universität innsbruck

Time Series Analysis

Model Specification

Model Specification

Contents

Contents

Contents

Goal: Statistical inference for ARIMA models.

Strategy: Box and Jenkins (1976). Iterate the following steps.

1. Choose appropriate values for p, d, and q for a given series (Chapter 6).
2. Estimate parameters (ϕ, θ, σ_e^2, μ) of a specific ARIMA(p, d, q) model for a given time series (Chapter 7).
3. Check appropriateness of the fitted model (Chapter 8).

Model Specification

Properties of the Sample Autocorrelation Function

Sample autocorrelation function

Recall: For $k = 1, 2, \ldots$, the *sample autocorrelation function* is

$$r_k \;=\; \frac{\sum_{t=k+1}^{n}(Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^{n}(Y_t - \bar{Y})^2}.$$

Goal: Recognize patterns in r_k that are characteristic of known patterns in ϱ_k for ARMA models.

Example: $\varrho_k = 0$ for $k > q$ in MA(q) models.

Needed: Sampling properties of r_k.

Note: r_k is ratio of two quadratic forms. Thus, even the mean is not trivial to obtain.

Sample autocorrelation function

Suppose:

$$Y_t = \mu + \sum_{j=0}^{\infty} \psi_j e_{t-j}$$

with e_t i.i.d. and

$$\sum_{j=0}^{\infty} |\psi_j| < \infty$$

$$\sum_{j=0}^{\infty} j\psi_j^2 < \infty$$

which is satisfied for all stationary ARMA models.

Sample autocorrelation function

Then: For any fixed m

$$\sqrt{n}\begin{pmatrix} r_1 - \varrho_1 \\ \vdots \\ r_m - \varrho_m \end{pmatrix} \xrightarrow{d} \mathcal{N}(0, C)$$

where $C = (c_{ij})_{i,j=1,\dots,m}$ with

$$c_{ij} = \sum_{k=-\infty}^{\infty} (\varrho_{k+i}\varrho_{k+j} + \varrho_{k-i}\varrho_{k+j} - 2\varrho_i\varrho_k\varrho_{k+j} - 2\varrho_j\varrho_k\varrho_{k+i} + 2\varrho_i\varrho_j\varrho_k^2).$$

Thus: For large n, r_k is approximately normally distributed with mean ϱ_k and variance c_{kk}/n. Furthermore, $\mathrm{Cor}(r_k, r_j) \approx c_{kj} \,/\sqrt{c_{kk}c_{jj}}$.

Sample autocorrelation function

Special cases: White noise.

$$\begin{aligned} \mathrm{Var}(r_k) &\approx \frac{1}{n}, \\ \mathrm{Cor}(r_k, r_j) &\approx 0. \end{aligned}$$

AR(1): $\varrho_k = \phi^k$.

$$\begin{aligned} \mathrm{Var}(r_k) &\approx \frac{1}{n}\left[\frac{(1+\phi^2)(1-\phi^{2k})}{1-\phi^2} - 2k\phi^{2k}\right], \\ \mathrm{Var}(r_1) &\approx \frac{1-\phi^2}{n}, \\ \mathrm{Var}(r_k) &\approx \frac{1}{n}\left[\frac{1+\phi^2}{1-\phi^2}\right] \quad \text{for large } k, \\ \mathrm{Cor}(r_k, r_j) &\approx 2\phi\sqrt{\frac{1-\phi^2}{1+2\phi^2-3\phi^4}}. \end{aligned}$$

Sample autocorrelation function

MA(1): $\varrho_1 = -\theta/(1+\theta^2)$.

$$
\begin{aligned}
c_{11} &= 1 - 3\varrho_1^2 + 4\varrho_1^4, \\
c_{kk} &= 1 + 2\varrho_1^2 \quad \text{for } k > 1, \\
c_{12} &= 2\varrho_1(1-\varrho_1^2).
\end{aligned}
$$

MA(q): For $k > q$.

$$
\mathrm{Var}(r_k) \approx \frac{1}{n}\left[1 + 2\sum_{j=1}^{q} \varrho_j^2\right].
$$

Sample autocorrelation function

Example: AR(1).

ϕ	$\sqrt{n \cdot \mathrm{Var}(r_1)}$	$\sqrt{n \cdot \mathrm{Var}(r_2)}$	$\sqrt{n \cdot \mathrm{Var}(r_{10})}$	$\mathrm{Cor}(r_1, r_2)$
± 0.9	0.44	0.81	2.44	± 0.97
± 0.7	0.71	1.12	1.70	± 0.89
± 0.4	0.92	1.11	1.18	± 0.66
± 0.2	0.98	1.04	1.04	± 0.38

Sample autocorrelation function

Example: MA(1).

θ	$\sqrt{n \cdot \mathrm{Var}(r_1)}$	$\sqrt{n \cdot \mathrm{Var}(r_k)}$, $k > 1$	$\mathrm{Cor}(r_1, r_2)$
± 0.9	0.71	1.22	∓ 0.86
± 0.7	0.73	1.20	∓ 0.84
± 0.5	0.79	1.15	∓ 0.74
± 0.3	0.89	1.07	∓ 0.53

Model Specification

The Partial and Extended Autocorrelation Functions

Partial autocorrelation function

So far:

- ACF for MA(q) processes cuts off after lag q.
- Hence, ACF useful for order selection of MA processes.
- ACF for AR(p) tails off slowly but does not cut off.
- Hence, ACF does not help for order selection of AR processes.

Question: Can we find a function with similar properties for AR(p) processes?

Answer: Yes, consider partial autocorrelation of Y_t and Y_{t-k} after removing the effect of the intervening variables $Y_{t-1}, \ldots, Y_{t-k+1}$.

Partial autocorrelation function

Partial autocorrelation in normal series $\{Y_t\}$ can be written as

$$\phi_{kk} \;=\; \mathrm{Cor}(Y_t, Y_{t-k} \mid Y_{t-1}, \dots, Y_{t-k+1})$$

More generally, for stationary series, define *partial autocorrelation function (PACF)* at lag k as

$$\begin{aligned}\phi_{kk} \;=\; \mathrm{Cor}(&Y_t - \beta_1 Y_{t-1} - \cdots - \beta_{k-1} Y_{t-k+1},\\ &Y_{t-k} - \beta_1 Y_{t-k+1} - \cdots - \beta_{k-1} Y_{t-1})\end{aligned}$$

where β is chosen to minimize the mean prediction error.

Remarks:

- Due to stationarity, “prediction” goes forward and backward in time.
- $\phi_{11} = \varrho_1$.
- Double subscript will become clear later.

Partial autocorrelation function

Example: Computation of ϕ_{22}.

