Syllabus.

- Chapter 1. Concepts in time series.
- Chapter 2. Univariate ARIMA models.
- Chapter 3. Model fitting and checking.
- Chapter 4. Prediction and model selection.
- Chapter 5. Outliers and influential observations.
- Chapter 6. Heterocedastic models.
- Chapter 7. Multivariate time series.

Chapter 3. Model fitting and checking.

- 3.1.Parameter estimation and diagnosis for AR(p) models.
- 3.2.Parameter estimation and diagnosis for MA(q) models
- 3.3.Parameter estimation and diagnosis for ARMA(p,q) Models.

Parameter estimation for an AR(p)

Observed time series are rarely centered. Then, it is inappropiate to fit a pure AR(p) process. All R routines by default assume the shifted process $Y_t = c + X_t$. Thus, we face the problem:

$$(Y_t - c) = \phi_1(Y_{t-1} - c) + \ldots + \phi_p(Y_{t-p} - c) + a_t$$

The goal is to estimate the global mean c, the AR-coeffcients $\phi_1, \ldots, \phi_p$, and in case we assume Gaussian white noise, the variance σ_a^2.

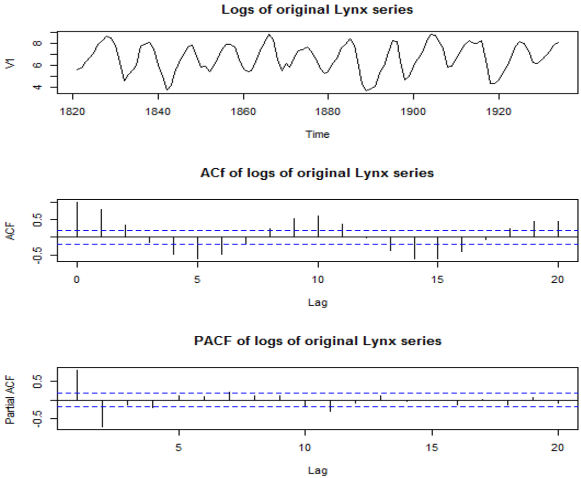
Parameter estimation for an AR(p)

We will discuss four different methods:

- Ordinary Least Squares estimation (OLS).
- Burg's algorithm.
- Yule-Walker equations.
- Maximum Likelihood Estimation (MLE)

Parameter estimation for an AR(p)

We will use the Lynx series and assume an ARIMA(2,0,0) on the logarithm transformation. So we need estimations of ϕ_1, ϕ_2, c and σ_a^2.



Parameter estimation for an AR(p):OLS estimation.

If we rethink the previously stated problem:

$$(Y_t - c) = \phi_1(Y_{t-1} - c) + \ldots + \phi_p(Y_{t-p} - c) + a_t$$

we recognize a multiple linear regression problem without intercept on the centered observations. What we need then is:

- Estimate $\hat{c} = \frac{1}{n}\sum_{t=1}^{n} y_t$ and determine $x_t = y_t - \hat{c}$
- Run a regression without intercept on x_t to obtain $\hat{\phi}_1, \ldots, \hat{\phi}_p$
- For $\hat{\sigma_a^2}$ take the residual standard error from the output.

Parameter estimation for an AR(p): OLS estimation.

For our lynx example,this can be done in R as:

```
> y<-llc[3:114]
> y1<-llc[2:113]
> y2<-llc[1:112]
> fit.ols<-lm(y ~ -1 + y1 + y2)
> summary(fit.ols)

Call:
lm(formula = y ~ -1 + y1 + y2)

Residuals:
     Min       1Q   Median       3Q      Max
-1.34040 -0.30418  0.06881  0.34246  1.18991

Coefficients:
   Estimate Std. Error t value Pr(>|t|)
y1  1.38435    0.06359   21.77   <2e-16 ***
y2 -0.74793    0.06364  -11.75   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.528 on 110 degrees of freedom
Multiple R-squared:  0.8341,    Adjusted R-squared:  0.8311
F-statistic: 276.5 on 2 and 110 DF,  p-value: < 2.2e-16

>
```

Parameter estimation for an AR(p):OLS estimation.

