Module 7 - Model Diagnostics

MATH1318 Time Series Analysis

Prepared by: Dr. Haydar Demirhan based on the textbook by Cryer and Chan, Time Series Analysis with R, Springer, 2008.

Nomenclature

π_i: Coefficients of residual series.
Q: Box and Pierce statistic.
$\hat{r}_k$: Sample autocorrelation at lag k.

Introduction

After specification of orders of autoregressive and moving average parts, we estimated parameters of specified ARMA models. In this module, we will concern with testing the goodness of fit of the model. If the goodness of fit is poor, we will suggest appropriate modifications. We will focus on two complementary approaches:

- analysis of residuals from the fitted model and
- analysis of overparameterized models; that is, models that are more general than the proposed model but that contain the proposed model as a special case.

Residual Analysis

Consider the following AR(2) model:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + e_t$$

After the estimation of parameters ϕ_1, ϕ_2, and θ_0, the residuals are defined as

$$\hat{e}_t = Y_t - \hat{\phi}_1 Y_{t-1} + \hat{\phi}_2 Y_{t-2} - \hat{\theta}_0.$$

For general ARMA models containing moving average terms, the residuals are defined as

$$\hat{e}_t = Y_t - \hat{\pi}_1 Y_{t-1} - \hat{\pi}_2 Y_{t-2} - \hat{\pi}_3 Y_{t-3} - \cdots .$$

Here the π's are not estimated directly but rather implicitly as functions of the ϕ's and θ's. So, residual can be thought of **residual = actual - predicted**.

> If the model is correctly specified and the parameter estimates are reasonably close to the true values, then the residuals should have nearly the properties of white noise. They should behave roughly like independent, identically distributed normal variables with zero means and common standard deviations. Deviations from these properties can help us discover a more appropriate model.

Plots of the Residuals

Our first diagnostic check is to inspect a plot of the residuals over time.

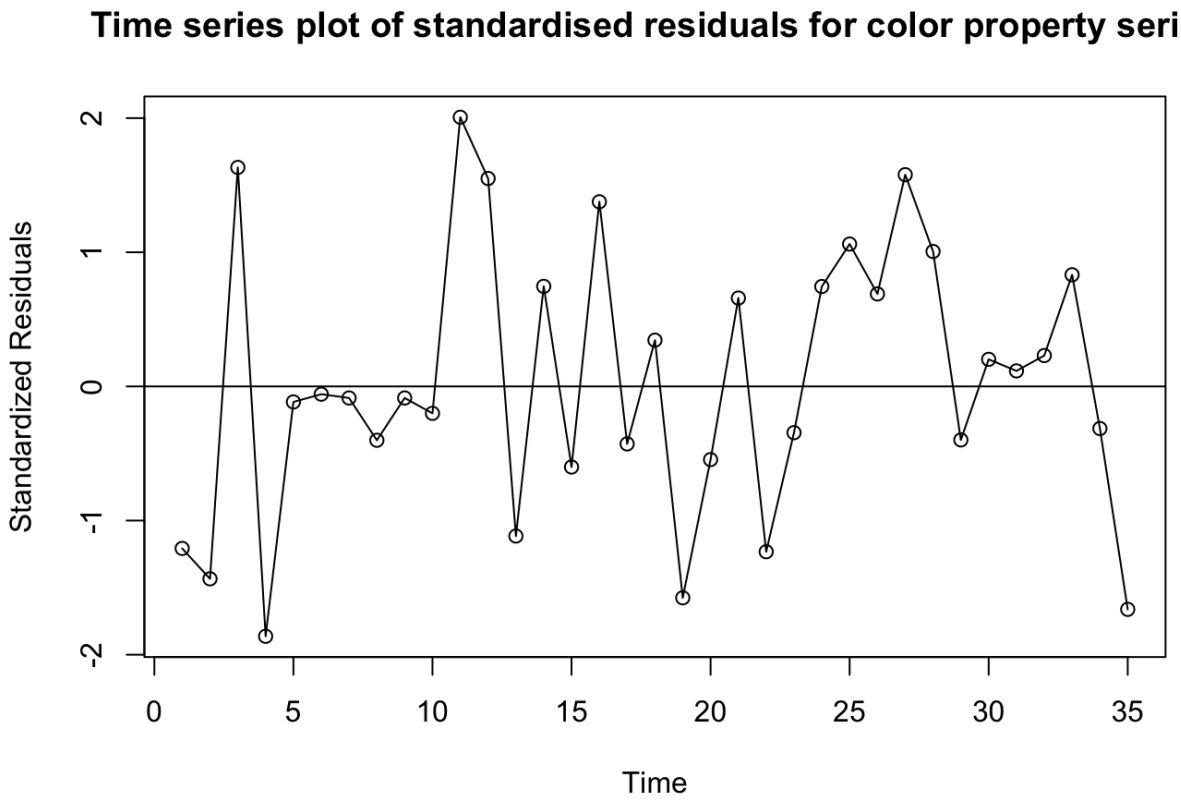
> If the model is adequate, we expect the plot to suggest a rectangular scatter around a zero horizontal level with no trends whatsoever.

The following shows a plot for the standardized residuals from the AR(1) model fitted to the industrial color property series. The parameters were estimated using maximum likelihood.

```
data(color)
m1.color=arima(color,order=c(1,0,0))
m1.color
```

```
##
## Call:
## arima(x = color, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.5705    74.3293
## s.e.  0.1435     1.9151
##
## sigma^2 estimated as 24.83:  log likelihood = -106.07,  aic = 216.15
```

```
plot(rstandard(m1.color),ylab ='Standardized Residuals',type='o',main=
  "Time series plot of standardised residuals for color property serie
  s")
abline(h=0)
```



Standardization allows us to see residuals of unusual size much more easily. Because there is no trend, this plot supports the specified model.

Consider the Canadian hare abundance series. When we estimate a subset AR(3) model with ϕ_2 set to zero, we get the following prediction model:

$$\sqrt{Y_t} = 3.483 + 0.919\sqrt{Y_{t-1}} - 0.5313\sqrt{Y_{t-3}} + e_t.$$

The time series plot of the standardized residuals from this model is shown below. Note that the intercept term given in R is actually the mean in the centered form of the ARMA model; that is, if `y(t)=sqrt(hare)-intercept` , then the model is $Y_t = 0.919Y_{t-1} - 0.5313Y_{t-3} + e_t$. So the true intercept equals $5.6889(1 - 0.919 + 0.5313) = 3.483$.

```
data(hare)
m1.hare=arima(sqrt(hare),order=c(3,0,0)); m1.hare
```

```
##
## Call:
## arima(x = sqrt(hare), order = c(3, 0, 0))
##
## Coefficients:
##          ar1      ar2      ar3  intercept
##       1.0519  -0.2292  -0.3931     5.6923
## s.e.  0.1877   0.2942   0.1915     0.3371
##
## sigma^2 estimated as 1.066:  log likelihood = -46.54,  aic = 101.08
```

