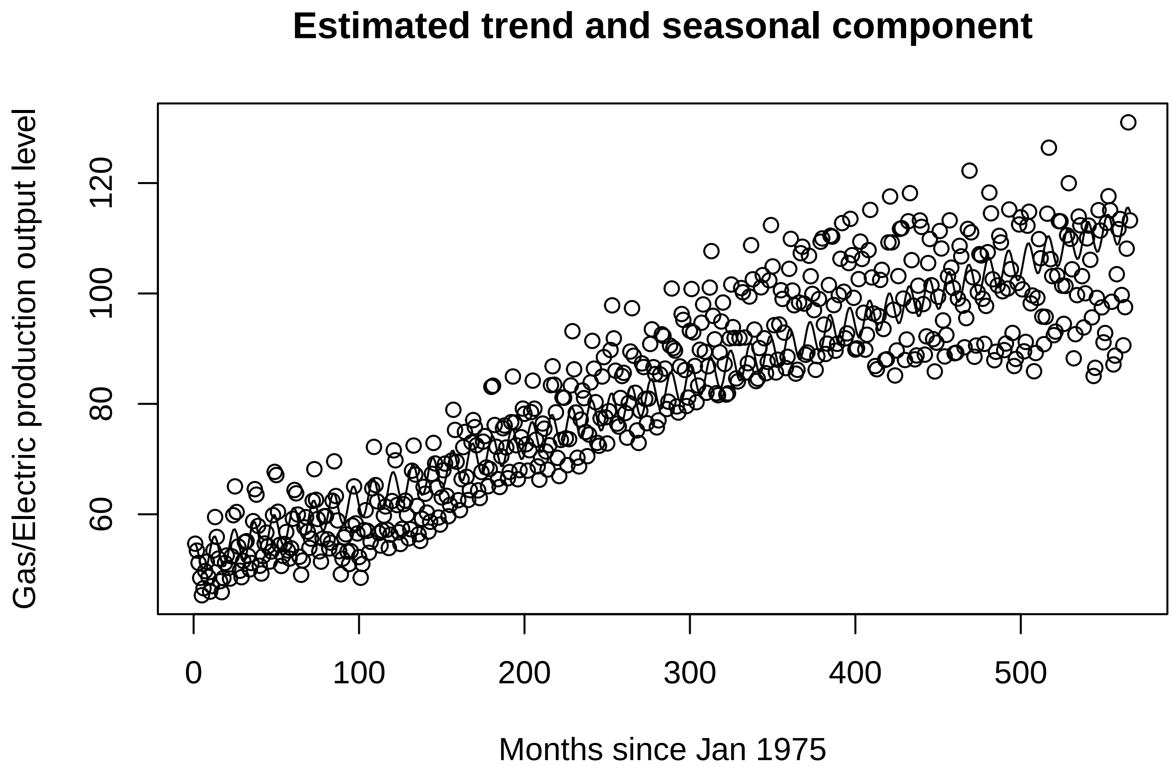

```r
acf(df, lag.max = 100, main = "ACF plot of Gas/Electric production output level data")
```



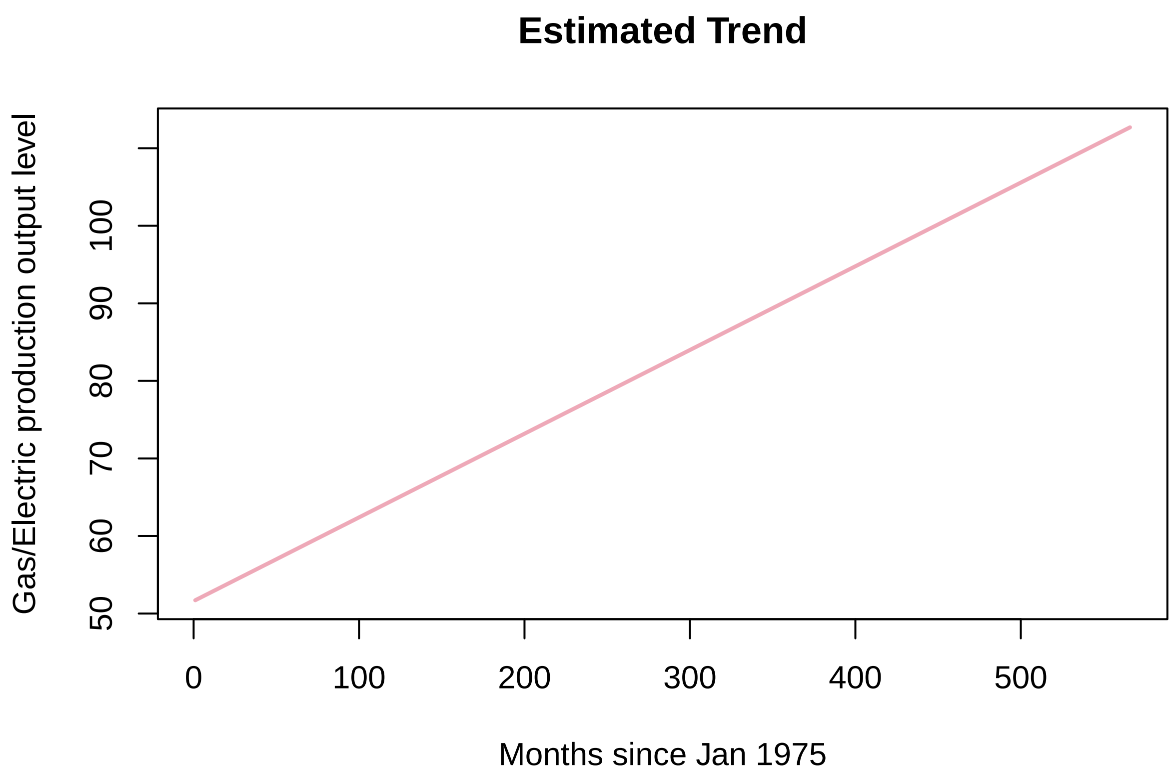

```
t = 1:length(df)
c = cos(2*pi*t/12)
s = sin(2*pi*t/12)
mod3 = lm(df.ts ~ t+c+s)
summary(mod3)
```

```
##
## Call:
## lm(formula = df.ts ~ t + c + s)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.436  -5.351  -0.671   5.823  20.999
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 51.605520   0.694956  74.257  < 2e-16 ***
## t            0.107923   0.002124  50.814  < 2e-16 ***
## c            2.990434   0.490748   6.094 2.05e-09 ***
## s            0.770111   0.490748   1.569    0.117
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.255 on 562 degrees of freedom
## Multiple R-squared:  0.8238, Adjusted R-squared:  0.8228
## F-statistic: 875.6 on 3 and 562 DF,  p-value: < 2.2e-16
```

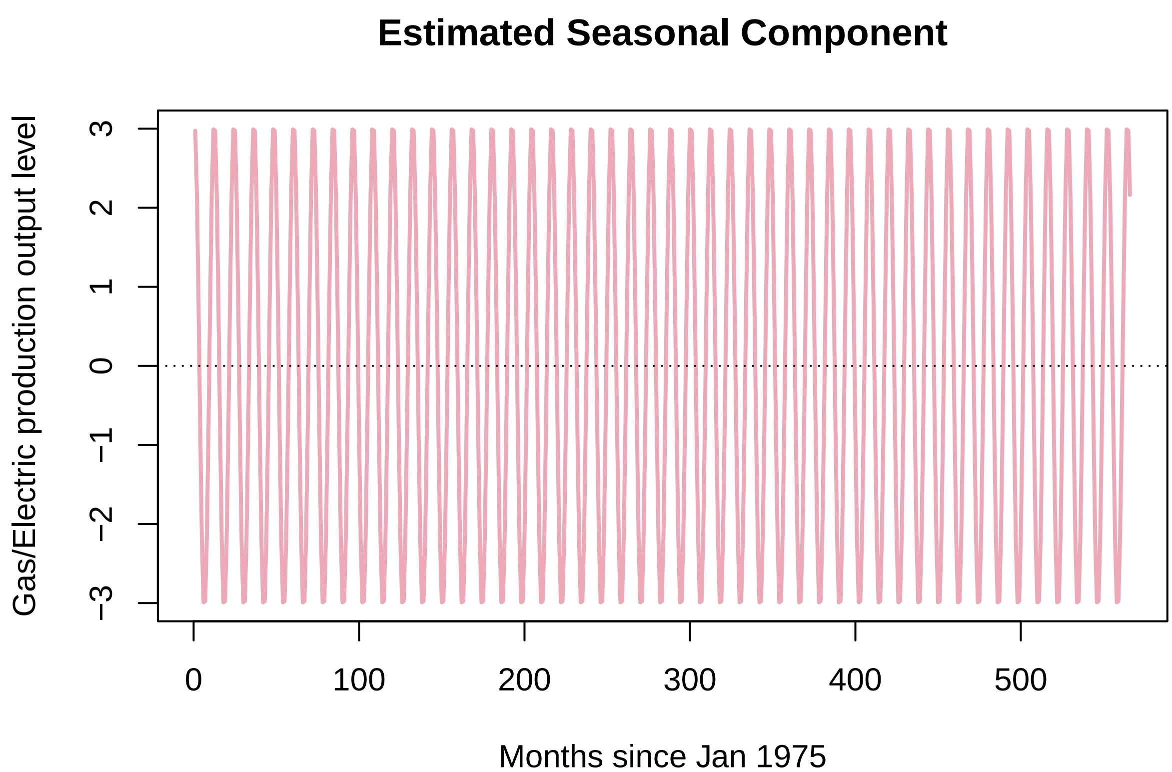
```
plot(t, df.ts, main = "Estimated trend and seasonal component", xlab = "Months since Jan 1975", ylab =
lines(mod3$fit)
```