Clear: In stationary series (i.e., with constant variance) with zero mean, the best linear predictor for Y_t based on only Y_{t-1} is $\varrho_1 Y_{t-1}$.

Thus:

$$
\begin{aligned}
\mathrm{Cov}(Y_t - \varrho_1 Y_{t-1}, Y_{t-2} - \varrho_1 Y_{t-1}) &= \gamma_0(\varrho_2 - \varrho_1^2 - \varrho_1^2 + \varrho_1^2) \\
&= \gamma_0(\varrho_2 - \varrho_1^2) \\
\mathrm{Var}(Y_t - \varrho_1 Y_{t-1}) &= \mathrm{Var}(Y_{t-2} - \varrho_1 Y_{t-1}) \\
&= \gamma_0(1 + \varrho_1^2 - 2\varrho_1^2) \\
&= \gamma_0(1 - \varrho_1^2) \\
\mathrm{Cor}(Y_t - \varrho_1 Y_{t-1}, Y_{t-2} - \varrho_1 Y_{t-1}) &= \frac{\varrho_2 - \varrho_1^2}{1 - \varrho_1^2}
\end{aligned}
$$

Partial autocorrelation function

Example: In AR(1) processes, $\varrho_k = \phi^k$.

$$\phi_{22} \ = \ \frac{\phi^2 - \phi^2}{1 - \phi^2} \ = \ 0.$$

Thus, PACF is non-zero at lag 1 but zero at lag 2.

More generally: For AR(p) processes PACF $\phi_{kk} = 0$ for all $k > p$.

Best linear predictor for Y_t based on $Y_{t-1}, \ldots, Y_{t-k+1}$ with $k > p$ is $\phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p}$. Thus

$$\begin{aligned} & Cov(Y_t - \phi_1 Y_{t-1} - \cdots - \phi_p Y_{t-p}, \\ & \qquad Y_{t-k} - \phi_1 Y_{t-k+1} - \cdots - \phi_p Y_{t-k+p}) \\ = \ & \mathrm{Cov}(e_t, Y_{t-k} - \phi_1 Y_{t-k+1} - \cdots - \phi_p Y_{t-k+p}) \\ = \ & 0 \end{aligned}$$

Partial autocorrelation function

Example: MA(1).

$$\begin{aligned}\phi_{22} &= -\frac{\theta^2}{1+\theta^2+\theta^4} \\ \phi_{kk} &= -\frac{\theta^k(1-\theta^2)}{1-\theta^{2(k+1)}} \quad \text{for } k \geq 1\end{aligned}$$

i.e., PACF decays exponentially fast (similar to the ACF of an AR(1) process).

More generally: PACF of MA(q) process can be shown to behave similarly to ACF of AR(q) process.

Partial autocorrelation function

Goal: Compute PACF ϕ_{kk} from known ACF ϱ_k.

It can be shown that Yule-Walker type equations also hold for partial autocorrelation coefficients ϕ_{kj} with $j = 1, \ldots, k$:

$$\varrho_j = \phi_{k1}\varrho_{j-1} + \phi_{k2}\varrho_{j-2} + \ldots + \phi_{kk}\varrho_{j-k}$$

This yields k linear equations

$$\begin{aligned}
\varrho_1 &= \phi_{k1} + \phi_{k2}\varrho_1 + \phi_{k3}\varrho_2 + \ldots + \phi_{kk}\varrho_{k-1} \\
\varrho_2 &= \phi_{k1}\varrho_1 + \phi_{k2} + \phi_{k3}\varrho_1 + \ldots + \phi_{kk}\varrho_{k-2} \\
&\vdots \\
\varrho_k &= \phi_{k1}\varrho_{k-1} + \phi_{k2}\varrho_{k-2} + \phi_{k3}\varrho_{k-3} + \ldots + \phi_{kk}
\end{aligned}$$

that can be solved for $\phi_{k1}, \ldots, \phi_{kk}$ recursively for $k \geq 2$.

For AR(p), these yield the usual Yule-Walker equations for $k = p$ and thus $\phi_{pp} = \phi_p$.

Partial autocorrelation function

Furthermore: Recursion can be computed efficiently using Durbin-Levinson algorithm.

$$
\begin{aligned}
\phi_{k,k} &= \frac{\varrho_k - \sum_{j=1}^{k-1} \phi_{k-1,j}\, \varrho_{k-j}}{1 - \sum_{j=1}^{k-1} \phi_{k-1,j}\, \varrho_j} \\
\phi_{k,j} &= \phi_{k-1,j} - \phi_{k,k}\, \phi_{k-1,k-j} \quad \text{for } j = 1, \ldots, k-1
\end{aligned}
$$

Thus: Initialize with $\phi_{11} = \varrho_1$.

$$
\begin{aligned}
\phi_{22} &= \frac{\varrho_2 - \phi_{11}\varrho_1}{1 - \phi_{11}\varrho_1} = \frac{\varrho_2 - \varrho_1^2}{1 - \varrho_1^2} \\
\phi_{21} &= \phi_{11} - \phi_{22}\phi_{11} \\
\phi_{33} &= \frac{\varrho_3 - \phi_{21}\varrho_2 - \phi_{22}\varrho_1}{1 - \phi_{21}\varrho_1 - \phi_{22}\varrho_2} \\
\phi_{31} &= \ldots
\end{aligned}
$$

Partial autocorrelation function

Definition: The *sample partial autocorrelation function* $\hat{\phi}_{kk}$ is the plug-in estimate of the PACF which replaced the ACF ϱ_k by the sample ACF $\hat{\varrho}_k = r_k$.

Remarks:

- Under white noise hypothesis, $\hat{\phi}_{kk}$ is approximately normal with zero mean and standard deviation $1/\sqrt{n}$.
- Also under AR(p) hypothesis, the same holds for $k > p$.

In R: `pacf()`.

The extended autocorrelation function

Problem: Order selection for ARMA(p, q) with $p, q > 0$ more difficult.

	AR(p)	MA(q)	ARMA(p, q)
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

Solutions: Graphical tools.

- Extended autocorrelation function (EACF).
- Smallest canonical correlations.
- ...

Remark: Rarely used in practice. More commonly, information criteria are used. Details later.

The extended autocorrelation function

Idea:

- "Filter out" AR component of ARMA process.
- Yields MA process with the same cutoff property in ACF.