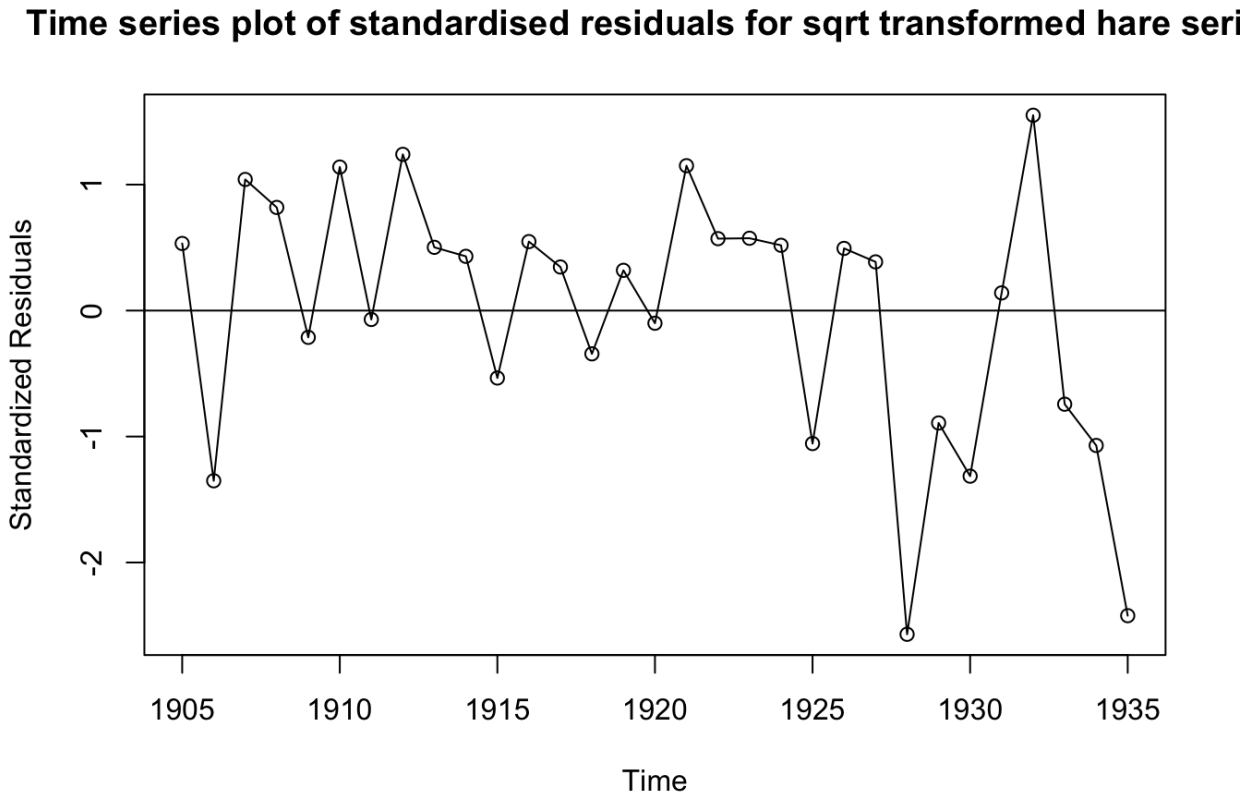
```
m2.hare=arima(sqrt(hare),order=c(3,0,0),fixed=c(NA,0,NA,NA))
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, x
reg =
## xreg, : some AR parameters were fixed: setting transform.pars = FALS
E
```

```
m2.hare
```

```
##
## Call:
## arima(x = sqrt(hare), order = c(3, 0, 0), fixed = c(NA, 0, NA, NA))
##
## Coefficients:
##          ar1  ar2      ar3  intercept
##       0.9190    0  -0.5313     5.6889
## s.e.  0.0791    0   0.0697     0.3179
##
## sigma^2 estimated as 1.088:  log likelihood = -46.85,  aic = 99.69
```

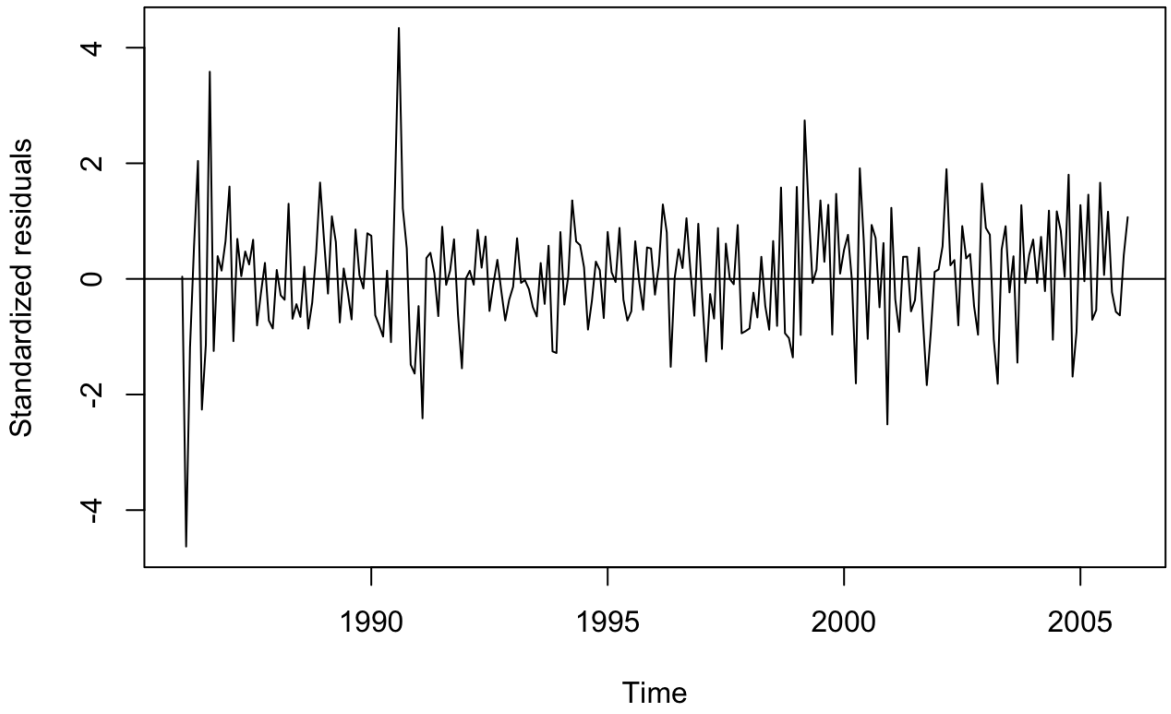
```
plot(rstandard(m2.hare),ylab='Standardized Residuals',type='o',main="Ti
  me series plot of standardised residuals for sqrt transformed hare se
  ries")
abline(h=0)
```



Here we see a possible reduced variation in the middle of the series and increased variation near the end of the series-not exactly an ideal plot of residuals.