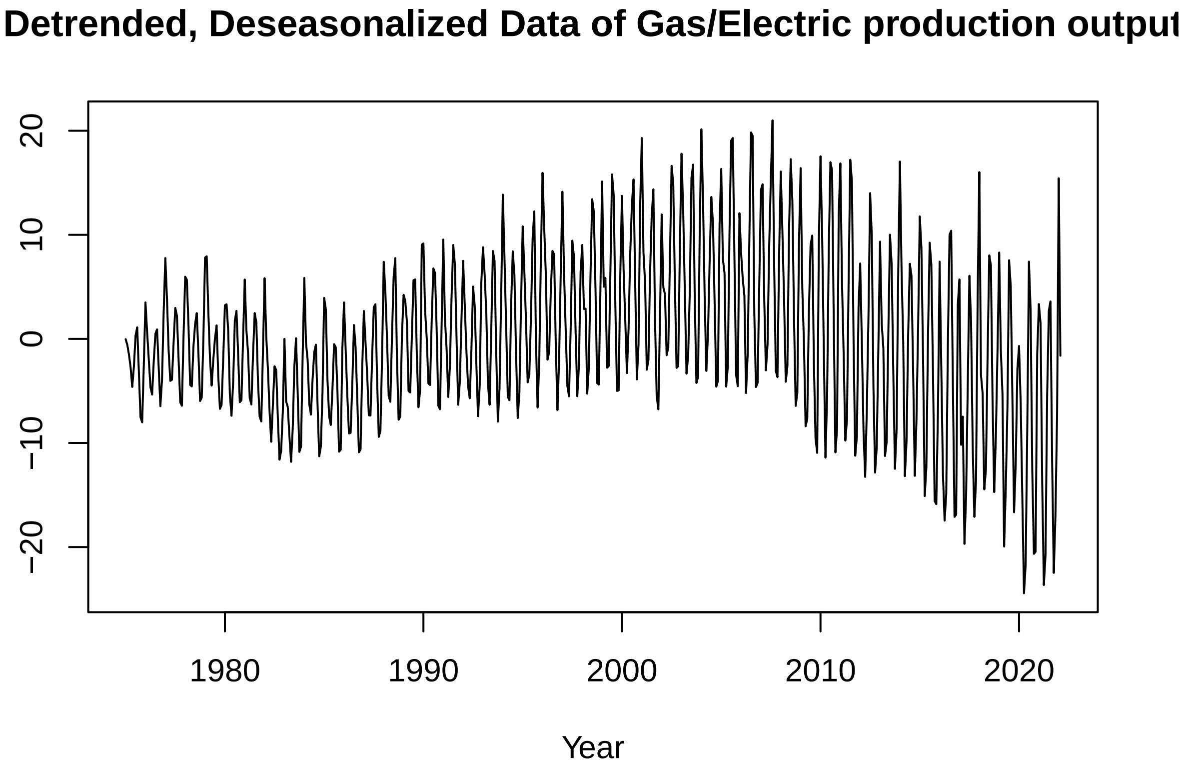

```
a <- summary(mod3)$coefficients
ut0 <- a[ 1,1 ]
ut1 <- a[ 2,1 ]
uc1 <- a[ 3,1 ]
uc2 <- a[ 4,1 ]



plot(time(df), ut0 + (ut1*t), type="l", col="pink2", lwd=2,
 main="Estimated Trend", xlab = "Months since Jan 1975", ylab = "Gas/Electric production output level" )
```



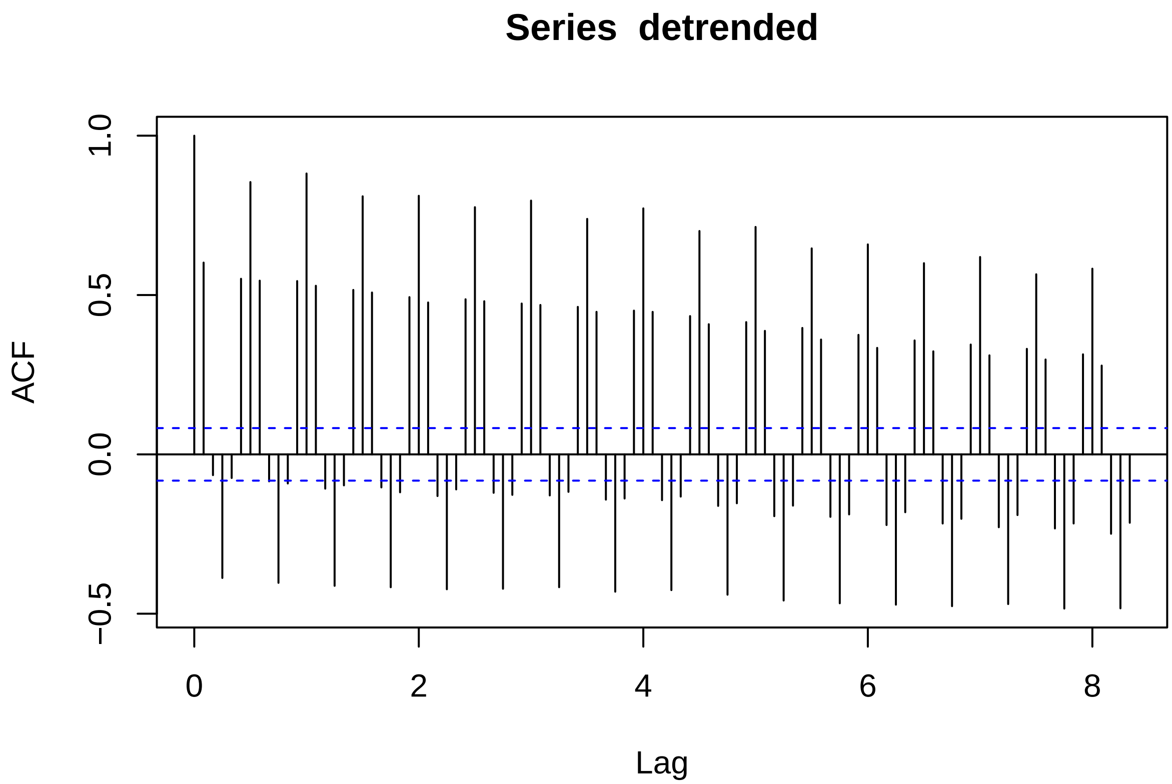

```
plot(time(df), (2.990434*(c) +  0.77011133*(s)), type="l", col="pink2", lwd=2,
 main="Estimated Seasonal Component", xlab = "Months since Jan 1975", ylab = "Gas/Electric production ou
abline(h=0, lty=3)
```



```
# Make a time series object with the detrended, deseasonalized data:
detrended = df - (ut0 + ut1*t) - ((2.9904347*(c))+((0.77011133)*(s)))
detrended = ts(detrended, frequency = 12, start=c(1975, 1))

plot.ts(detrended, ylab="", main="Detrended, Deseasonalized Data of Gas/Electric production output",xlal
```



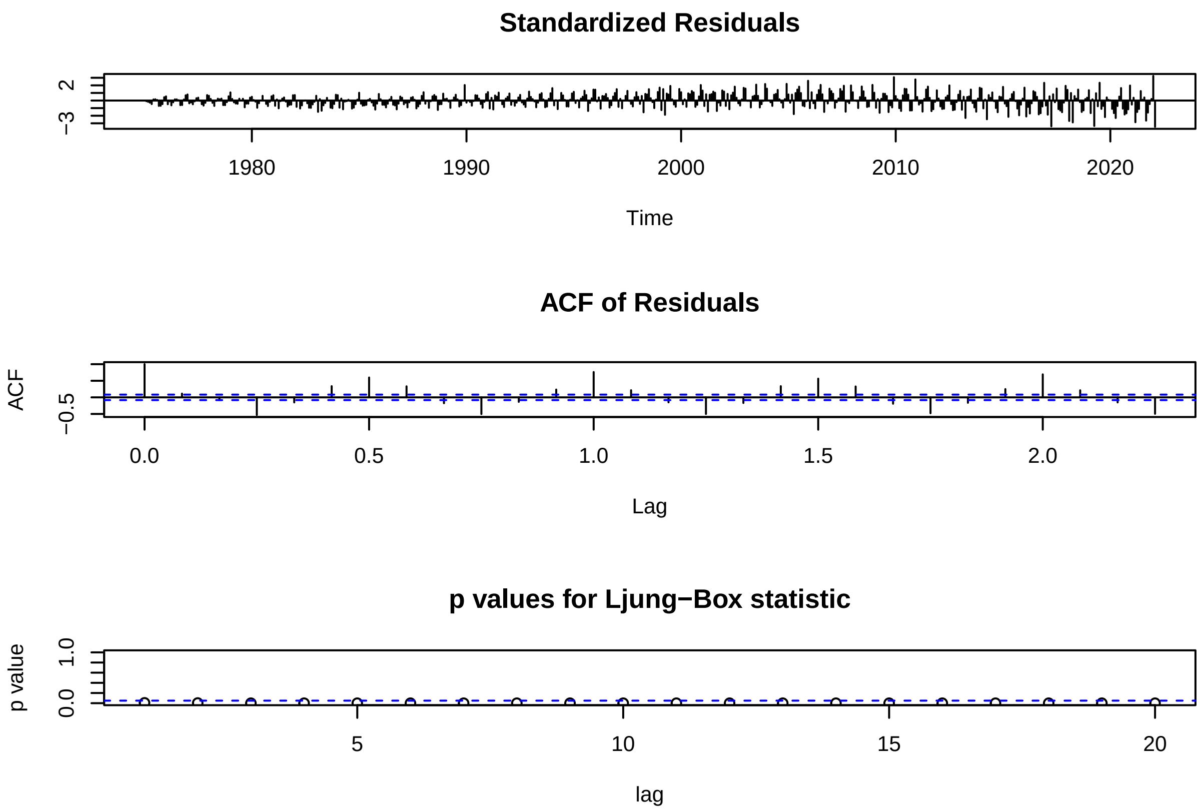

```
acf(detrended, lag.max = 100)
```




```r
varve.arma = arima(detrended, order = c(1, 0, 1))
varve.arma
```

```
##
## Call:
## arima(x = detrended, order = c(1, 0, 1))
##
## Coefficients:
##          ar1     ma1  intercept
##       0.4002  0.6304    -0.0377
## s.e.  0.0435  0.0303     0.6242
##
## sigma^2 estimated as 29.95:  log likelihood = -1765.79,  aic = 3539.58
```