Example: ARMA(1, 1).

$$Y_t \;=\; \phi Y_{t-1} \;+\; e_t \;-\; \theta e_{t-1}.$$

- Linear regression of Y_t on Y_{t-1} is inconsistent as $\varrho_1 = (\phi - \theta)(1 - \phi\theta)/(1 - 2\phi\theta + \theta^2) \neq \phi$.
- But residuals from this regression contain information about $\{e_t\}$.
- Linear regression of Y_t on Y_{t-1} and residuals from first regression yields coefficient $\tilde{\phi}$ of Y_{t-1} consistent for ϕ.
- $W_t = Y_t - \tilde{\phi} Y_{t-1}$ is approximately MA(1).

The extended autocorrelation function

More generally:

- For ARMA(1, 2), regress Y_t on its lag 1, lag 1 of residuals from second regression, and lag 2 of residuals from the first regression.
- For ARMA(p, q), estimate regressions iteratively and investigate ACF of residuals at last stage.
- With orders p and q unknown, investigate up to certain maximal orders.

In R: `eacf()` in package *TSA*. (*Warning:* Redefines also `acf()` which contains some inconsistencies/bugs.)

Model Specification

Specification of Some Simulated Time Series

Specification of simulated time series

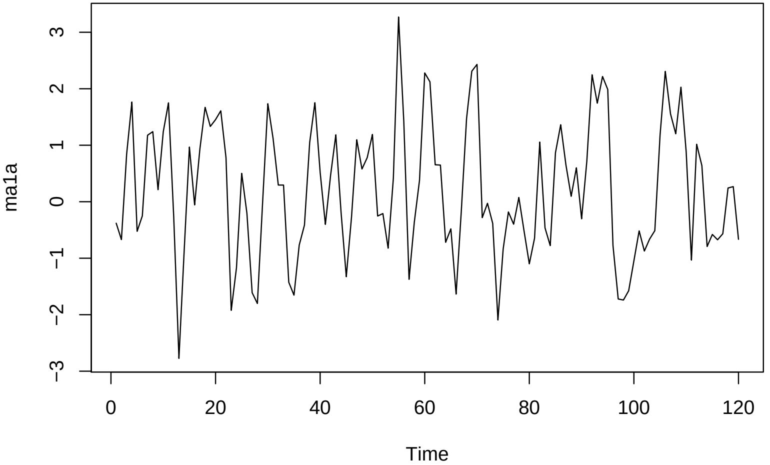
In practice:

- Visualize sample ACF and PACF for observed series.
- Add pointwise confidence bounds for white noise, i.e., $\pm 1.96/\sqrt{n}$ at 95% level.
- Alternatively, employ confidence bounds for MA($k-1$) process for ACF at lag k.
- Recall that confidence bounds for AR($k-1$) process for PACF at lag k coincide with white noise bounds.
- In R, `acf()` and `pacf()` yield plots as by-products. For `acf()`, `ci.type = "ma"` can be added.

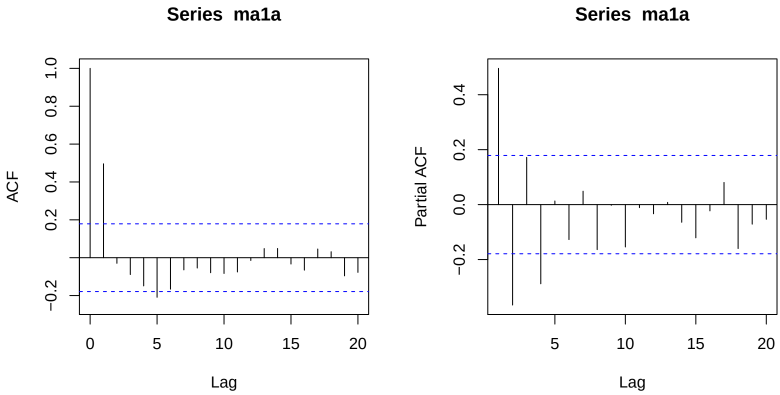
Illustration: Reconsider simulated series from Chapter 4.

Specification of simulated time series

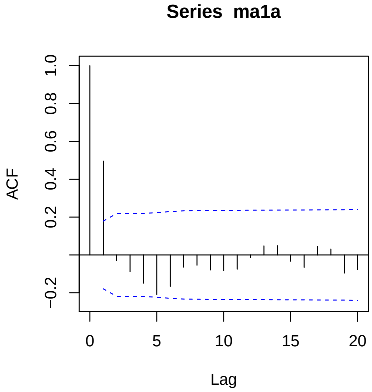
Illustration: MA(1) with $\theta = -0.9$.

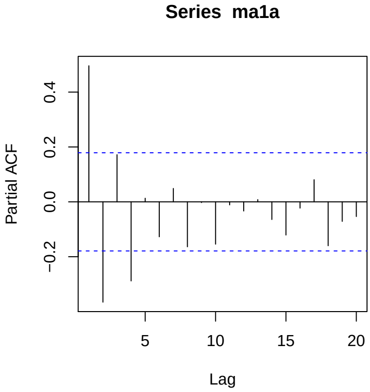


Specification of simulated time series



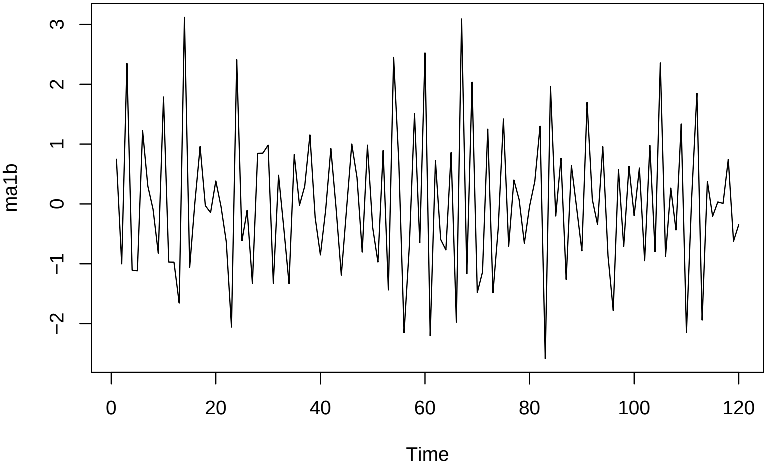
Specification of simulated time series



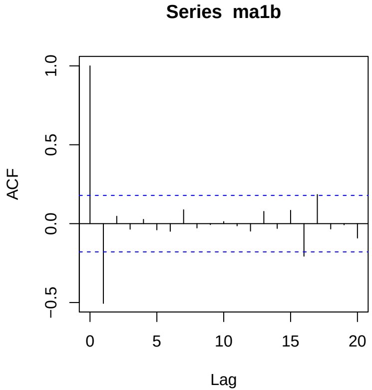


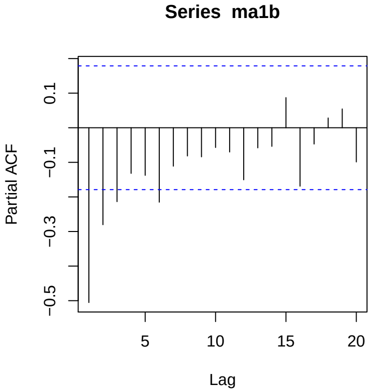
Specification of simulated time series

Illustration: MA(1) with $\theta = 0.9$.