The time series plot of the standardized residuals from the IMA(1,1) model estimated for the logarithms of the oil price time series is given below:

```
data(oil.price)
m1.oil=arima(log(oil.price),order=c(0,1,1))
plot(rstandard(m1.oil),ylab='Standardized residuals',type='l')
abline(h=0)
```

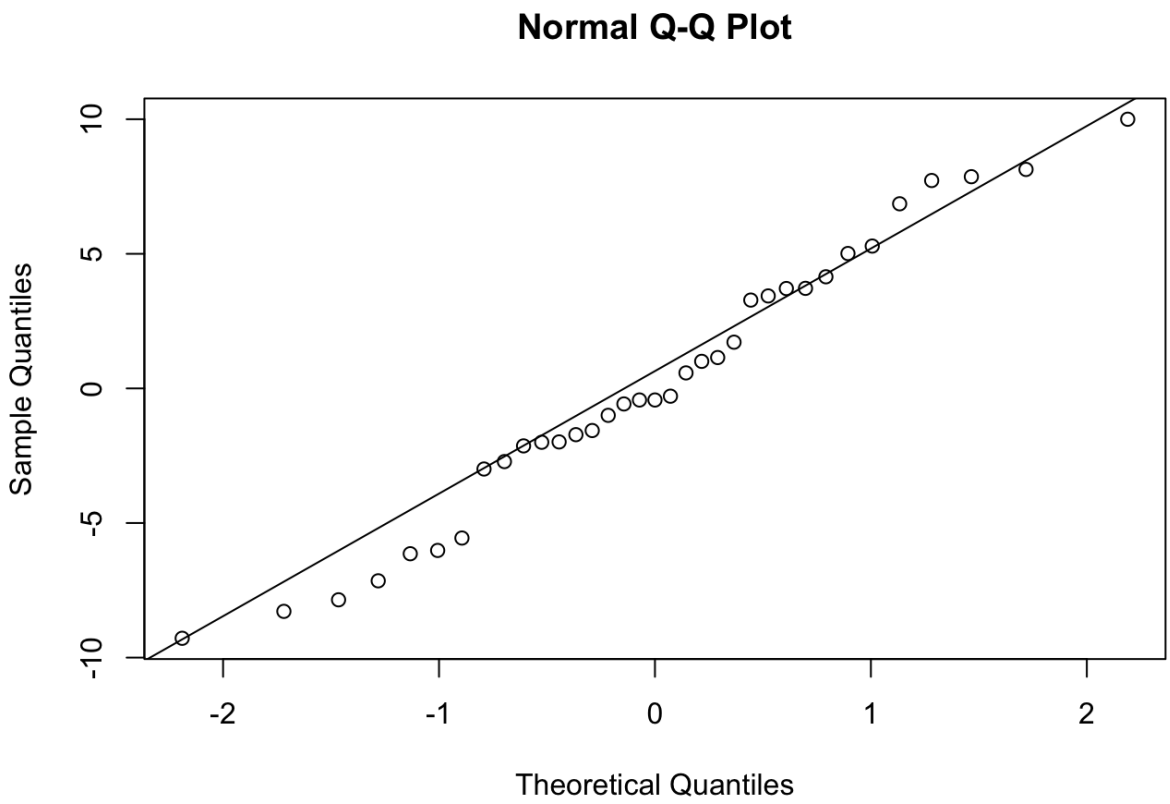


The model was fitted using maximum likelihood estimation. There are at least two or three residuals early in the series with magnitudes larger than 3-very unusual in a standard normal distribution. Ideally, we should go back to those months and try to learn what outside factors may have influenced unusually large drops or unusually large increases in the price of oil.

Normality of the Residuals

We apply quantile-quantile plots for the analysis of normality of residuals. A quantile-quantile plot of the residuals from the AR(1) model estimated for the industrial color property series is given below:

```
e = residuals(m1.color)
qqnorm(e)
qqline(e)
```



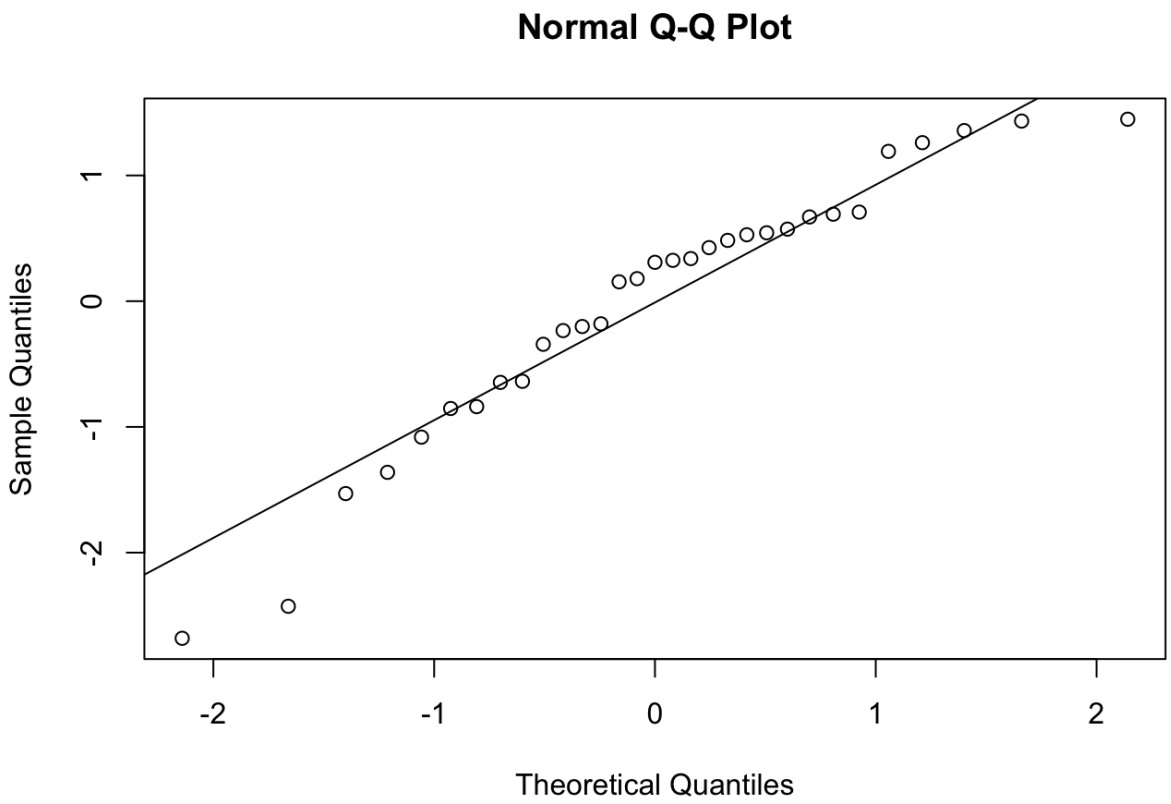

```
shapiro.test(e)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  e
## W = 0.97536, p-value = 0.6057
```

Because the points including the extreme ones closely follow the straight line, we cannot reject normality of the error terms in this model. The Shapiro-Wilk normality test applied to the residuals supports our conclusion. The test statistic for this test is $W = 0.9754$ corresponding to a p-value of 0.6057, and we would not reject normality based on this test.