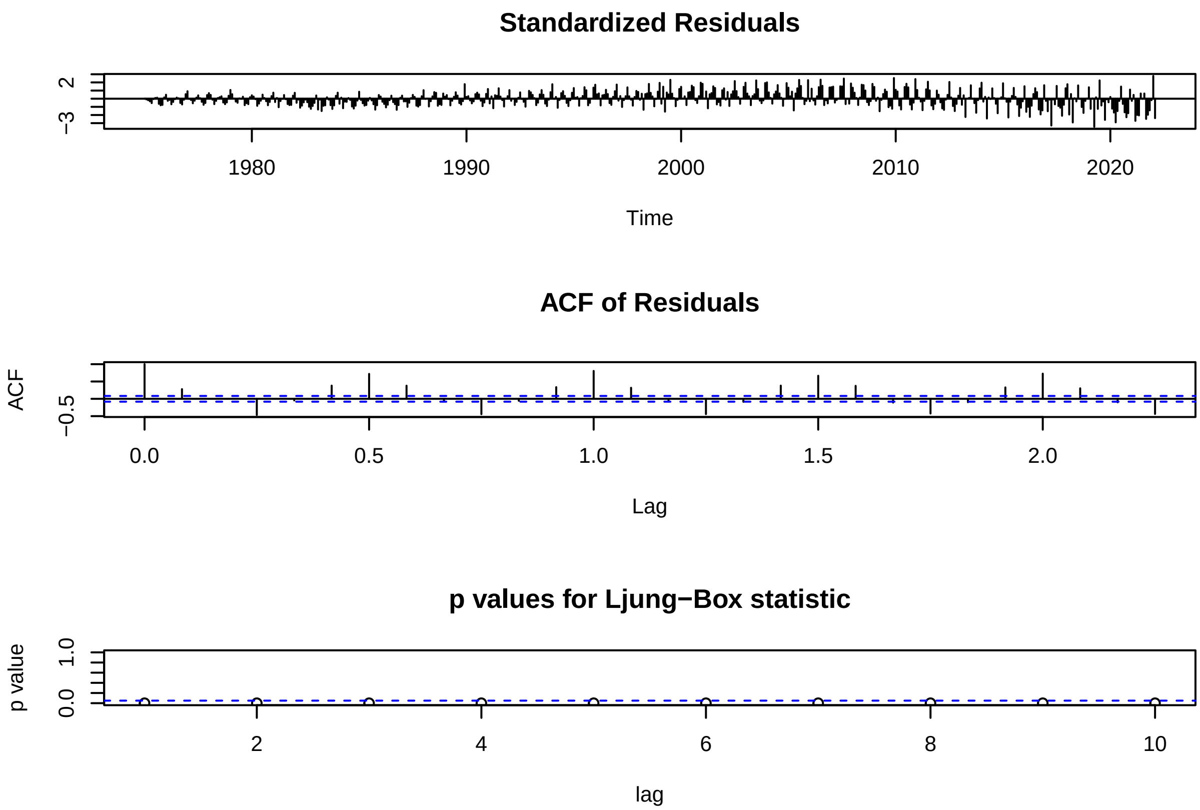
```r
tsdiag(varve.arma, gof.lag=20)
```




```r
varve.arma = arima(detrended, order = c(0, 0, 1))
varve.arma
```

```
##
## Call:
## arima(x = detrended, order = c(0, 0, 1))
##
## Coefficients:
##          ma1  intercept
##       0.7583    -0.0198
## s.e.  0.0196     0.4303
##
## sigma^2 estimated as 33.95:  log likelihood = -1801.12,  aic = 3608.24
```

```r
tsdiag(varve.arma, gof.lag=10)
```



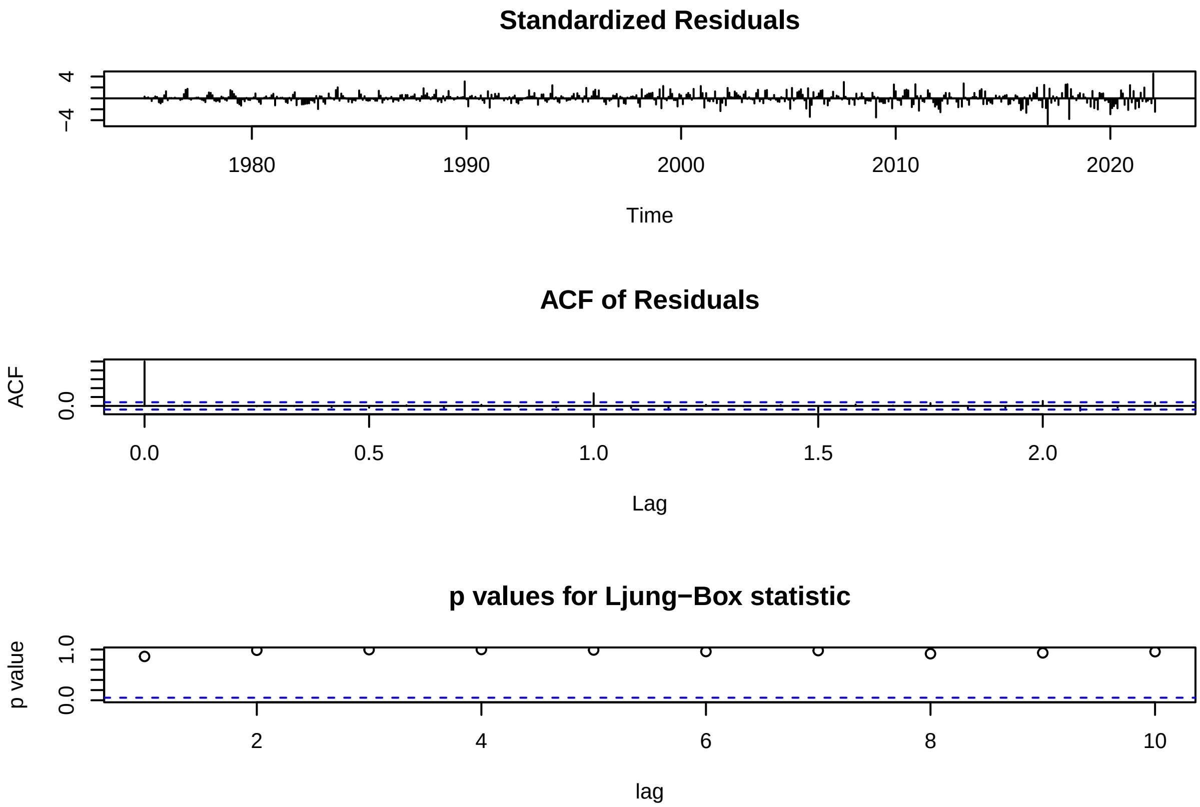

```r
varve.arma = arima(detrended, order = c(7, 0, 12))
```

```
## Warning in arima(detrended, order = c(7, 0, 12)): possible convergence problem:
## optim gave code = 1
```

```r
varve.arma
```

```
##
## Call:
## arima(x = detrended, order = c(7, 0, 12))
##
## Coefficients:

## Warning in sqrt(diag(x$var.coef)): NaNs produced

##         ar1      ar2      ar3     ar4      ar5     ar6     ar7      ma1     ma2
##       0.564  -0.0631  -0.5427  0.5793  -0.0374  0.4309  0.0399  -0.1234  0.0162
## s.e.    NaN      NaN      NaN     NaN      NaN     NaN     NaN      NaN     NaN
##          ma3      ma4     ma5     ma6      ma7     ma8      ma9     ma10
##       0.7135  -0.2863  0.0266  -0.374  -0.1988  0.0555  -0.1785  -0.0052
## s.e.     NaN      NaN     NaN     NaN      NaN  0.0280      NaN   0.0380
##         ma11     ma12  intercept
##       0.0630  0.0473    -2.7035
## s.e.  0.0266     NaN     3.5611
##
## sigma^2 estimated as 7.094:  log likelihood = -1363.1,  aic = 2768.19
```

```r
tsdiag(varve.arma, gof.lag=10)
```