In R we have the function *ar.ols()* which is less cumbersome and time consuming that the procedure before:

```
> f.ar.ols<-ar.ols(log(lynx),aic=F,intercept=F,order=2)
> f.ar.ols

Call:
ar.ols(x = log(lynx), aic = F, order.max = 2, intercept = F)

Coefficients:
      1        2
 1.3844  -0.7479

Order selected 2  sigma^2 estimated as  0.2738
> f.ar.ols$x.mean
      V1
6.685933
>
```

The parameter c is automatically estimated and, therefore *intercept=F* and *aic=F*, *order=2* specifies a fixed $p = 2$.

Parameter estimation for an AR(p): Burg's algorithm.

While OLS works,the first p observations are never evaluated as responses. This is solved by Burg's algorithm, which uses the property of time-reversal in stochastic processes. We thus evaluate the Residual Sum Square (RSS) of forward and backward prediction errors:

$$\sum_{t=p+1}^{n}\left\{\left(X_t-\sum_{k=1}^{p}\phi_k X_{t-k}\right)^2+\left(X_{t-p}-\sum_{k=1}^{p}\phi_k X_{t-p+k}\right)^2\right\}$$

In contrast to OLS, there is no explicit solution and numerical optimization is required. This can be done with a recursive method. R uses the Durbin-Levinson algorithm.

Parameter estimation for an AR(p): Burg's algorithm.

```
> f.burg<-ar.burg(log(lynx),aic=F,order.max=2)
> f.burg

Call:
ar.burg.default(x = log(lynx), aic = F, order.max = 2)

Coefficients:
      1          2
 1.3831   -0.7461

Order selected 2  sigma^2 estimated as  0.2707
> f.burg$x.mean
[1] 6.685933
> sum(na.omit(f.burg$resid)^2/112)
[1] 0.2737614
>
```

Parameter estimation for AR(p). Yule Walker equations.

The Yule-Walker equations yield a Linear system of equations that connect the true ACF with the true AR-parameters. We plug-in the estimated ACF coefficients:

$$\hat{\rho}(k) = \hat{\phi}_1 \hat{\rho}(k-1) + \ldots + \hat{\phi}_p \hat{\rho}(k-p) \quad \text{\textit{for}} \quad k = 1, \ldots, p$$

and can solve the system to obtain the AR-parameter estimates.

Parameter estimation for AR(p). Yule Walker equations.

Notice that,because it is obtained from the fitted coefficients via the autocovariance, $\hat{\sigma}_a^2$ is different in this case.
While the Yule-Walker method is asymptotically equivalent to OLS and Burg's algorithm, it generally yields a solution with worse Gaussian likelihood on finite samples.

```
> ar.yw(x=log(lynx),aic=F,order.max=2)

Call:
ar.yw.default(x = log(lynx), aic = F, order.max = 2)

Coefficients:
      1        2
 1.3504  -0.7200

Order selected 2  sigma^2 estimated as  0.3109
> c<-mean(log(lynx))
> c
[1] 6.685933
```

Parameter estimation for AR(p). Maximum likelihood estimation.

The MLE is based on determining the model coefficients such that the likelihood given the data is maximized, i.e. the density function takes its maximal value under the present observations.
This requires the choice of a probability model for the time series. By assuming Gaussian innovations, $a_t \sim N(0, \sigma_a^2)$, any AR(p) process has a multivariate normal distribution:

$$Y = (Y_1, \ldots, Y_n) \sim N(c \cdot i_n, V)$$

with V depending on $\phi_1, \ldots, \phi_p$ and σ_a^2. MLE then provides simultaneous estimates by optimizing (some constants being omitted):

$$L(\phi_1, \ldots, \phi_p, c, \sigma_a^2) \propto exp\left(\sum_{t=1}^{n}(x_t - \hat{x}_t)^2\right)$$

Parameter estimation for AR(p). Maximum likelihood estimation.

While MLE by default assumes Gaussian innovations, it still performs reasonably for other distributions as long as they are not extremly skewed or have big outliers.

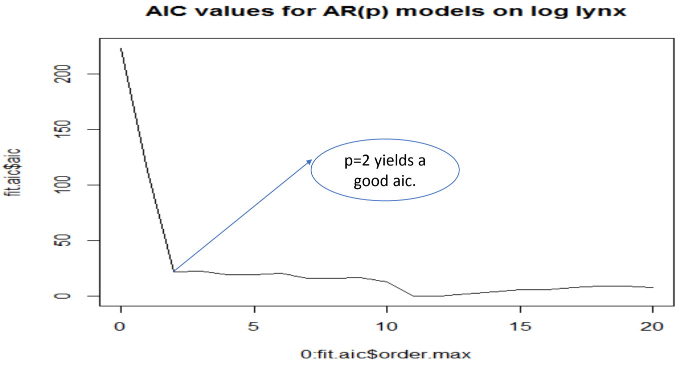
```
> arima(x=log(lynx),order=c(2,0,0))

Call:
arima(x = log(lynx), order = c(2, 0, 0))

Coefficients:
         ar1      ar2  intercept
      1.3776  -0.7399     6.6863
s.e.  0.0614   0.0612     0.1349

sigma^2 estimated as 0.2708:  log likelihood = -88.58,  aic = 185.15
>
```