The quantile-quantile plot for the residuals from the AR(3) model for the square root of the hare abundance time series is displayed below:

```
e = residuals(m1.hare)
qqnorm(e)
qqline(e)
```



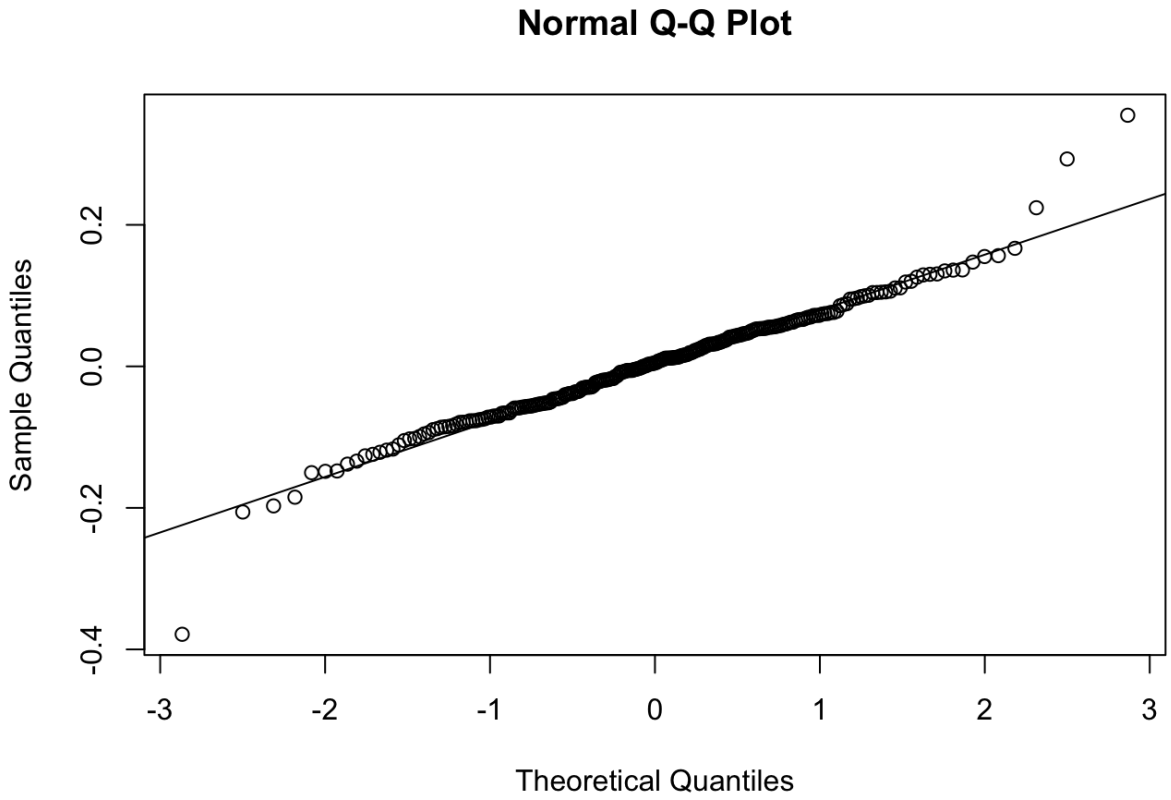

```
shapiro.test(e)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  e
## W = 0.93509, p-value = 0.06043
```

Here the extreme values look suspect. However, the sample is small $(n = 31)$ and, as stated earlier, the Bonferroni criteria for outliers do not indicate the cause for alarm. Also, we are able to reject the null hypothesis at 5% level of significance level with the Shapiro-Wilk test.

The next plot gives the quantile-quantile plot for the residuals from the IMA(1,1) model that was used to model the logarithms of the oil price series.

```
e = residuals(m1.oil)
qqnorm(e)
qqline(e)
```




```
shapiro.test(e)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  e
## W = 0.96883, p-value = 3.937e-05
```

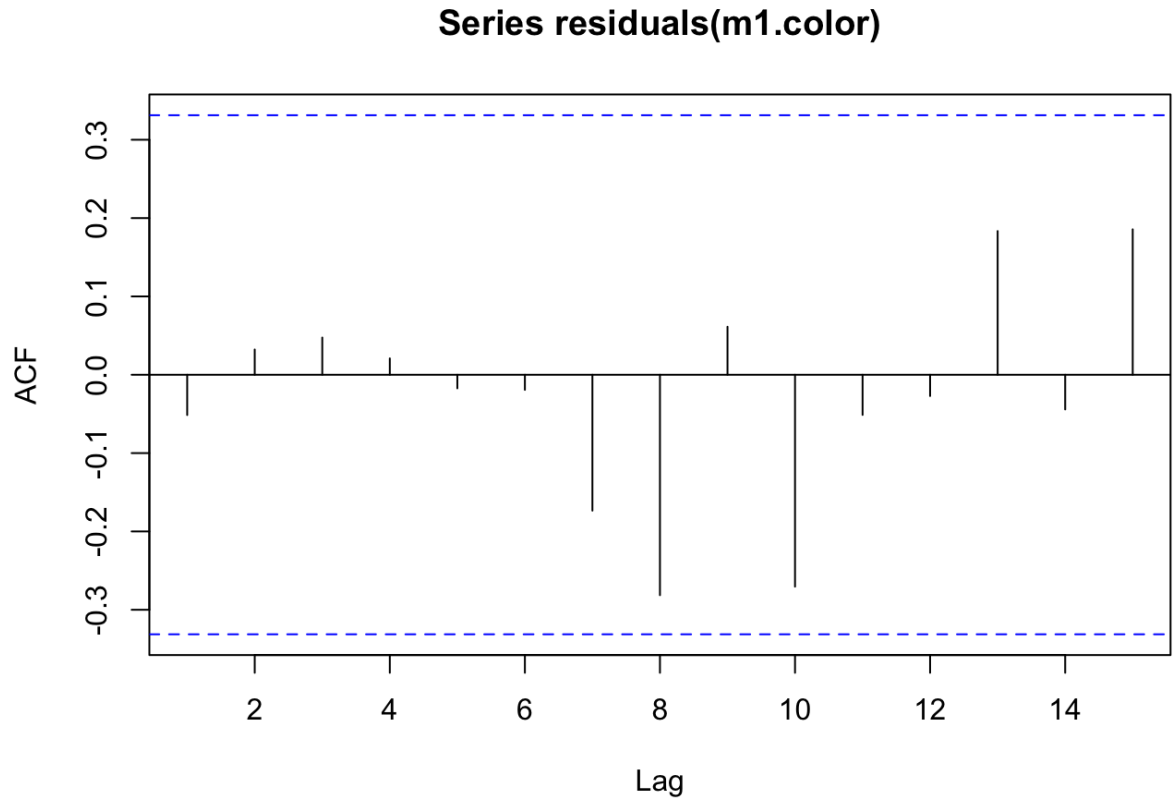
Here we have several outliers apparently. These outliers are influential on the results of the Shapiro-Wilk test.