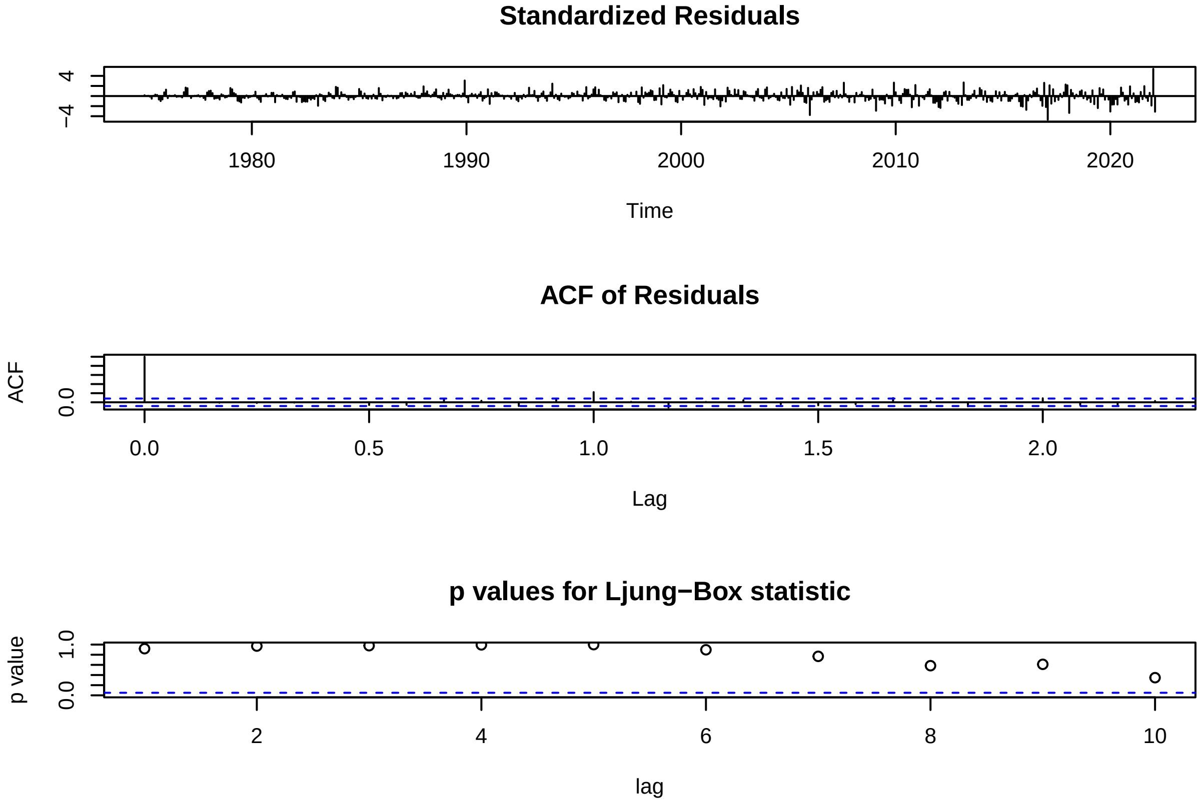

```r
varve.arma = arima(detrended, order = c(6, 0, 11))
```

```
## Warning in arima(detrended, order = c(6, 0, 11)): possible convergence problem:
## optim gave code = 1
```

```r
varve.arma
```

```
##
## Call:
## arima(x = detrended, order = c(6, 0, 11))
##
## Coefficients:
##           ar1     ar2      ar3      ar4     ar5     ar6     ma1      ma2
##       -0.5157  0.7063  -0.1784  -0.5597  0.7537  0.7750  0.9979  -0.2561
## s.e.   0.1425  0.0492   0.1482   0.1578  0.0449  0.1352  0.1639   0.1288
##          ma3     ma4      ma5      ma6      ma7      ma8      ma9     ma10
##       0.2036  0.9016  -0.3348  -0.9430  -0.1437  -0.0364  -0.3570  -0.0843
## s.e.  0.1179  0.1462   0.1864   0.1247   0.1760   0.1130   0.1162   0.1511
##         ma11  intercept
##       0.0938    -1.7701
## s.e.  0.2049     4.1497
##
## sigma^2 estimated as 6.86:  log likelihood = -1354.74,  aic = 2747.48
```

```r
tsdiag(varve.arma, gof.lag=10)
```



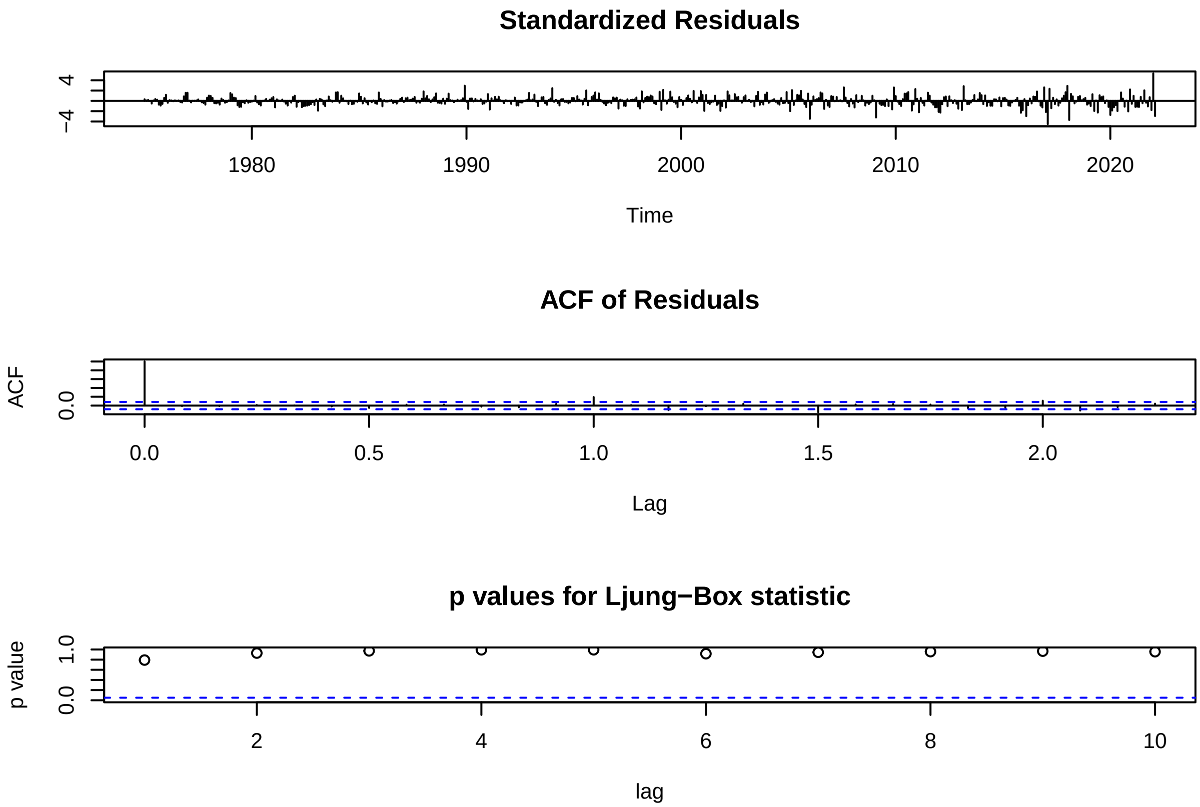

```r
varve.arma = arima(detrended, order = c(7, 0, 13))
```

```
## Warning in arima(detrended, order = c(7, 0, 13)): possible convergence problem:
## optim gave code = 1
```

```r
varve.arma
```

```
##
## Call:
## arima(x = detrended, order = c(7, 0, 13))
##
## Coefficients:
##           ar1     ar2      ar3      ar4     ar5     ar6     ar7     ma1     ma2
##       -0.6789  0.2744  -0.0565  -0.2584  0.3066  0.9127  0.4488  1.1704  0.2933
## s.e.   0.1170  0.0809   0.0834   0.0988  0.0915  0.0799  0.1317  0.1123  0.1424
##          ma3     ma4     ma5      ma6      ma7      ma8      ma9     ma10
##       0.2877  0.5759  0.1428  -0.7627  -0.7347  -0.2719  -0.1299  -0.0845
## s.e.  0.0928  0.1024  0.1158   0.0958   0.1243   0.0887   0.0838   0.0821
##          ma11    ma12    ma13  intercept
##       -0.1225  0.1434  0.1890    -2.8537
## s.e.   0.0885  0.0733  0.0388     3.5748
##
## sigma^2 estimated as 6.783:  log likelihood = -1350.85,  aic = 2745.69
```

```r
tsdiag(varve.arma, gof.lag=10)
```



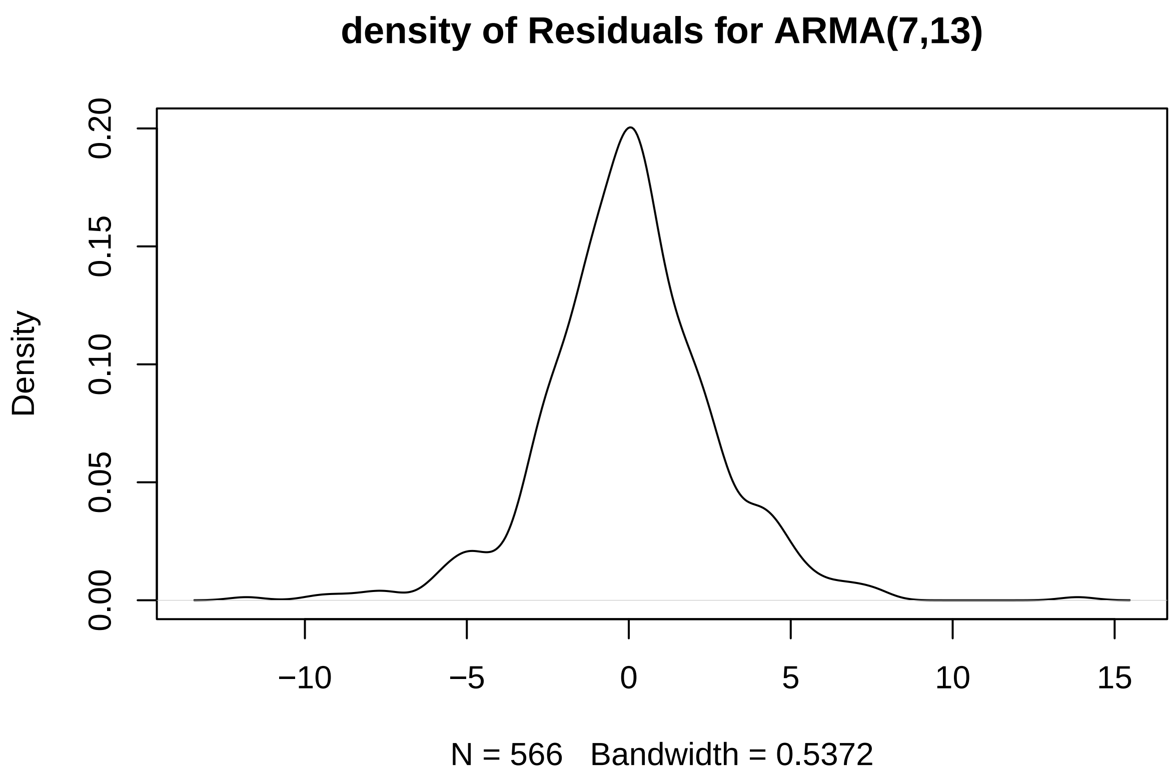

```r
arima(detrended, order = c(1, 0, 1))
```

```
##
## Call:
## arima(x = detrended, order = c(1, 0, 1))
##
## Coefficients:
##          ar1     ma1  intercept
##       0.4002  0.6304    -0.0377
## s.e.  0.0435  0.0303     0.6242
##
## sigma^2 estimated as 29.95:  log likelihood = -1765.79,  aic = 3539.58
```

```r
a22 <- arima(detrended, order = c(7, 0, 13))
```

```
## Warning in arima(detrended, order = c(7, 0, 13)): possible convergence problem:
## optim gave code = 1
```

```r
library(astsa)

plot(density(a22$residuals), main = "density of Residuals for ARMA(7,13)")
```




```
mod4 = sarima(df, 7,0,13, no.constant=T, details=F)
```