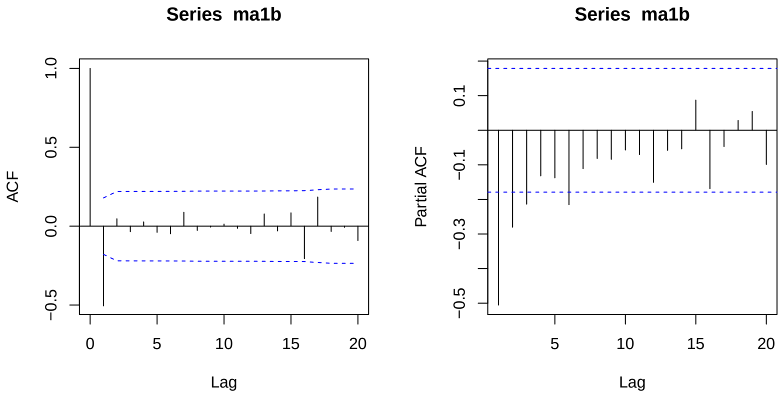


Specification of simulated time series



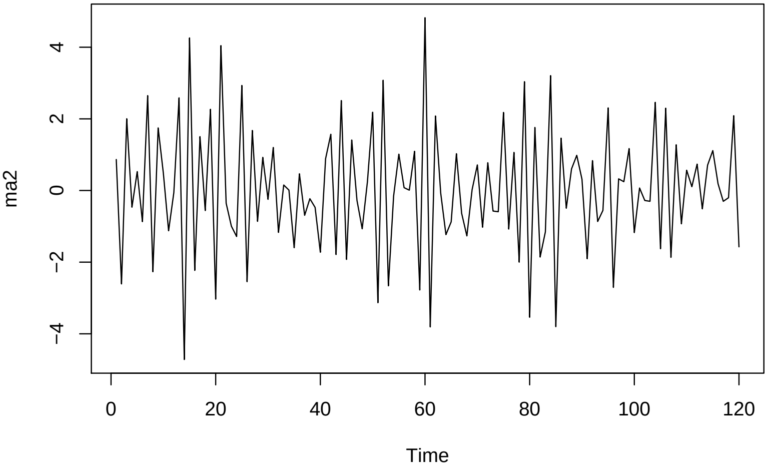


Specification of simulated time series

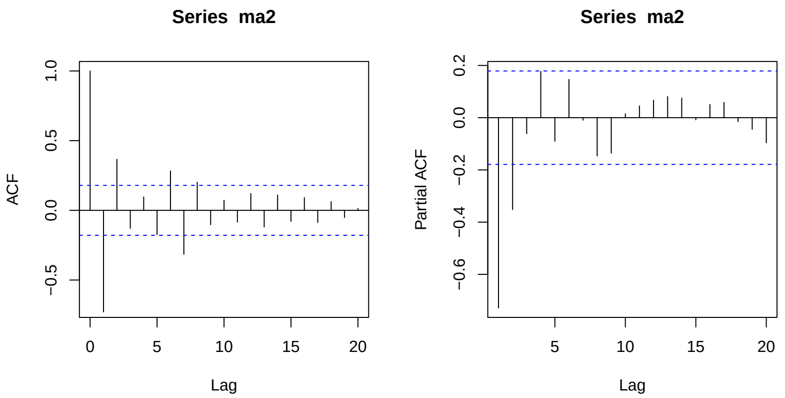


Specification of simulated time series

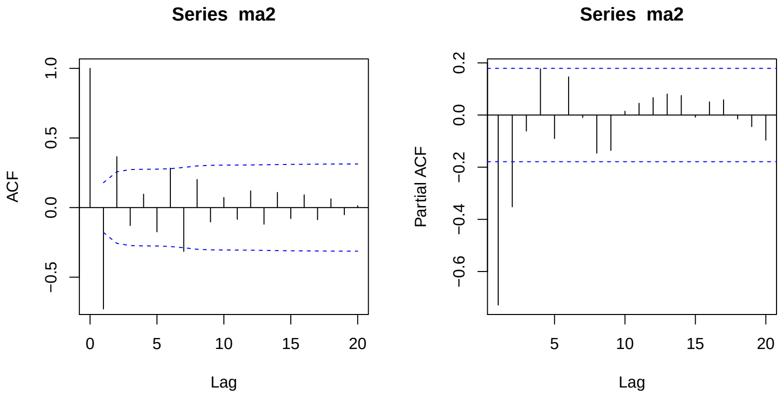
Illustration: MA(2) with $\theta_1 = 1$ and $\theta_2 = -0.6$.



Specification of simulated time series

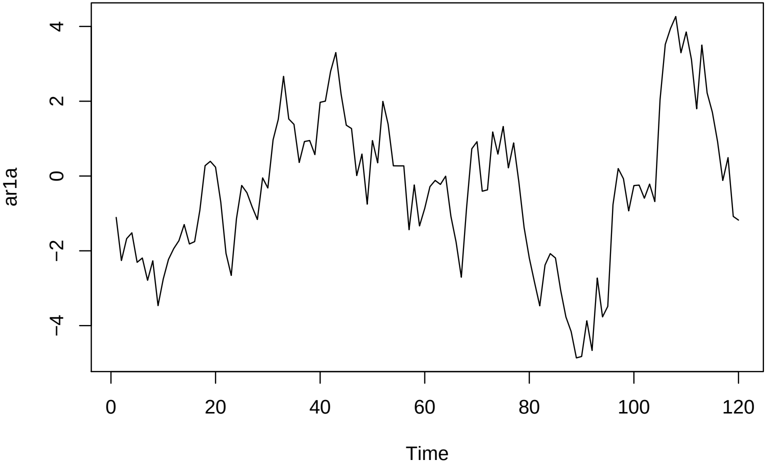


Specification of simulated time series

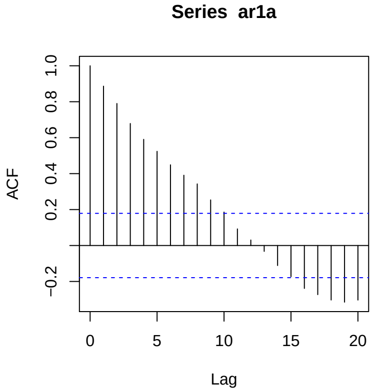


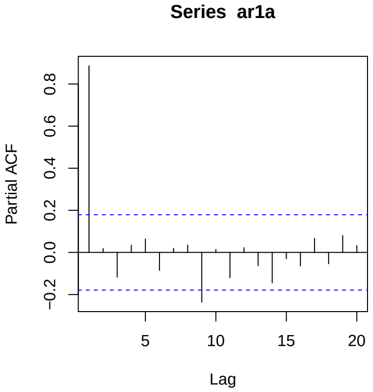
Specification of simulated time series

Illustration: AR(1) with $\phi = 0.9$.