Autocorrelation of the Residuals

To check on the independence of the noise terms in the model, we consider the sample autocorrelation function of the residuals. We know that for true white noise and large n, the sample autocorrelations are approximately uncorrelated and normally distributed with zero means and variance $1/n$. Unfortunately, even residuals from a correctly specified model with efficiently estimated parameters have somewhat different properties. Thus, we cannot expect to see an exact picture with these properties. However, the sample ACF plot of residuals is still useful to detect anomalies in terms of independence of residuals.

A graph of the sample ACF of residuals resulting from the AR(1) model fitted to the industrial color property time series.

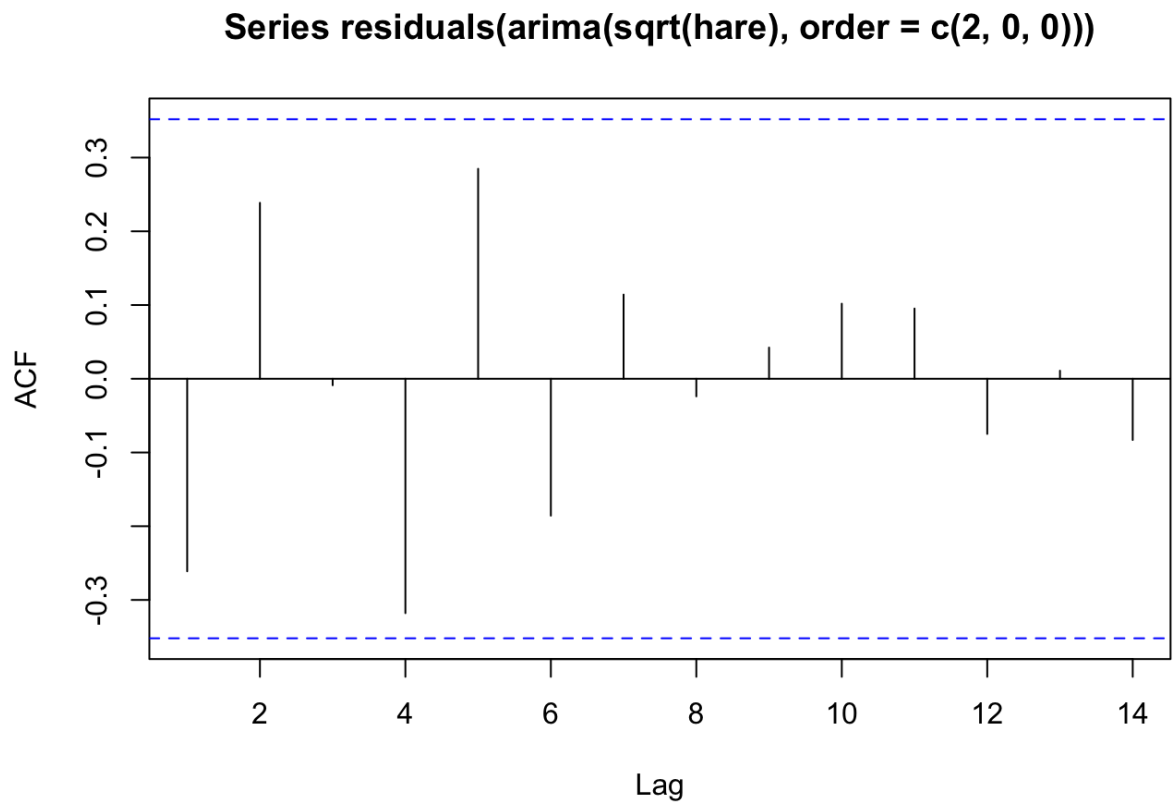
```
acf(residuals(m1.color))
```



There is no evidence of autocorrelation in the residuals of this model.

The following plot shows the sample ACF of the residuals from the AR(2) model of the square root of the hare abundance.

```
acf(residuals(arima(sqrt(hare),order=c(2,0,0))))
```



The lag 1 autocorrelation here equals -0.261, which is close to 2 standard errors below zero but not quite. The lag 4 autocorrelation equals -0.318, but its standard error is 0.169. We conclude that the graph does not show statistically significant evidence of nonzero autocorrelation in the residuals.

The Ljung-Box Test

The Ljung-Box test provides an overall test for looking at residual correlations as a whole. For example, it may be that most of the residual autocorrelations are moderate, some even close to their critical values, but, taken together, they seem excessive. To address this, Box and Pierce (1970) proposed the statistic

$$Q = n(\hat{r}_1^2 + \hat{r}_2^2 + \cdots + \hat{r}_k^2),$$

where $\hat{r}_k$ is sample autocorrelation at lag k. If the correct ARMA(p,q) model is estimated then, for large n, Q has an approximate chi-square distribution with $K - p - q$ degrees of freedom. However, fitting an erroneous model would tend to inflate Q. Thus, a general "portmanteau" test would reject the ARMA(p,q) model if $Q > \chi_{K-p-q}$. Here the maximum lag K is selected somewhat arbitrarily but large enough that the ψ-weights are negligible for $j > K$.