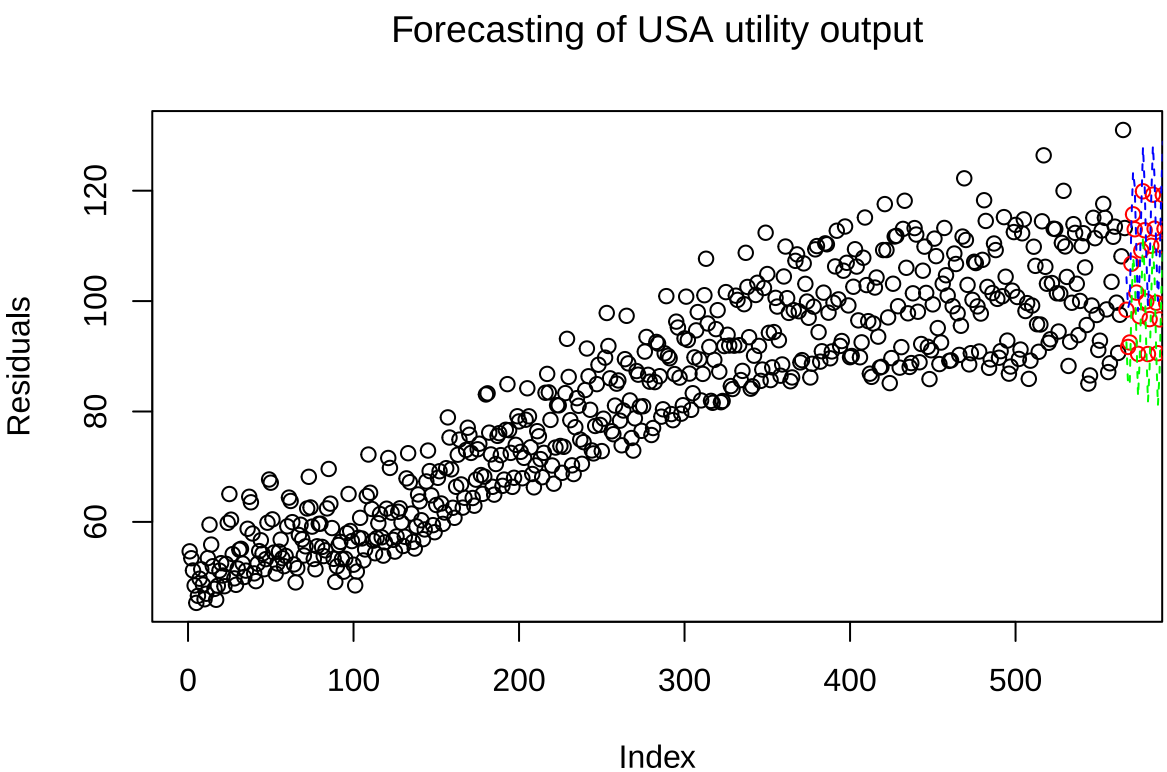
```
## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced

## Warning in arima(xdata, order = c(p, d, q), seasonal = list(order = c(P, :
## possible convergence problem: optim gave code = 1

## Warning in sqrt(diag(fitit$var.coef)): NaNs produced

## Warning in sqrt(diag(fitit$var.coef)): NaNs produced
```

```
mod4a = arima(df, order=c(7,0,13), include.mean=F)
```

```
## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced

## Warning in arima(df, order = c(7, 0, 13), include.mean = F): possible
## convergence problem: optim gave code = 1
```

```
mod4.pr = predict(mod4a, n.ahead=2)

PI.nov = c(mod4.pr$pr[1] - 2*mod4.pr$se[1], mod4.pr$pr[1] + 2*mod4.pr$se[1])
PI.dec = c(mod4.pr$pr[2] - 2*mod4.pr$se[2], mod4.pr$pr[2] + 2*mod4.pr$se[2])



mod4.pr = predict(mod4a, n.ahead=45, interval = "pred")

zhat = mod4.pr$pr
pi.z.upper = mod4.pr$pr + 2*mod4.pr$se
```

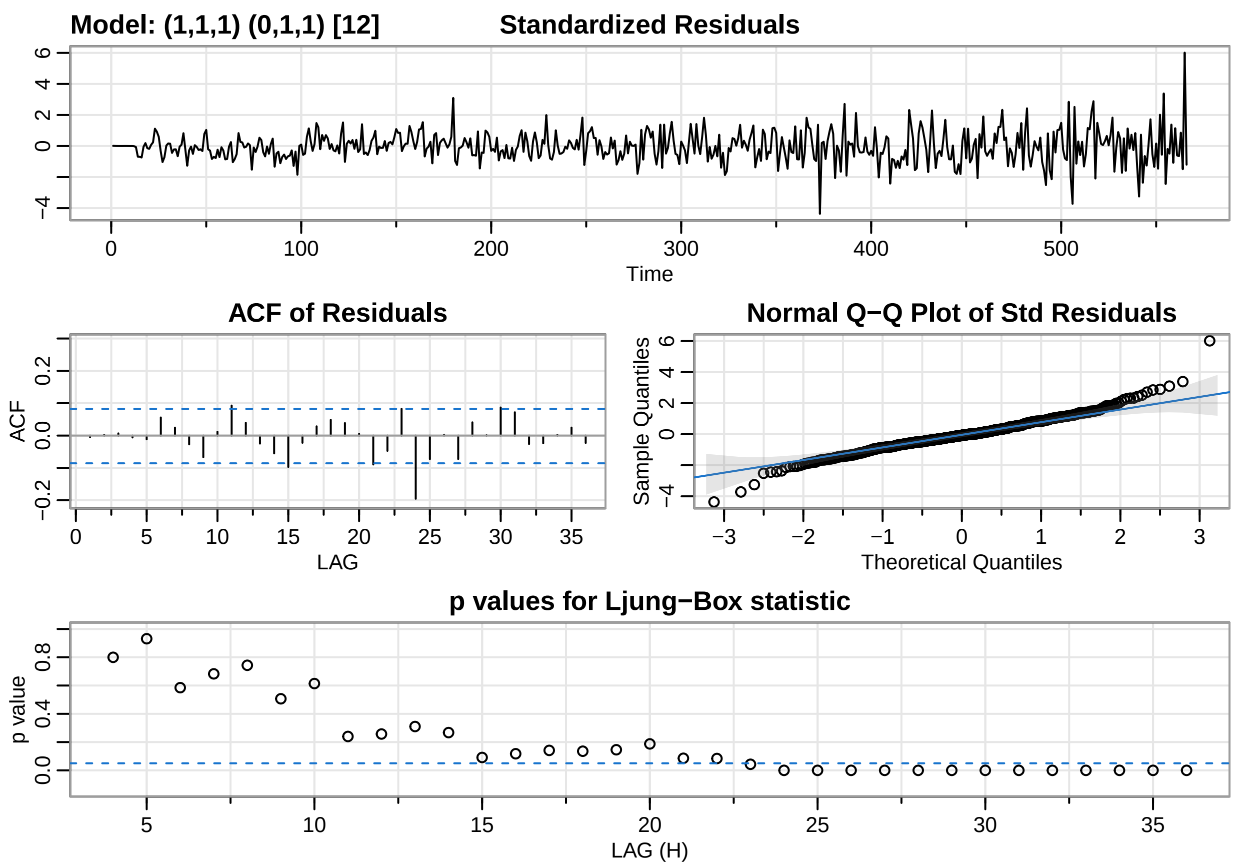
```r
pi.z.lower = mod4.pr$pr - 2*mod4.pr$se


plot(df, ylab="Residuals", main=expression("Forecasting of USA utility output"))
points(mod4.pr$pred, col="red")
lines(pi.z.upper, lty=2, col="blue")
lines(pi.z.lower, lty=2, col="green")
```




```r
mod4a2 <- sarima(df, 1, 1, 1, P = 0, D = 1, Q = 1, S = 12)
```

```
## initial  value 1.196814
## iter   2 value 1.060460
## iter   3 value 0.966577
## iter   4 value 0.959790
## iter   5 value 0.955861
## iter   6 value 0.941389
## iter   7 value 0.930513
## iter   8 value 0.922721
## iter   9 value 0.921147
## iter  10 value 0.904351
## iter  11 value 0.895591
## iter  12 value 0.886471
## iter  13 value 0.885673
## iter  14 value 0.885659
## iter  15 value 0.885601
## iter  16 value 0.885599
## iter  17 value 0.885599
## iter  17 value 0.885599
## iter  17 value 0.885599
## final  value 0.885599
```