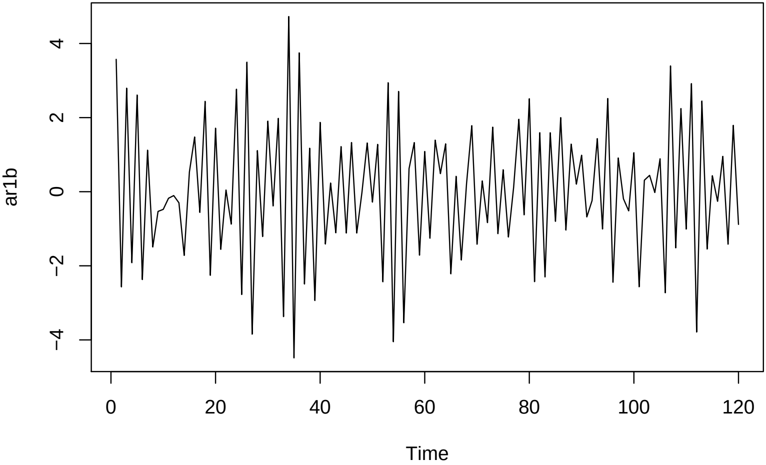
Specification of simulated time series



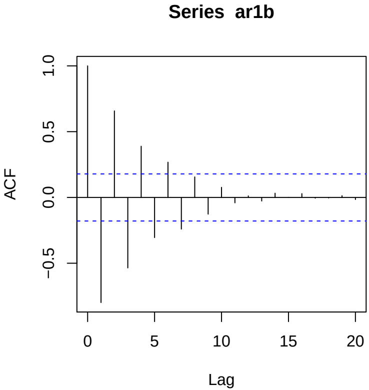


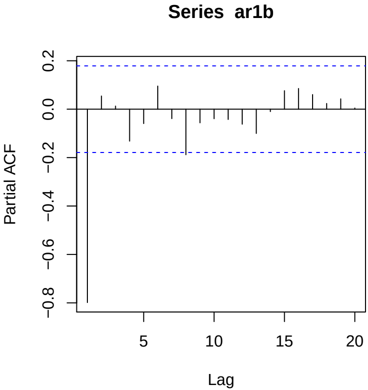
Specification of simulated time series

Illustration: AR(1) with $\phi = -0.8$.



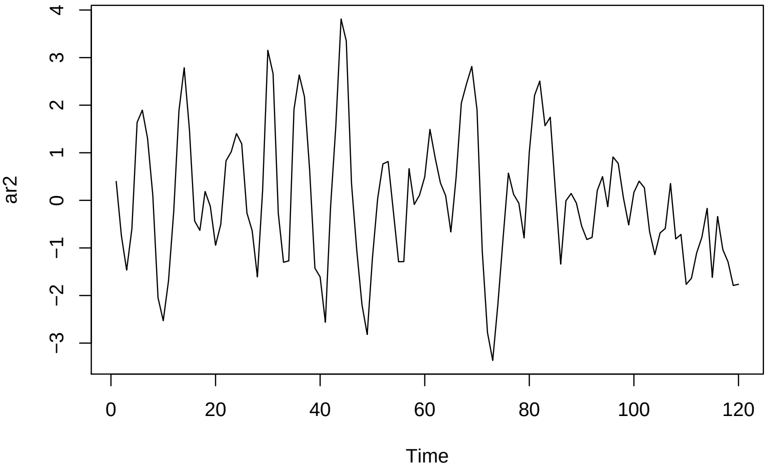
Specification of simulated time series



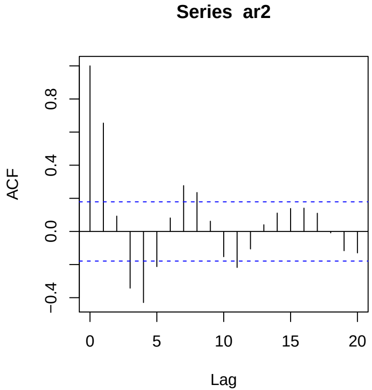


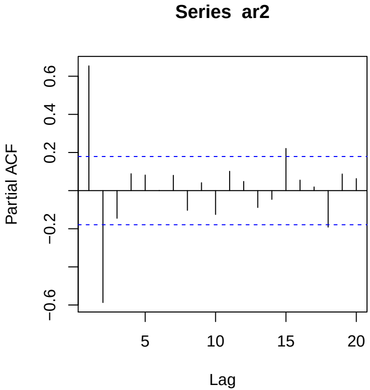
Specification of simulated time series

Illustration: AR(2) with $\phi_1 = 1$ and $\phi_2 = -0.6$.