The chi-square distribution for Q is based on a limit theorem as $n \to \infty$, but Ljung and Box (1978) subsequently discovered that even for $n = 100$, the approximation is not satisfactory. By modifying the Q statistic slightly, they defined a test statistic whose null distribution is much closer to chi-square for typical sample sizes. The modified Box-Pierce, or **Ljung-Box**, statistic is given by

$$Q_* = n(n+2)\left(\frac{\hat{r}_1^2}{n-1} + \frac{\hat{r}_2^2}{n-2} + \cdots + \frac{\hat{r}_K^2}{n-K}\right).$$

The following code chunk displays the first six autocorrelations of the residuals from the AR(1) fitted model for the color property series with $n = 35$.

```
acf(residuals(m1.color),plot=F)$acf
```

```
## , , 1
##
##               [,1]
##  [1,] -0.05138241
##  [2,]  0.03224346
##  [3,]  0.04749703
##  [4,]  0.02088157
##  [5,] -0.01729829
##  [6,] -0.01924314
##  [7,] -0.17341651
##  [8,] -0.28136877
##  [9,]  0.06127493
## [10,] -0.27045217
## [11,] -0.05123367
## [12,] -0.02712861
## [13,]  0.18344079
## [14,] -0.04423987
## [15,]  0.18564229
```

```
signif(acf(residuals(m1.color),plot=F)$acf[1:6],2)
```

```
## [1] -0.051  0.032  0.047  0.021 -0.017 -0.019
```

```
# display the first 6 acf values to 2 significant digits
```

The Ljung-Box and Box-Pierce version of the test are conducted with the following code K = 6:

```
Box.test(residuals(m1.color), lag = 6, type = "Box-Pierce", fitdf = 0)
```

```
##
##  Box-Pierce test
##
## data:  residuals(m1.color)
## X-squared = 0.24645, df = 6, p-value = 0.9997
```

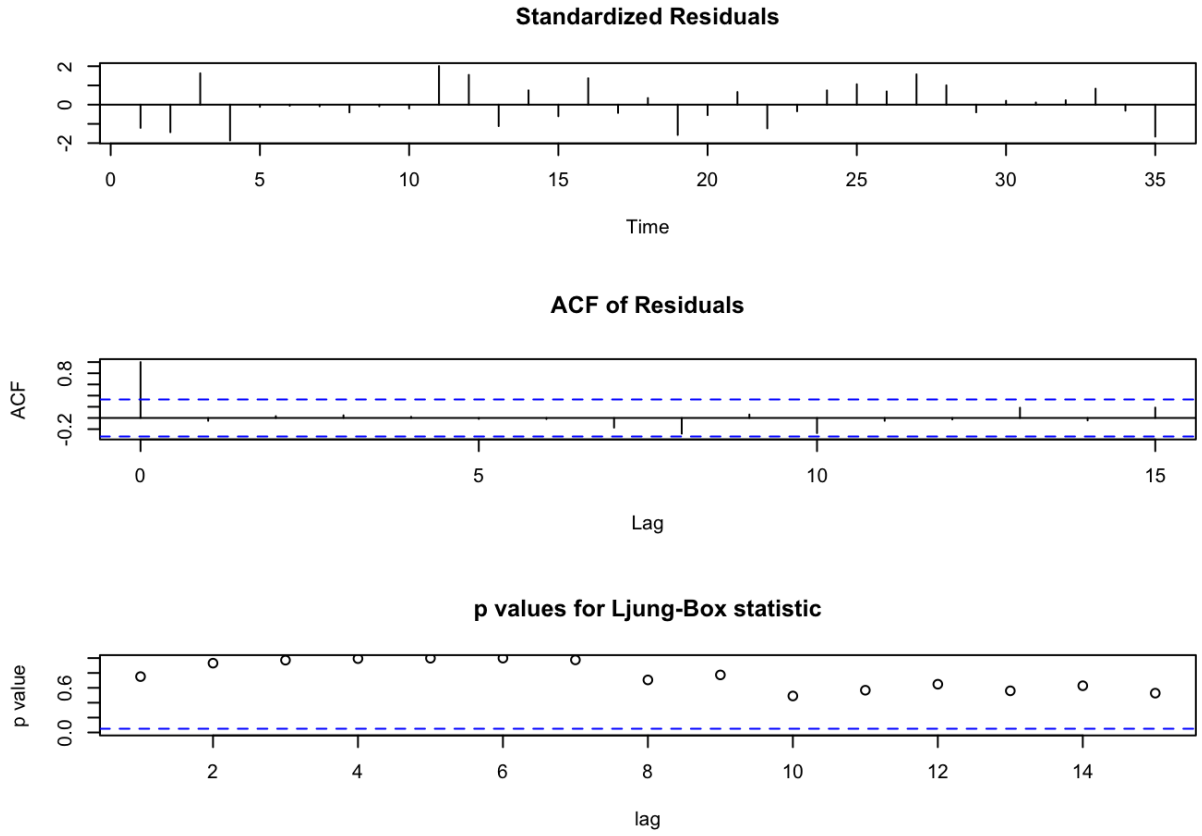
```
Box.test(residuals(m1.color), lag = 6, type = "Ljung-Box", fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  residuals(m1.color)
## X-squared = 0.28032, df = 6, p-value = 0.9996
```

There is not a notable difference between the results of Box-Pierce and Ljung-Box versions in terms of p-values, both of which are greater than 0.99. So, we have no evidence to reject the null hypothesis that the error terms are uncorrelated.

The next visualisation shows three diagnostic tools in one display: a sequence plot of the standardized residuals, the sample ACF of the residuals, and p-values for the Ljung-Box test statistic for a whole range of values of K from 5 to 15. The horizontal dashed line at 5% helps judge the size of the p-values.

```
tsdiag(m1.color,gof=15,omit.initial=F)
```



The estimated AR(1) model seems to be capturing the dependence structure of the color property time series quite well.

Overfitting and Parameter Redundancy

We can use overfitting as another tool to detect anomalies in terms of goodness of fit. After specifying and fitting what we believe to be an adequate model, we fit a slightly more general model; that is, a model "close by" that contains the original model as a special case. For example, if an AR(2) model seems appropriate, we might overfit with an AR(3) model. The original AR(2) model would be confirmed if:

1. the estimate of the additional parameter, ϕ_3, is not significantly different from zero, and
2. the estimates for the parameters in common, ϕ_1 and ϕ_2, do not change significantly from their original estimates.

Let's consider the industrial color property time series. We fitted an AR(1) model to this series. Now, to ensure overfitting, we fit AR(2) to this series and compare the results with that of AR(1).

```
arima(color,order=c(1,0,0))
```

```
##
## Call:
## arima(x = color, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.5705    74.3293
## s.e.  0.1435     1.9151
##
## sigma^2 estimated as 24.83:  log likelihood = -106.07,  aic = 216.15
```

```
arima(color,order=c(2,0,0))
```

```
##
## Call:
## arima(x = color, order = c(2, 0, 0))
##
## Coefficients:
##          ar1     ar2  intercept
##       0.5173  0.1005    74.1551
## s.e.  0.1717  0.1815     2.1463
##
## sigma^2 estimated as 24.6:  log likelihood = -105.92,  aic = 217.84
```

Because

- the parameter ϕ_2 is not statistically different from zero in the AR(2) fitting,
- the two estimates of ϕ_2 are quite close in both AR(1) and AR(2) models, and
- while the AR(2) model has a slightly larger log-likelihood value, the AR(1) fit has a smaller AIC value. The penalty for fitting the more complex AR(2) model is sufficient to choose the simpler AR(1) model,

we think that AR(1) fits very well to this color property series. A different overfit to this series would be ARMA(1,1). The results of this fit are shown below:

```
arima(color,order=c(1,0,1))
```

```
##
## Call:
## arima(x = color, order = c(1, 0, 1))
##
## Coefficients:
##          ar1      ma1  intercept
##       0.6721  -0.1467    74.1730
## s.e.  0.2147   0.2742     2.1357
##
## sigma^2 estimated as 24.63:  log likelihood = -105.94,  aic = 217.88
```

The standard errors of the estimated coefficients for this fit are rather larger than what we see for AR(1) and AR(2) models. The estimate of ϕ_1 from this fit is not significantly different from the estimate in the AR(1) and AR(2) models. Furthermore, the estimate of the new parameter, θ, is not significantly different from zero. This adds further support to the AR(1) model.