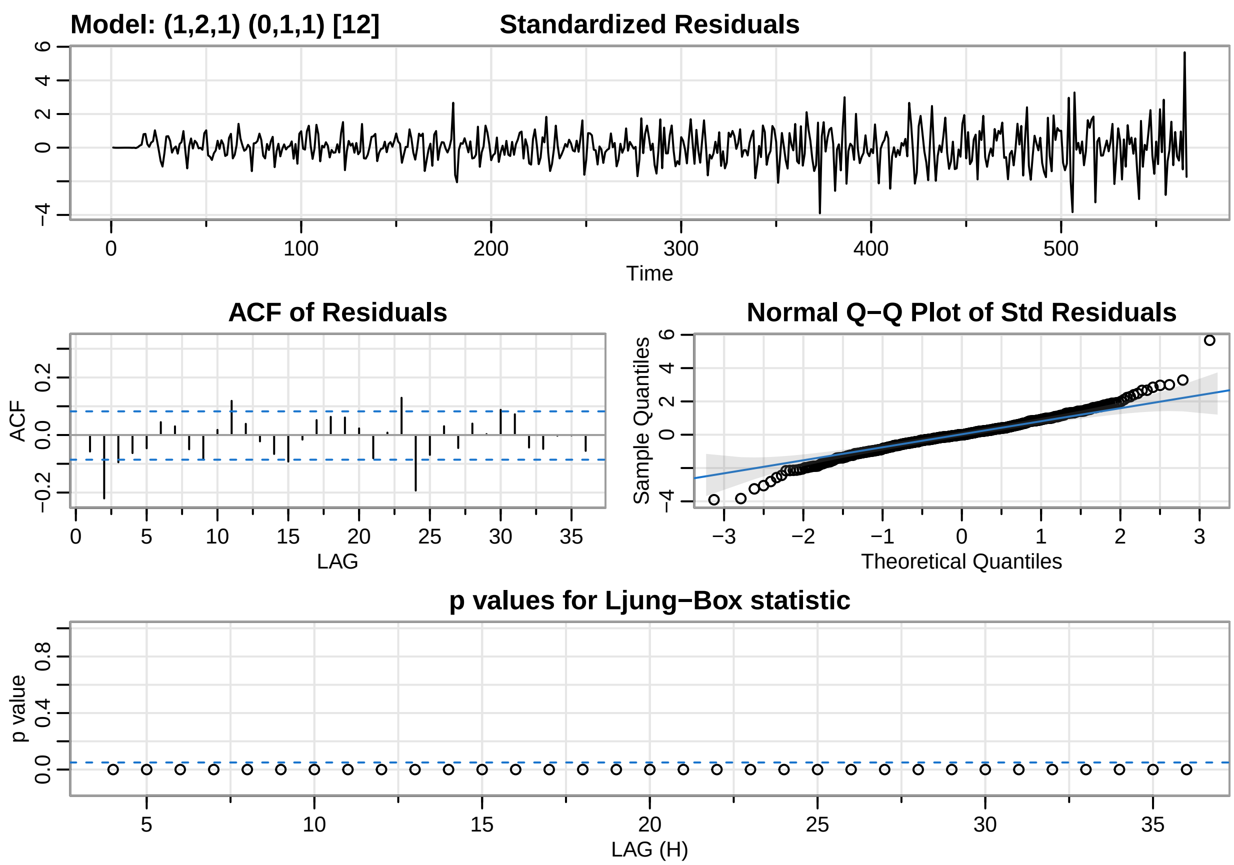
```
## converged
## initial  value 0.895738
## iter   2 value 0.895658
## iter   3 value 0.895657
## iter   4 value 0.895604
## iter   5 value 0.895603
## iter   5 value 0.895603
## iter   5 value 0.895603
## final  value 0.895603
## converged
```




```
mod4a3 <- sarima(df, 1, 2, 1, P = 0, D = 1, Q = 1, S = 12)
```

```
## initial  value 1.664193
## iter   2 value 1.249103
## iter   3 value 1.196355
## iter   4 value 1.030922
## iter   5 value 1.027807
## iter   6 value 1.022797
## iter   7 value 1.019772
## iter   8 value 1.019676
## iter   9 value 1.019633
## iter  10 value 1.019633
## iter  11 value 1.019633
## iter  11 value 1.019633
## iter  11 value 1.019633
## final  value 1.019633
```

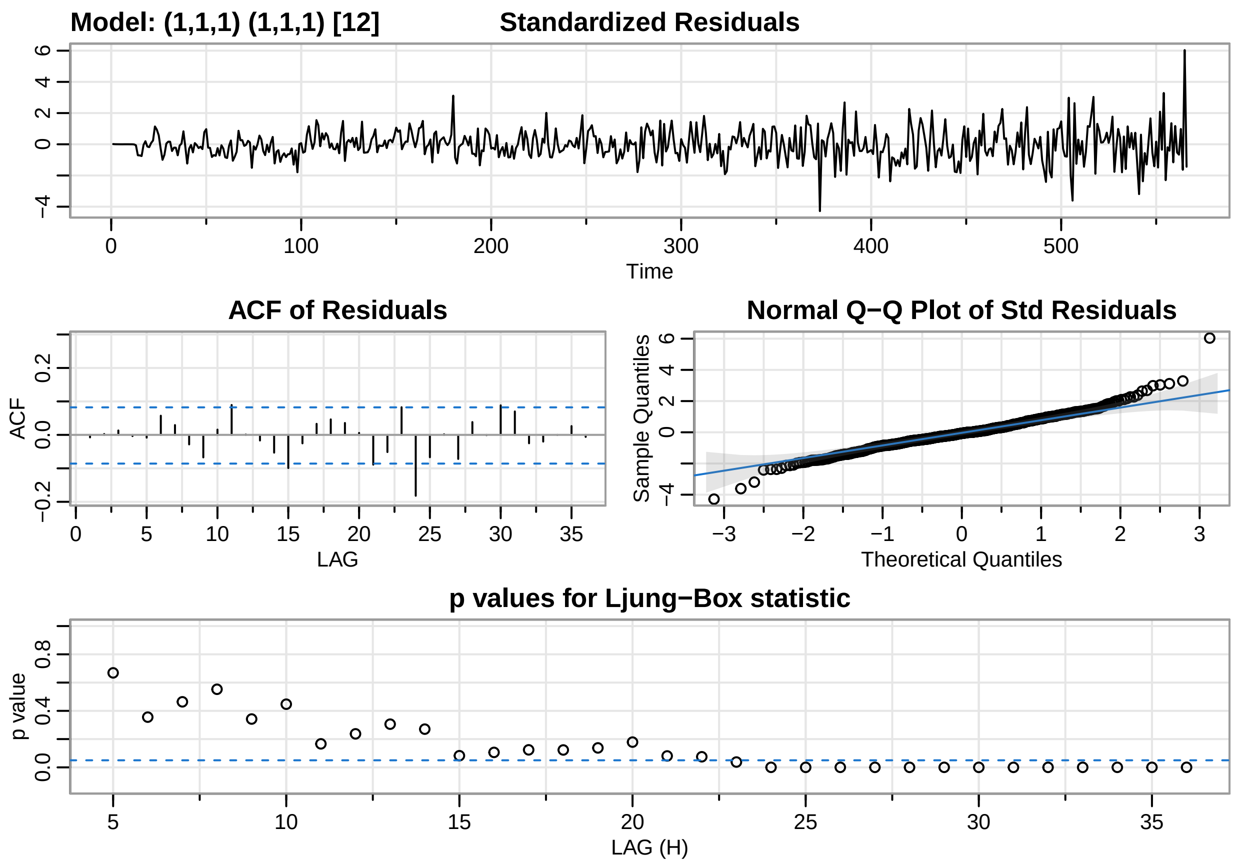
```
## converged
## initial  value 1.015640
## iter   2 value 0.992087
## iter   3 value 0.989338
## iter   4 value 0.988499
## iter   5 value 0.988386
## iter   6 value 0.988363
## iter   7 value 0.988361
## iter   8 value 0.988361
## iter   8 value 0.988361
## iter   8 value 0.988361
## final  value 0.988361
## converged
```




```r
mod4a4 <- sarima(df, 1, 1, 1, P = 1, D = 1, Q = 1, S = 12)
```

```
## initial  value 1.206077
## iter   2 value 1.061995
## iter   3 value 1.018643
## iter   4 value 0.989856
## iter   5 value 0.975888
## iter   6 value 0.958861
## iter   7 value 0.945745
## iter   8 value 0.941229
## iter   9 value 0.924644
## iter  10 value 0.913846
## iter  11 value 0.910599
```