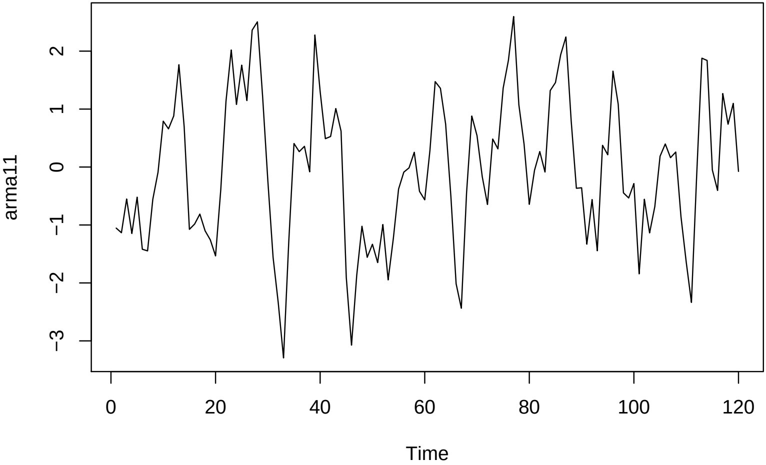
Specification of simulated time series



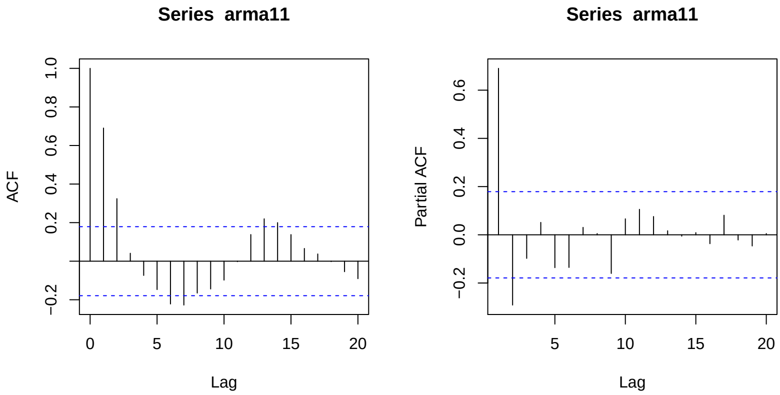


Specification of simulated time series

Illustration: ARMA(1, 1) with $\phi = 0.6$ and $\theta = 0.3$.



Specification of simulated time series



Model Specification

Nonstationarity

Nonstationarity

Recall: Many nonstationary series can be captured by integrated ARMA models.

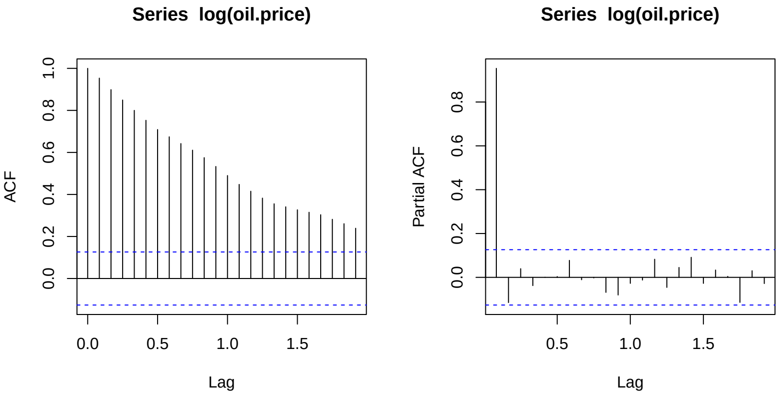
Question: What are the properties of the sample (P)ACF for integrated series?

Remark: Note that the (P)ACF is not even well-defined for nonstationary series. (P)ACF always relies on (co)variances being constant over time.

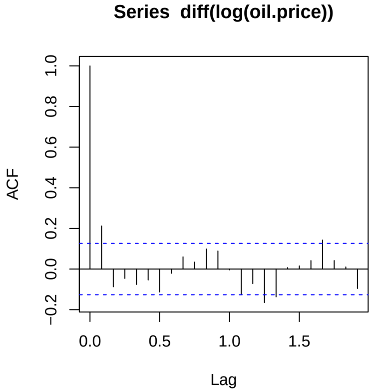
However: Sample ACF typically fails to die out rapidly, i.e., decays only slowly. PACF at lag 1 often close to 1.

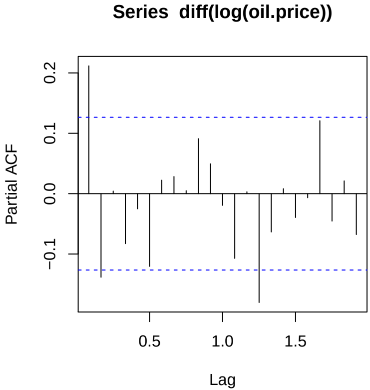
Illustration: (P)ACF for log oil prices and associated differences.

Nonstationarity



Nonstationarity





Overdifferencing

Clear: Differences of stationary series are also stationary.

However: Overdifferencing introduces unnecessary correlations.

Example: $\{Y_t\}$ random walk.

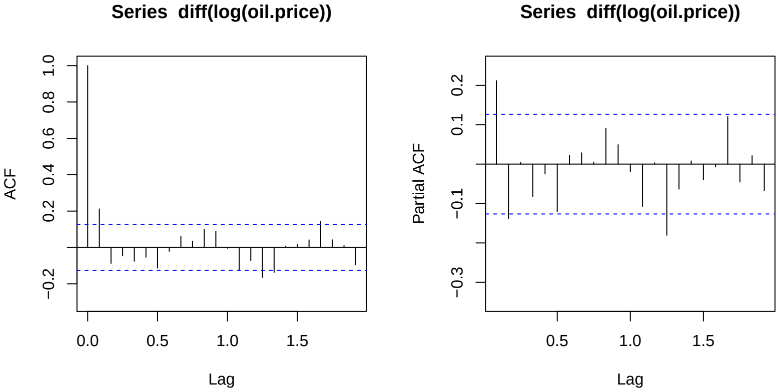
$$\begin{aligned} Y_t &= Y_{t-1} + e_t \\ \Delta Y_t &= Y_t - Y_{t-1} = e_t \\ \Delta^2 Y_t &= e_t - e_{t-1} \end{aligned}$$

Remarks:

- $\Delta^2 Y_t$ is MA(1) but with $\theta = 1$.
- Overdifferencing leads to non-invertible model.
- ARIMA(0, 2, 1) model for Y_t is unnecessarily complex.
- ARIMA(0, 1, 0) model for Y_t is more appropriate.

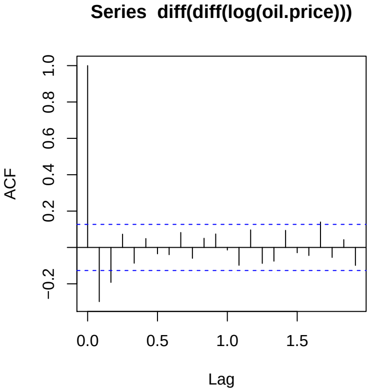
Overdifferencing

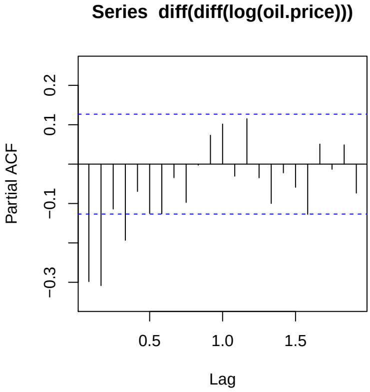
Illustration: Oil prices.



Overdifferencing

Illustration: Oil prices.