When generalizing ARMA models, we must be aware of the problem of parameter **redundancy** or **lack of identifiability**. To clarify this, let's consider ARMA(1,2) model as given below:

$$Y_t = \phi Y_{t-1} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2}.$$

Now replace t by $t-1$ to obtain

$$Y_{t-1} = \phi Y_{t-2} + e_{t-1} - \theta_1 e_{t-2} - \theta_2 e_{t-3}.$$

If we multiply both sides of this equation by any constant c and then subtract it from the other equation, we obtain

$$Y_t - (\phi + c)Y_{t-1} + \phi c Y_{t-2} = e_t - (\theta_1 + c)e_{t-1} - (\theta_2 - \theta_1 c)e_{t-2} + c\theta_2 e_{t-3}$$

This apparently defines an ARMA(2,3) process. But notice that we have the factorizations

$$1 - (\phi + c)x + \phi c x^2 = (1 - \phi x)(1 - cx)$$

and

$$1 - (\theta_1 + c)x + (\theta_2 - c\theta_1)x^2 + c\theta_2 x^3 = (1 - \theta_1 x - \theta_2 x^2)(1 - cx).$$

Thus the AR and MA characteristic polynomials in the ARMA(2,3) process have a common factor of $(1 - cx)$. Even though Y_t does satisfy the ARMA(2,3) model, clearly the parameters in that model are not unique which means the constant c is completely arbitrary. We say that we have parameter redundancy in the ARMA(2,3) model.

The implications for fitting and overfitting models are as follows:

1. Specify the original model carefully. If a simple model seems at all promising, check it out before trying a more complicated model.
2. When overfitting, do not increase the orders of both the AR and MA parts of the model simultaneously.
3. Extend the model in directions suggested by the analysis of the residuals. For example, if after fitting an MA(1) model, the substantial correlation remains at lag 2 in the residuals, try an MA(2), not an ARMA(1,1).

For example, consider the color property series. We have seen that an AR(1) model fits quite well. Suppose we try an ARMA(2,1) model. The results of this fit are shown below.

```
arima(color,order=c(2,0,1))
```

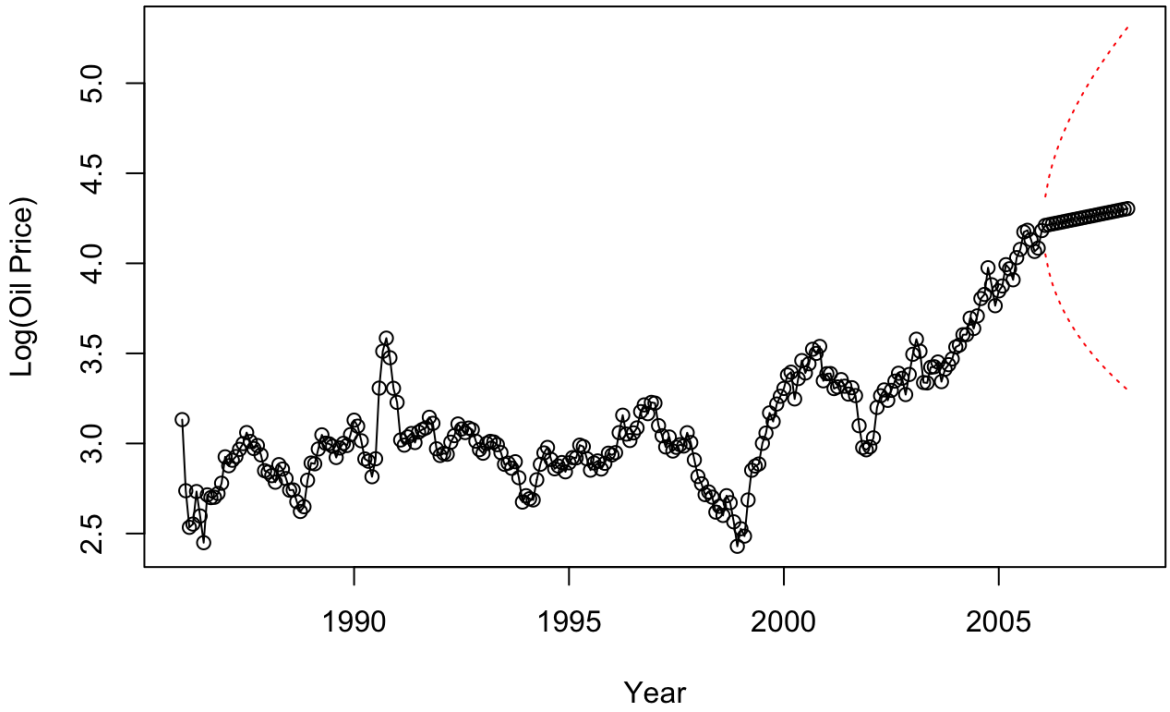
```
##
## Call:
## arima(x = color, order = c(2, 0, 1))
##
## Coefficients:
##          ar1     ar2     ma1  intercept
##       0.2189  0.2735  0.3036    74.1653
## s.e.  2.0056  1.1376  2.0650     2.1121
##
## sigma^2 estimated as 24.58:  log likelihood = -105.91,  aic = 219.82
```

Even though the estimate of σ_e^2 and the log-likelihood and AIC values are not too far from their best values, the estimates of ϕ_1, ϕ_2, and θ are way off, and none would be considered different from zero statistically.