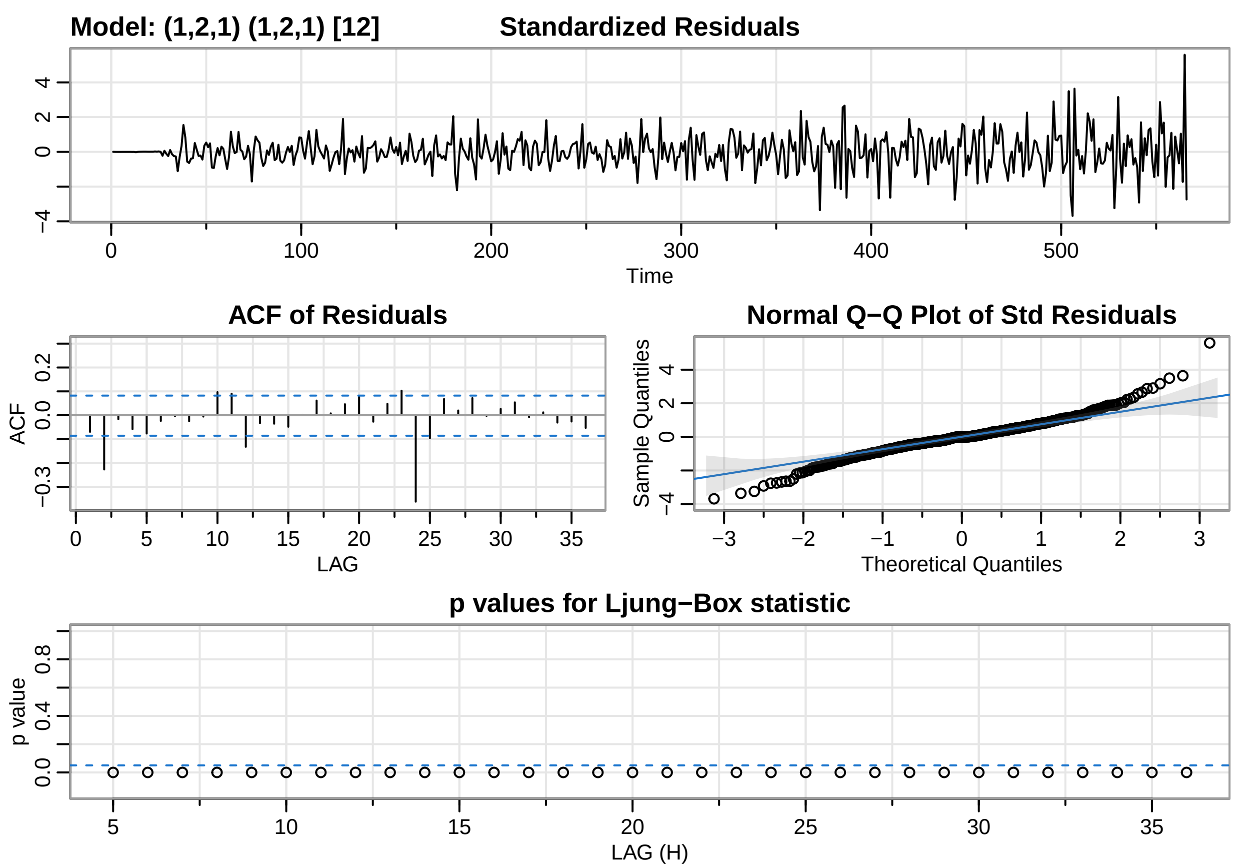
```
## iter  12 value 0.902243
## iter  13 value 0.900396
## iter  14 value 0.900382
## iter  15 value 0.899479
## iter  16 value 0.899245
## iter  17 value 0.899178
## iter  18 value 0.899177
## iter  19 value 0.899177
## iter  19 value 0.899177
## final  value 0.899177
## converged
## initial  value 0.895335
## iter   2 value 0.894065
## iter   3 value 0.893668
## iter   4 value 0.893609
## iter   5 value 0.893553
## iter   6 value 0.893538
## iter   7 value 0.893537
## iter   8 value 0.893536
## iter   8 value 0.893536
## iter   8 value 0.893536
## final  value 0.893536
## converged
```




```r
mod4a5 <- sarima(df, 1, 2, 1, P = 1, D = 2, Q = 1, S = 12)
```

```
## initial  value 2.132952
```

```
## iter   2 value 1.506780
## iter   3 value 1.446489
## iter   4 value 1.318096
## iter   5 value 1.260581
## iter   6 value 1.255807
## iter   7 value 1.255537
## iter   8 value 1.255187
## iter   9 value 1.255092
## iter  10 value 1.255035
## iter  11 value 1.255034
## iter  11 value 1.255034
## iter  11 value 1.255034
## final  value 1.255034
## converged
## initial  value 1.237722
## iter   2 value 1.186550
## iter   3 value 1.165124
## iter   4 value 1.164197
## iter   5 value 1.163324
## iter   6 value 1.162741
## iter   7 value 1.162714
## iter   8 value 1.162713
## iter   8 value 1.162713
## iter   8 value 1.162713
## final  value 1.162713
## converged
```

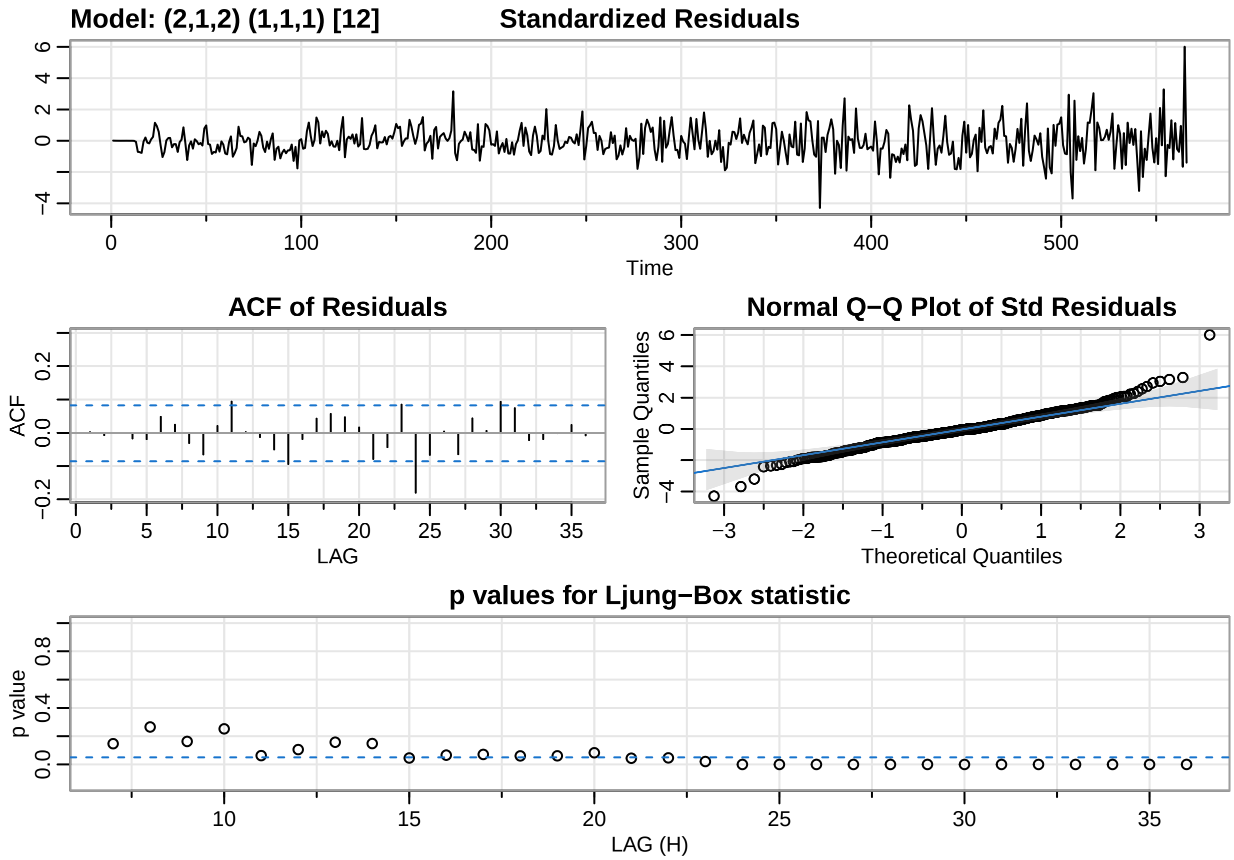


```r
mod4a6 <- sarima(df, 2, 1, 2, P = 1, D = 1, Q = 1, S = 12)
```

```
## initial  value 1.206557
## iter   2 value 1.030166
## iter   3 value 0.987616
## iter   4 value 0.961544
## iter   5 value 0.939853
## iter   6 value 0.921408
## iter   7 value 0.902978
## iter   8 value 0.896040
## iter   9 value 0.894071
## iter  10 value 0.893116
## iter  11 value 0.892428
## iter  12 value 0.892360
## iter  13 value 0.892351
## iter  14 value 0.892344
## iter  15 value 0.892343
## iter  16 value 0.892339
## iter  17 value 0.892328
## iter  18 value 0.892278
## iter  19 value 0.892188
## iter  20 value 0.892050
## iter  21 value 0.892023
## iter  22 value 0.892019
## iter  23 value 0.892019
## iter  23 value 0.892019
## iter  23 value 0.892019
## final  value 0.892019
## converged
## initial  value 0.893879
## iter   2 value 0.893708
## iter   3 value 0.893600
## iter   4 value 0.893554
## iter   5 value 0.893488
## iter   6 value 0.893487
## iter   7 value 0.893487
## iter   8 value 0.893486
## iter   9 value 0.893486
## iter  10 value 0.893486
## iter  11 value 0.893486
## iter  12 value 0.893486
## iter  13 value 0.893485
## iter  14 value 0.893483
## iter  15 value 0.893482
## iter  16 value 0.893481
## iter  17 value 0.893480
## iter  18 value 0.893480
## iter  19 value 0.893480
## iter  20 value 0.893480
## iter  21 value 0.893480
## iter  22 value 0.893480
## iter  23 value 0.893479
## iter  24 value 0.893478
## iter  25 value 0.893472
```

```
## iter  26 value 0.893466
## iter  27 value 0.893431
## iter  28 value 0.893409
## iter  29 value 0.893321
## iter  30 value 0.893313
## iter  31 value 0.893305
## iter  32 value 0.893302
## iter  33 value 0.893302
## iter  34 value 0.893302
## iter  35 value 0.893302
## iter  36 value 0.893300
## iter  37 value 0.893297
## iter  38 value 0.893287
## iter  39 value 0.893273
## iter  40 value 0.893271
## iter  41 value 0.893261
## iter  42 value 0.893252
## iter  43 value 0.893249
## iter  44 value 0.893249
## iter  45 value 0.893248
## iter  46 value 0.893247
## iter  47 value 0.893243
## iter  48 value 0.893234
## iter  49 value 0.893214
## iter  50 value 0.893186
## iter  51 value 0.893141
## iter  52 value 0.893087
## iter  53 value 0.893085
## iter  54 value 0.893054
## iter  55 value 0.893047
## iter  56 value 0.893042
## iter  57 value 0.893041
## iter  58 value 0.893040
## iter  59 value 0.893036
## iter  60 value 0.893025
## iter  61 value 0.892991
## iter  62 value 0.892970
## iter  63 value 0.892944
## iter  64 value 0.892940
## iter  65 value 0.892859
## iter  66 value 0.892859
## iter  67 value 0.892850
## iter  68 value 0.892836
## iter  69 value 0.892835
## iter  70 value 0.892834
## iter  71 value 0.892832
## iter  72 value 0.892828
## iter  73 value 0.892816
## iter  74 value 0.892788
## iter  75 value 0.892761
## iter  76 value 0.892741
## iter  77 value 0.892733
## iter  77 value 0.892733
## iter  78 value 0.892733
```

```
## iter  79 value 0.892733
## iter  79 value 0.892733
## iter  79 value 0.892733
## final  value 0.892733
## converged
```




```r
mod4a2$AIC
```

```
## [1] 4.64355
```

```r
mod4a3$AIC
```

```
## [1] 4.832716
```

```r
mod4a4$AIC
```

```
## [1] 4.643033
```