Unit root tests

Question: How can we test the null hypothesis that a process is integrated vs. alternative that it is (trend) stationary?

Idea: Consider linear regression

$$Y_t \;=\; \alpha Y_{t-1} \;+\; e_t$$

and test

$$H_0 : \alpha = 1 \quad \text{vs.} \quad H_1 : |\alpha| < 1$$

Problem:

- Asymptotic distribution of OLS estimator $\hat{\alpha}$ is non-standard under the null hypothesis.
- Only for $|\alpha| < 1$, the standard result holds: $\sqrt{n}\,(\hat{\alpha} - \alpha)$ is asymptotically $\mathcal{N}(0, 1 - \alpha^2)$.
- For $\alpha = 1$, $\hat{\alpha}$ is "super-consistent": $n\,(\hat{\alpha} - \alpha)$ has non-degenerate non-normal asymptotic distribution.

Unit root tests

Rewrite:

$$
\begin{aligned}
Y_t &= \alpha Y_{t-1} + e_t \\
Y_t - Y_{t-1} &= (\alpha - 1) Y_{t-1} + e_t \\
\Delta Y_t &= a Y_{t-1} + e_t
\end{aligned}
$$

with $a = \alpha - 1$. Hence, test

$$H_0 : a = 0 \quad \text{vs.} \quad H_1 : a < 0$$

Remarks:

- Standard t statistic for a can be obtained from OLS regression.
- However, asymptotic distribution is also not normal.
- Model is too restrictive: Under alternative $\mathrm{E}(Y_t) = 0$ and e_t is assumed to be white noise.

Unit root tests

More generally: Allow deterministic trend μ_t and more general “error” series X_t.

$$
\begin{aligned}
Y_t &= \mu_t + \alpha Y_{t-1} + X_t \\
\Delta Y_t &= \mu_t + a Y_{t-1} + X_t
\end{aligned}
$$

Assuming X_t is (approximately) an AR(p) process

$$\Delta Y_t = \mu_t + a Y_{t-1} + \phi_1 X_{t-1} + \ldots + \phi_p X_{t-p} + e_t.$$

Under null hypothesis $\alpha = 1$ and additionally $\mu_t = 0$, we would have $X_t = Y_t - Y_{t-1} = \Delta Y_t$. Hence consider auxiliary regression

$$\Delta Y_t = \mu_t + a Y_{t-1} + \phi_1 \Delta Y_{t-1} + \ldots + \phi_p \Delta Y_{t-p} + e_t.$$

Unit root tests

Test: Employ t statistic of $\widehat{a}$ from auxiliary regression as test statistic. This is called *augmented Dickey-Fuller (ADF) test*.

Remarks:

- Null distribution is not normal *and* depends on specification of μ_t.
- Typically, $\mu_t = \beta_1 + \beta_2 t$ is used or $\mu_t = \beta_1$.
- Tables of critical values from the different distributions are available.
- Finite-order AR(p) process may be insufficient approximation of X_t. Hence, increase p along with n, e.g., via information criteria or via heuristics such as $p = \lfloor (n-1)^{1/3} \rfloor$.
- Both the specification of μ_t and the number of lags p should be reported in practice.

Unit root tests

In R:

- Various implementations. None is fully convenient superset of all others.
- `adf.test()` in *tseries*. Employs linear trend and heuristic for selection of *p*.
- `CADFtest()` in *CADFtest*. Supports various μ_t, trend by default. *p* is by default 1 (fixed), but can be selected via information criteria. Additional regressors (and their lags) can be added to auxiliary regression.
- `ur.df()` in *urca*. Supports various μ_t, uses $\mu_t = 0$ by default. *p* is by default 1 (fixed), but can be selected via information criteria.
- `ADF.test()` in *uroot* (employed in Cryer & Chan). Not in active CRAN repository anymore.

Unit root tests

Alternatives:

- Phillips-Perron (PP) test:
  Same idea as ADF, but nonparametric (HAC) correction for autocorrelation.
  In R: `pp.test()` from *tseries*.
- Elliott-Rothenberg-Stock (ERS):
  Same idea as ADF, but GLS detrending.
  In R: `ur.ers()` from *urca*.

Problem: All tests have typically rather poor power for $\alpha = 1 - \varepsilon$.

Stationarity tests

Question: How can we test the null hypothesis of (trend) stationarity against the alternative that a process is integrated?

Idea:

$$Y_t = \mu_t + X_t + e_t$$

- μ_t: deterministic component.
- X_t: random walk.
- e_t: stationary or, more precisely, $I(0)$.

Test: Null hypothesis $H_0 : X_t \equiv 0$.

Versions:

- Level stationarity (under H_0): $\mu_t = \beta_1$.
- Trend stationarity (under H_0): $\mu_t = \beta_1 + \beta_2 t$.

Stationarity tests

Idea: Residuals from auxiliary OLS regression under H_0 should not fluctuate “too much”.

$$\widehat{e}_t \;=\; Y_t \;-\; \hat{\mu}_t$$

Kwiatkowski, Phillips, Schmidt, Shin (KPSS) suggest

$$\begin{aligned} KPSS \;&=\; \frac{1}{n^2\;\hat{\sigma}_e^2}\sum_{t=1}^{n} S_t^2 \\ &=\; \frac{1}{n^2\;\hat{\sigma}_e^2}\sum_{t=1}^{n}\left(\sum_{j=1}^{t}\widehat{e}_j\right)^2 \end{aligned}$$

where $\hat{\sigma}_e^2$ is the Newey-West HAC estimate of the variance (so-called long-run variance).

In R: `kpss.test()` in *tseries*. (Or `ur.kpss()` in *urca*.)

Model Specification

Other Specification Methods

Other specification methods

Information criteria:

- Do not choose p and q in ARIMA(p, d, q) in advance, but estimate all conceivable models in certain ranges for p and q.
- Select that model which optimizes some (penalized) objective function, typically the maximized likelihood.
- Details after model estimation.

Subset ARMA models:

- Idea: Only some of the p AR and q MA coefficients, respectively, are non-zero.
- p and q give order of maximal non-zero coefficient but only subset of remaining coefficients is needed.