Practical aspects on the estimation of AR(p)

- All four estimation methods are asymptotically equivalent and even on finite samples, the differences are usually small.
- All four methods are non-robust against outliers and perform best on data that are approximately gaussian.
- Function *arima ()* provides standard errors for the parameters, so that statements about significance become feasible and confidence intervals for the parameters can be built.
- *ar.ols(), ar.yw() and ar.burg()* allow for convenient choice of the optimal model order p using the AIC criterion. Among these methods, *ar.burg()* seems to perfom better.

AIC-based order choice

```
$g(lynx))
$r.max,fit.aic$aic, type="l", main="AIC values for AR(p) models on log lynx")
```



Summary of the four methods for lynx.

	Phi1	Phi2	Sigma^2
OLS	1.3844	-0.7479	0.2738
BURG	1.3831	-0.7461	0.2737
YW	1.3504	-0.7200	0.3109
MLE	1.3776	-0.7399	0.2708

Model diagnosis.

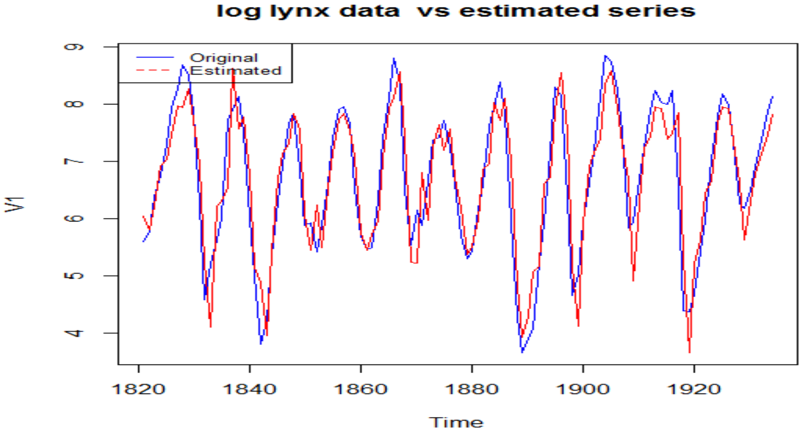
What we do is residual analysis: *residuals=estimated innovations* $= \hat{a}_t$ where,

$$\hat{a}_t = (y_t - \hat{c}) - (\hat{\phi}_1(y_{t-1} - \hat{c}) \ldots - \hat{\phi}_p(y_{t-p} - \hat{c}))$$

Remember the assumptions we made:

$$a_t \sim Niid(0, \sigma_a^2)$$

Model diagnosis. Plotting the fit

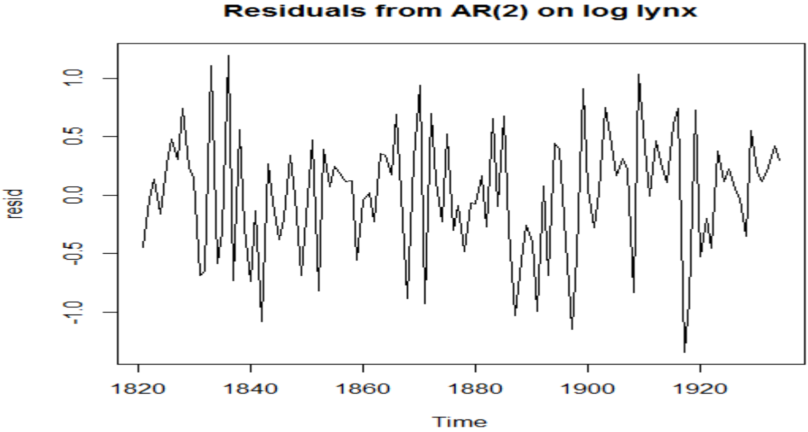


Model diagnosis. Residual analysis

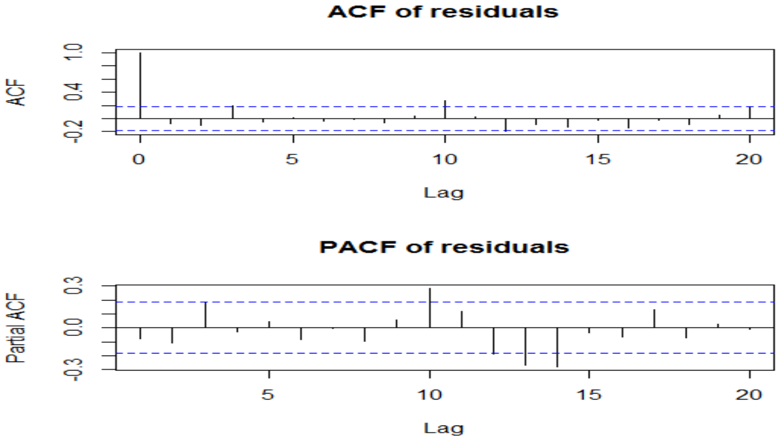
We check the assumptions we made with the following plots:

- Time series plot of $\hat{a}_t$
- ACF/PACF plot of $\hat{a}_t$ (could use Box-Ljung test)
- QQ-plot of $\hat{a}_t$ (could use Jarque-Bera test on normality.)

Model diagnosis. Residual analysis.

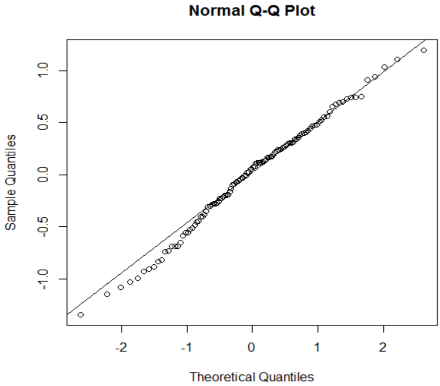


Model diagnosis. Residual analysis.



Ljung-Box test (36)=53.706; p-value=0.0291 $\longrightarrow$ Rejection of the null.

Model diagnosis. Residual analysis.



Jarque-bera test =1.3746; p-value=0.5029 → No Rejection of the null.

Parameter estimation for MA(q).

The simplest idea is to exploit the relation between model parameters and autocorrelation coefficients ("Yule- Walker") after the global mean c has been estimated and substracted:

$$\rho(k) = \left\{ \begin{array}{lll} \sum_{j=0}^{q-k} \theta_j \theta_{j+k} / \sum_{j=0}^{q} \theta_j^2 & for & k = 1, ...q \\ 0 & for & k > q \end{array} \right\}$$

In contrast to the Yule-Walker method for AR(p) models, this yields an ineficient estimator that generally generates poor results and hence should not be used in practice.

Parameter estimation.Conditional Sum of Squares.

Based on the idea of expressing $\sum a_t^2$ in terms of $X_1, \ldots, X_n$ and $\theta_1, \ldots \theta_q$, as the innovations are unobservable. This is possible for any invertible MA(q) process. For example, for the MA(1) process using the AR(∞) representation:

$$a_t = X_t - \theta_1 X_{t-1} + \theta_1^2 X_{t-2} + \ldots + (-\theta_1)^{t-1} X_1 + \theta_1^t a_0$$

Conditonal on the assumption of $a_0 = 0$, it is possible to rewrite $\sum a_t^2$ for any MA(1) using just $X_1, \ldots, X_n$ and θ_1. Numerical optimization is required for finding the optimal parameter θ_1, but it is available in R function *arima ()*.

Parameter estimation.Maximum likelihood estimation.

- In practice, it is preferable to use CCS only to obtain a first estimate of the coefficients $\theta, \ldots, \theta_q$ and use them as starting values for a Maximum Likelihood Estimation (MLE) based on the assumption of Gaussian innovations.
- The linear combination of Gaussians, $X_t = a_t + \theta_1 a_{t-1} + \ldots + \theta_q a_{t-q}$, follows a Gaussian too. By taking the covariance terms into account, we obtain a multivariate Gaussian for the time series vector:

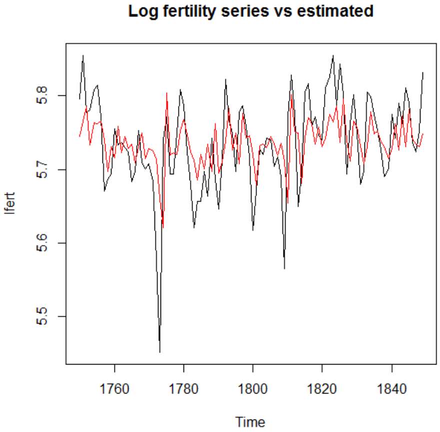
$$X = (X_1, \ldots, X_n) \sim N(0, V), \quad , X_t = Y_t - c$$

MLE then relies on determining the parameters simultaneously by maximizing the probability density function of the above multivariate Gaussian assuming the data $x_1, \ldots, x_n$ as given quantities. That is a non-linear problem which need to be solved numerically.

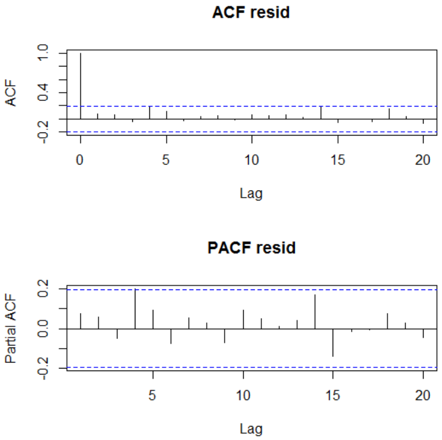
Example of Swedish fertility.

```
> lfert<-log(ts(read.table(file="clipboard"),start=1750,freq=1))
> f<-arima(lfert,order=c(0,0,1),method="CSS-ML")
> f

Call:
arima(x = lfert, order = c(0, 0, 1), method = "CSS-ML")

Coefficients:
         ma1  intercept
      0.5592     5.7367
s.e.  0.0734     0.0085

sigma^2 estimated as 0.002994:  log likelihood = 148.48,  aic = -290.95
> resid<-f$resid
> yest=lfert-resid
> lines(yest,col="red")
> par(mfrow=c(2,1))
> acf(resid,main="ACF resid")
> pacf(resid,main="PACF resid")
> Box.test(resid)

        Box-Pierce test

data:  resid
X-squared = 0.57822, df = 1, p-value = 0.447

> par(mfrow=c(1,1))
> qqnorm(resid)
> qqline(resid)
> jarque.bera.test(resid)

        Jarque Bera Test

data:  resid
X-squared = 16.696, df = 2, p-value = 0.0002368

>
```

Example of Swedish fertility.

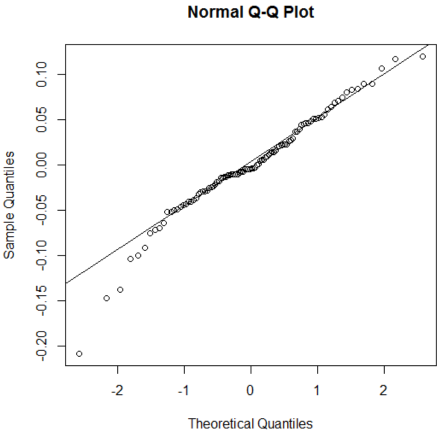


Example of Swedish fertility.



Ljung Box test =0.5782; p-value=0.447 → No Rejection of the null.

Example of Swedish fertility.



Jarque-bera test =16.696; p-value=0.0002 → Rejection of the null.

Parameter estimation for ARMA(p,q).Conditional Sum of Squares.

- Based on the idea of expressing $\sum a_t^2$ in terms of $X_1, \ldots, X_n$ and $\phi_1, \ldots, \phi_p, \theta_1, \ldots \theta_q$, as the innovations are unobservable. This is possible for any stationary and invertible ARMA(q) process.
- Conditional on the assumption of:

$$a_0 = a_{-1} = a_{-2} \ldots = 0,$$

  it is possible to rewrite $\sum a_t^2$ for any ARMA(p,q) using just $X_1, \ldots, X_n$ and $\phi_1, \ldots, \phi_p, \theta_1, \ldots \theta_q$.
- Numerical optimization is required for finding the optimal parameters, but it is available in R function *arima ()*.

Parameter estimation.Maximum likelihood estimation.