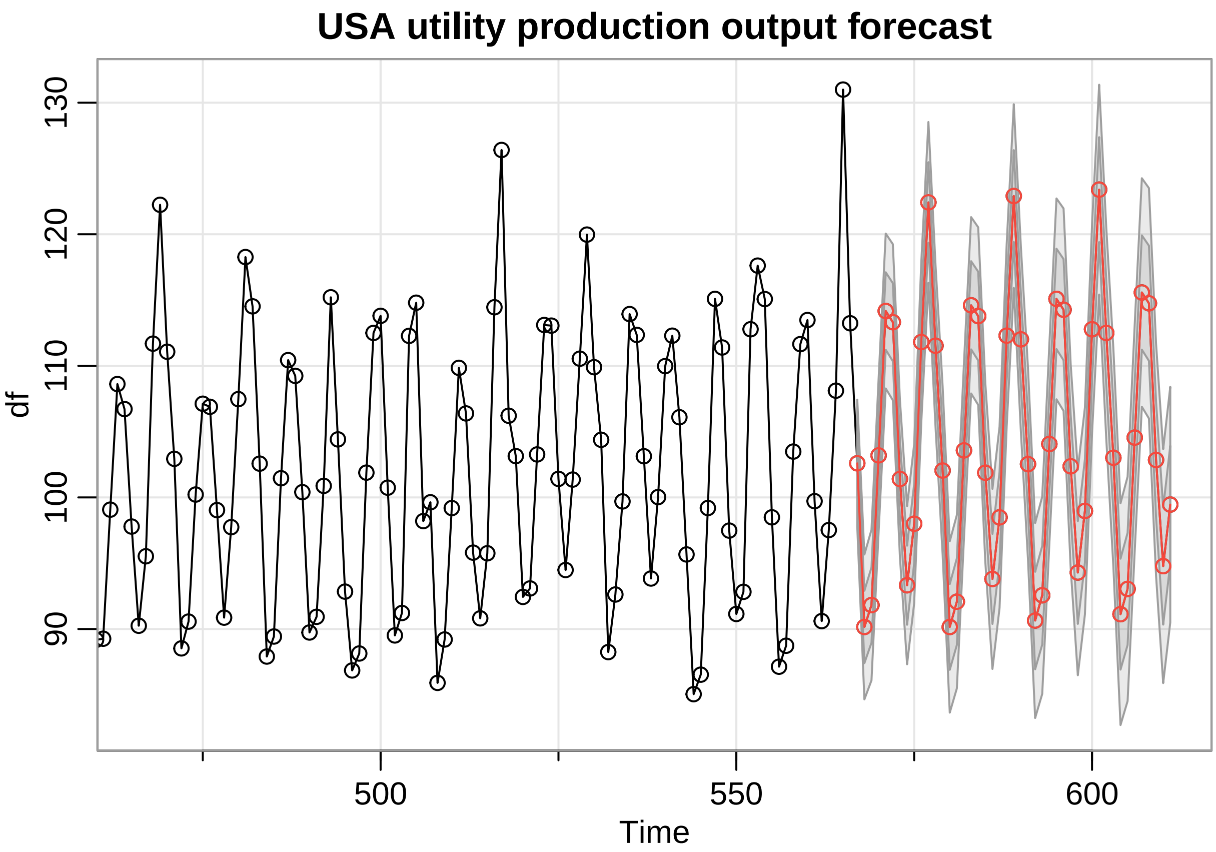
```r
mod4a5$AIC
```

```
## [1] 5.181823
```

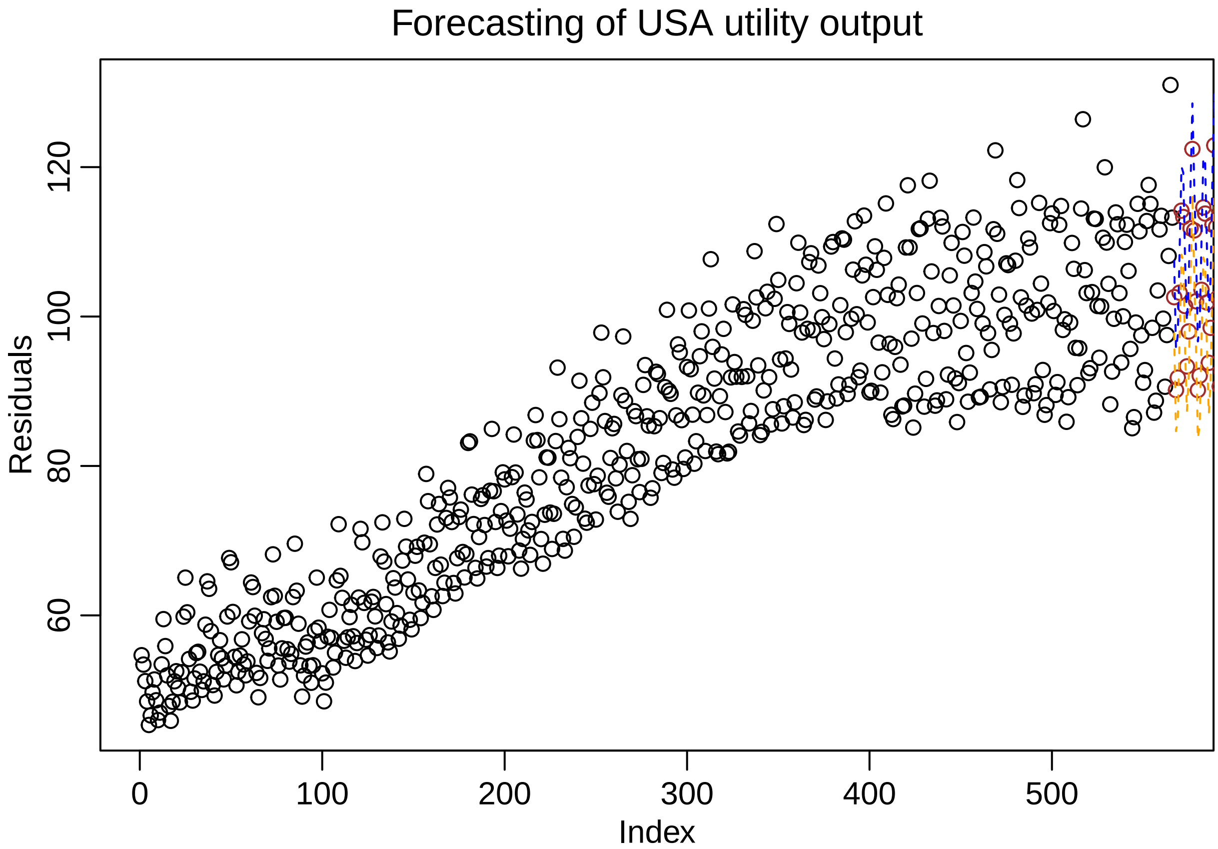
```r
mod4a6$AIC
```

```
## [1] 4.648659
```

```
mod4a <- sarima.for(df,n.ahead = 45,1,1,1,P=0,D=1,Q=1,S=12, main = "USA utility production output foreca
```




```
#mod4.pr = predict(mod4a, n.ahead=45, interval = "pred")

mod4.pr = mod4a



PI.nov = c(mod4.pr$pr[1] - 2*mod4.pr$se[1], mod4.pr$pr[1] + 2*mod4.pr$se[1])
PI.dec = c(mod4.pr$pr[2] - 2*mod4.pr$se[2], mod4.pr$pr[2] + 2*mod4.pr$se[2])


zhat = mod4.pr$pr
pi.z.upper = mod4.pr$pr + 2*mod4.pr$se
pi.z.lower = mod4.pr$pr - 2*mod4.pr$se


plot(df, ylab="Residuals", main=expression("Forecasting of USA utility output"))
points(mod4.pr$pred, col="brown")
lines(pi.z.upper, lty=2, col="blue")
lines(pi.z.lower, lty=2, col="orange")
```


Forecasting of USA utility output
Residuals
120
100
80
60
0
100
200
300
400
500
Index