Model Specification

Specification of Some Actual Time Series

Actual time series: Oil prices

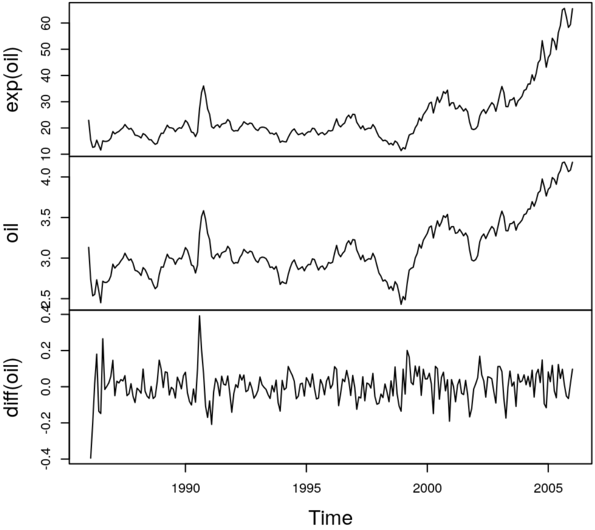
Illustration: Oil prices series.

- Evidence that log prices are non-stationary or, more precisely, integrated.
- Corresponding returns appear to be stationary.
- ARIMA(0, 1, 1) model seems to be appropriate for log prices.
- Outliers may be problematic.

In R:

```
R> data("oil.price", package = "TSA")
R> oil <- log(oil.price)
R> oil.ret <- diff(oil)
R> plot(ts.union(exp(oil), oil, diff(oil)), main = "")
```

Actual time series: Oil prices



Actual time series: Oil prices

Augmented Dickey-Fuller test:

```
R> library("tseries")
R> adf.test(oil)

        Augmented Dickey-Fuller Test

data:  oil
Dickey-Fuller = -1.1, Lag order = 6, p-value = 0.9
alternative hypothesis: stationary

R> adf.test(oil.ret)

        Augmented Dickey-Fuller Test

data:  oil.ret
Dickey-Fuller = -6.7, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(oil.ret) : p-value smaller than printed p-value
```

Actual time series: Oil prices

Equivalently, using *CADFtest* package:

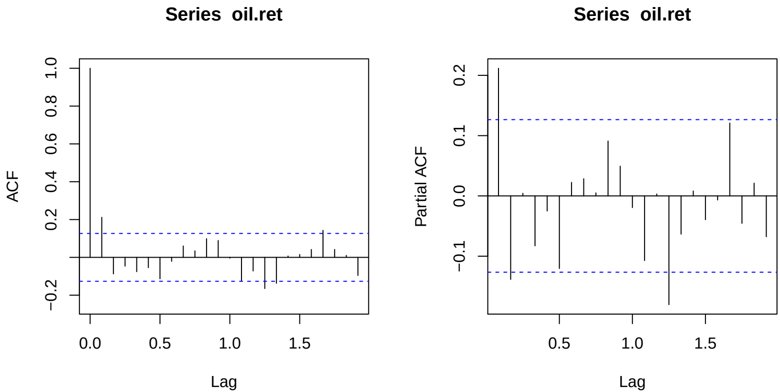
```
R> library("CADFtest")
R> CADFtest(oil ~ 1, max.lag.y = 6)

        ADF test

data:  oil ~ 1
ADF(6) = -1.1, p-value = 0.9
alternative hypothesis: true delta is less than 0
sample estimates:
   delta
-0.02431
```

Actual time series: Oil prices

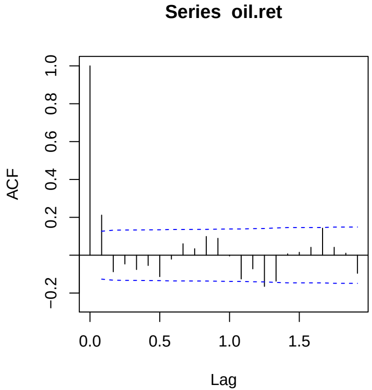
KPSS stationarity test:

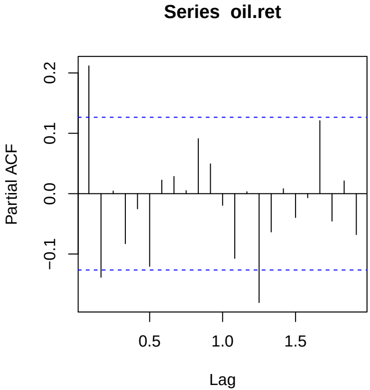
```
R> kpss.test(oil)

        KPSS Test for Level Stationarity

data:  oil
KPSS Level = 2.5, Truncation lag parameter = 4, p-value
= 0.01

Warning message:
In kpss.test(oil) : p-value smaller than printed p-value
R> kpss.test(oil.ret)

        KPSS Test for Level Stationarity

data:  oil.ret
KPSS Level = 0.19, Truncation lag parameter = 4, p-value
= 0.1

Warning message:
In kpss.test(oil.ret) : p-value greater than printed p-value
```

Actual time series: Oil prices



Actual time series: Oil prices





Actual time series: DAX

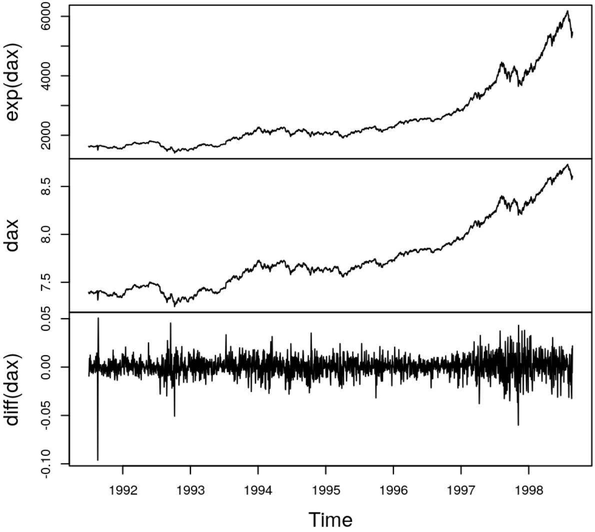
Illustration: DAX closing prices.

- Evidence that log prices are non-stationary or, more precisely, integrated.
- Corresponding returns appear to be (close to) white noise.
- ARIMA(0, 1, 0) model seems to be appropriate for log prices.

In R:

```
R> data("EuStockMarkets", package = "datasets")
R> dax <- log(EuStockMarkets[, "DAX"])
R> dax.ret <- diff(dax)
R> plot(ts.union(exp(dax), dax, diff(dax)), main = "")
```

Actual time series: DAX



Actual time series: DAX

```
R> kpss.test(dax)

        KPSS Test for Level Stationarity

data:  dax
KPSS Level = 18, Truncation lag parameter = 8, p-value =
0.01

Warning message:
In kpss.test(dax) : p-value smaller than printed p-value
R> adf.test(dax.ret)

        Augmented Dickey-Fuller Test

data:  dax.ret
Dickey-Fuller = -11, Lag order = 12, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(dax.ret) : p-value smaller than printed p-value
```

Actual time series: DAX