- In practice, it is preferable to use CCS only to obtain a first estimate of the coefficients and use them as starting values for a Maximum Likelihood Estimation (MLE) based on the assumption of Gaussian innovations.
- The linear combination of Gaussians,

$$X_t = \phi_1 X_{t-1} + \ldots + \phi_p X_{t-p} + a_t + \theta_1 a_{t-1} + \ldots + \theta_q a_{t-q}$$

follows a Gaussian too. By taking the covariance terms into account, we obtain a multivariate Gaussian for the time series vector:

$$X = (X_1, \ldots, X_n) \sim N(0, V), \quad , X_t = Y_t - c$$

MLE then relies on determining the parameters simultaneously by maximizing the probability density function of the above multivariate Gaussian assuming the data $x_1, \ldots, x_n$ as given quantities.

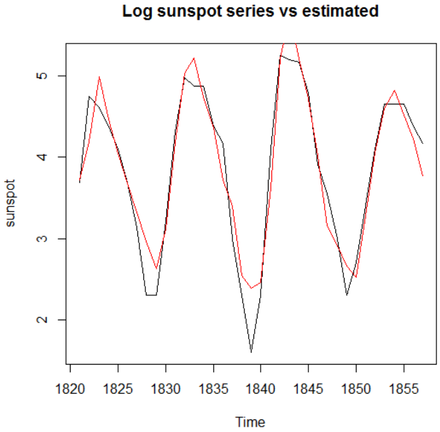
Maximum likelihood estimation remarks.

- The MLE approach would work for any distribution. However, for innovation distributions other than the gaussian, the joint distribution might be difficult to obtain.
- For reasonable deviations from the normality assumption, MLE mostly yields good results. Big outliers could be a problem.
- Besides the parameter estimates, we also obtain their standar errors, assessing the precision of the estimates.
- Tramo performs MLE with Hannan- Risannen algorithm to solve the non-linearity.

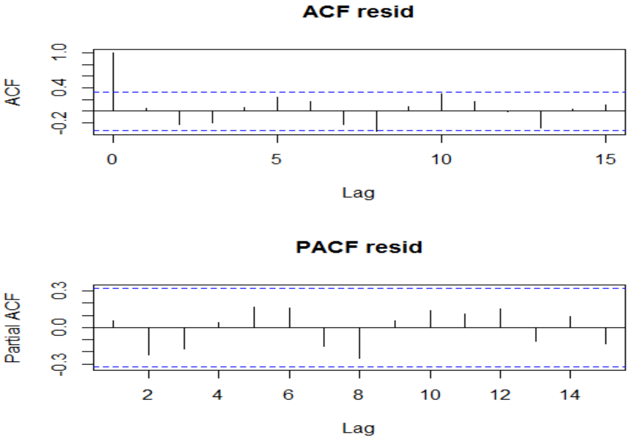
Example of Sunspots.

```
> sunspot<-log(ts(read.table(file="clipboard"),start=1821,freq=1))
> library(forecast)
> auto.arima(sunspot)
Series: sunspot
ARIMA(0,0,3) with non-zero mean

Coefficients:
         ma1     ma2     ma3    mean
      1.3245  1.0019  0.4109  3.8248
s.e.  0.1651  0.1664  0.1607  0.2666

sigma^2 estimated as 0.2249:  log likelihood=-23.76
AIC=57.53   AICc=59.46   BIC=65.58
> f<-arima(sunspot,order=c(2,0,1),method="CSS-ML")
> f

Call:
arima(x = sunspot, order = c(2, 0, 1), method = "CSS-ML")

Coefficients:
         ar1      ar2      ma1  intercept
      1.5865  -0.9310  -1.0000     3.7846
s.e.  0.0462   0.0433   0.1095     0.0140

sigma^2 estimated as 0.09192:  log likelihood = -11.65,  aic = 33.31
>
```

Example of Sunspots.

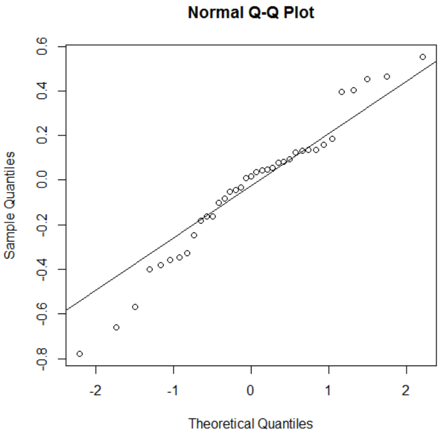


Example of Sunspots.



Ljung Box test =0.1158; p-value=0.7336 → No Rejection of the null.

Example of Sunspots.



Jarque-bera test =0.7497; p-value=0.6878 → No Rejection of the null.