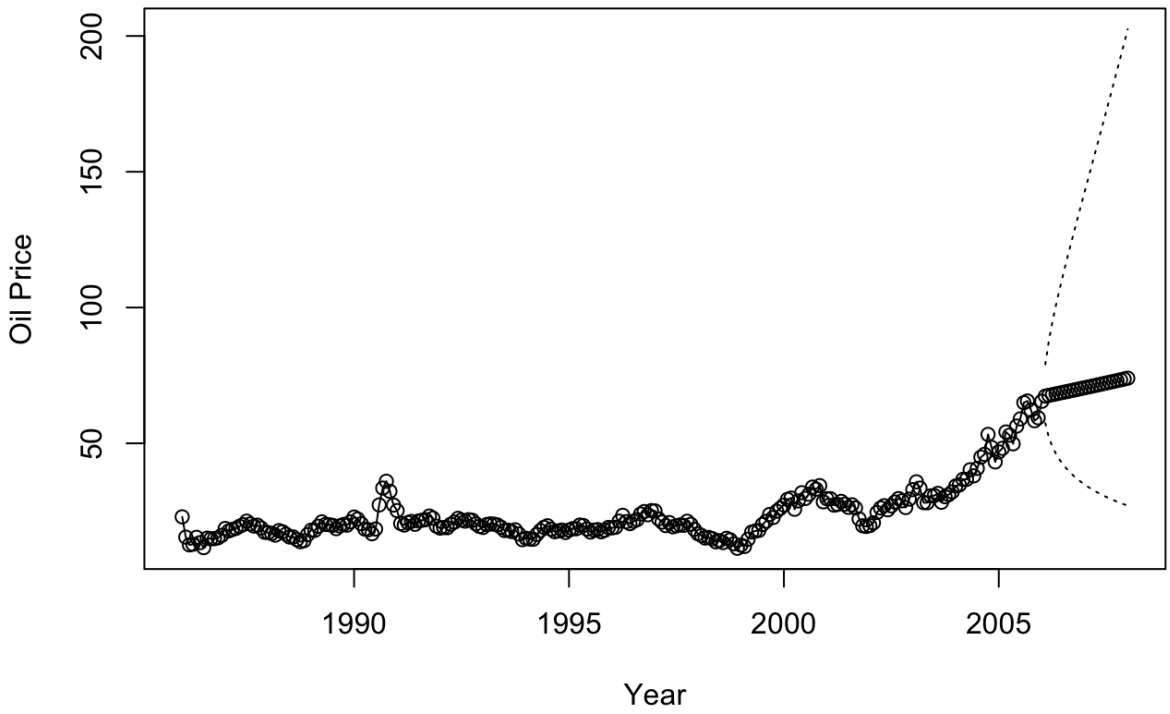
Forecasting with ARIMA models

After specifying the best model for the time series data, we can use the model to find forecasts. The `plot.Arima()` function can plot the time series data and its predictions with 95% prediction bounds. To illustrate, let's use the oil price series we modelled earlier and forecast 24 months ahead:

```
data(oil.price) # Load the data
m1.oil=arima(log(oil.price),order=c(0,1,1), #Fit ARIMA(0,1,1)
xreg=data.frame (constant=seq(oil.price))) # Create matrix of the covar
  iates
n=length(oil.price)
n.ahead=24 #Forecast 24 months ahead (default is 12)
newxreg=data.frame(constant=(n+1):(n+n.ahead)) #Create matrix of covari
  ates over the period of prediction
#Plot Log transformed series
plot(m1.oil, n.ahead=n.ahead, newxreg=newxreg,
     ylab='Log(Oil Price)',xlab='Year',n1=c(1986,1), col='red', type=
  'o')
```

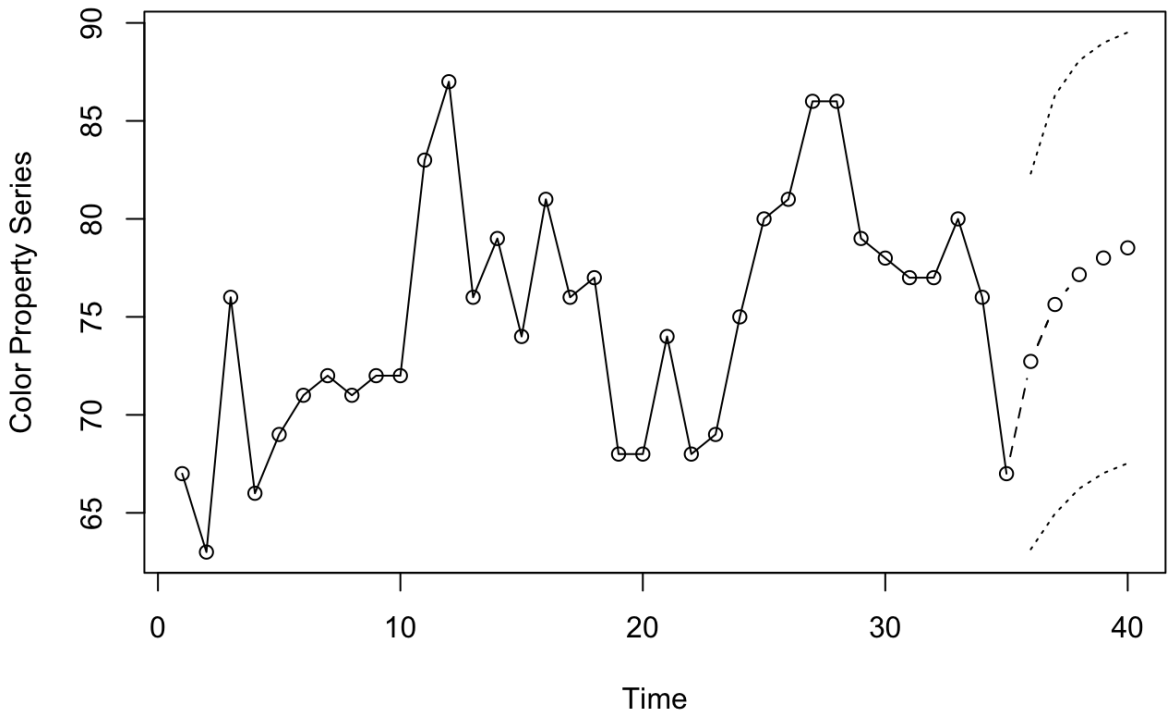



```
# Transform back to original scale (transform=exp)
plot(m1.oil,n.ahead=n.ahead,newxreg=newxreg,
     ylab='Oil Price',xlab='Year',n1=c(1986,1),transform=exp)
```



We can use the same method to forecast 5 steps ahead for the color property series:

```
data(color) # Load the data
m1.color=arima(color,order=c(1,0,0),  #Fit AR(1) Model
# create the design matrix of the covariate for prediction
xreg=data.frame (constant=seq(color)))
n=length(color)
n.ahead=5 #Forecast 5 steps ahead
newxreg=data.frame(constant=(n+1):(n+n.ahead))
# Plot the Forecast
plot(m1.color,n.ahead=n.ahead,newxreg=newxreg,
           ylab='Color Property Series',xlab='Time')
```



Summary

We focused on the analysis of residuals in this module. We examined various plots of the residuals, checking the error terms for constant variance, normality, and independence.

The properties of the sample autocorrelation of the residuals play a significant role in these diagnostics. The Ljung-Box statistic portmanteau test was discussed as a summary of the autocorrelation in the residuals.

Lastly, the ideas of overfitting and parameter redundancy were presented.